Efficient Non-stationary Online Learning by Wavelets with Applications to Online Distribution Shift Adaptation

Yu-Yang Qian [1 2] Peng Zhao [1 2] Yu-Jie Zhang [3] Masashi Sugiyama [4 3] Zhi-Hua Zhou [1 2]

Abstract

Dynamic regret minimization offers a principled way for non-stationary online learning, where the algorithm's performance is evaluated against changing comparators. Prevailing methods often employ a two-layer online ensemble, consisting of a group of base learners with different configurations and a meta learner that combines their outputs. Given the evident computational overhead associated with two-layer algorithms, this paper investigates how to attain optimal dynamic regret *without* deploying a model ensemble. To this end, we introduce the notion of *underlying dynamic regret*, a specific form of the general dynamic regret that can encompass many applications of interest. We show that almost optimal dynamic regret can be obtained using a single-layer model alone. This is achieved by an adaptive restart equipped with wavelet detection, wherein a novel streaming wavelet operator is introduced to online update the wavelet coefficients via a carefully designed binary indexed tree. We apply our method to the *online label shift* adaptation problem, leading to new algorithms with optimal dynamic regret and significantly improved computation/storage efficiency compared to prior arts. Extensive experiments validate our proposal.

1. Introduction

Non-stationary online learning is an emerging field that has received much attention in recent years, with appeals both in theory and practice (Besbes et al., 2015; Zhang et al., 2018; Baby & Wang, 2019; Cutkosky, 2020; Zhao et al., 2020; Wu et al., 2021; Zhang et al., 2023a; Zhao et al., 2024). A standard formulation is the online convex optimization framework (Hazan, 2016), in which the online learning process is deemed as a T-round iterative game between a learner and the environment. At iteration $t \in \{1, \ldots, T\}$, the learner selects a decision $\boldsymbol{\theta}_t$ from a convex set $\Theta \subseteq \mathbb{R}^d$, and the environment simultaneously selects an online function $f_t : \Theta \to \mathbb{R}$. Subsequently, the learner will suffer a loss $f_t(\boldsymbol{\theta}_t)$ and observe certain gradient information as the feedback. Recent developments have demonstrated a principled way for non-stationary online learning based on *dynamic regret minimization* — aiming to optimize the cumulative regret against *changing* comparators,

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathbf{u}_t\}_{t=1}^T) \triangleq \sum_{t=1}^{T} f_t(\boldsymbol{\theta}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t). \qquad (1)$$

The comparators $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \Theta$ can be arbitrarily chosen to model the unknown changes of non-stationary environments, so a desired dynamic regret upper bound should hold for all feasible comparators universally. Therefore, this measure is usually referred to as *universal* dynamic regret, and there have been rich theoretical developments (Zhang et al., 2018; Cutkosky, 2020; Zhao et al., 2020; Baby & Wang, 2021; Zhao et al., 2024) as well as applications to online distribution shift adaptation (Bai et al., 2022; Zhang et al., 2023a; Baby et al., 2023; Qian et al., 2023; Wu et al., 2024).

To handle the fundamental uncertainty due to the unknown environmental non-stationarity (e.g., manifested as the variation quantity $P_T = \sum_{t=2}^T \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$), prevailing methods often employ a two-layer online ensemble to optimize (1), which maintains diverse multiple base learners and uses a meta learner to combine them to track the best one on the fly. While achieving optimal dynamic regret, the computational overhead associated with this ensemble structure is evident, and it remains unclear how to attain optimal dynamic regret *without* deploying an ensemble.

In this paper, we discover that while it is hard to use a single-layer model to trace *all* kinds of possible comparators, many real-world problems exhibit specific structures that contain certain information about comparators. We propose a new performance measure named *underlying dynamic regret*, where the learner can observe an unbiased empirical es-

[1] National Key Laboratory for Novel Software Technology, Nanjing University, China [2] School of Artificial Intelligence, Nanjing University, China [3] The University of Tokyo, Chiba, Japan [4] RIKEN AIP, Tokyo, Japan. Correspondence to: Peng Zhao <zhaop@lamda.nju.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

timation of the underlying comparator at each round. The formulation is presented in Eq. (4) of Section 2. This measure is a special form of general dynamic regret, but we demonstrate that it already encompasses many applications of interests, including online label shift (Wu et al., 2021).

For optimizing underlying dynamic regret, we demonstrate that deploying a *single-layer model* alone can provably achieve *almost optimal* dynamic regret.[1] This is accomplished through an adaptive restart scheme that resets the learning model when detected environmental changes exceed a certain threshold. The key lies in specifying suitable restarting criteria. We employ a wavelet-based detection inspired by the line of works in (online) trend filtering (Baby & Wang, 2019; 2020). By decomposing the observed empirical comparators using a series of orthogonal wavelet bases, we capture high-frequency, short-duration noises and low-frequency, long-duration trends. This trend information provides an estimation of the intensity of environmental changes, prompting our method to restart the model whenever *the norm of wavelet coefficients* exceeds a threshold. Therefore, it becomes crucial to calculate this norm online, i.e., the restart criteria. To this end, we propose a novel *streaming wavelet operator*, which organizes the coefficients using a binary indexed tree by lazily updating only a subset and removing outdated ones. This operator not only achieves an exponential speed-up in computational and storage complexities compared to previous methods, but also demonstrates favorable parallelism, making it suitable for practical deployments on GPU facilities. Furthermore, our detection module and streaming wavelet operator are flexible enough to capture *higher-order smoothness* in online data, enabling them to handle complex non-stationarity patterns beyond simple linear gradual changes.

We apply our proposed method to adapt to online label shift, in which label distribution $\mathcal{D}_t(y)$ changes over time while class-conditional distribution $\mathcal{D}_t(\mathbf{x} \mid y)$ remains unchanged. We demonstrate that by using certain unbiased risk estimators, OLS can be framed as a problem of underlying dynamic regret minimization. This leads to new algorithms, exhibiting significant advantages in terms of the computation and storage efficiency compared to prior arts based on ensemble structures (Bai et al., 2022; Baby et al., 2023), and importantly, maintaining the same *optimal* dynamic regret guarantees. Extensive experiments validate our proposals.

Related Work. We here discuss several most relevant works and include more discussions in Appendix B. First, using wavelets for non-stationarity detection was explored in online trend filtering. Baby & Wang (2019) focus on the first-order smoothness of online data and develop an efficient Haar wavelet-based detection along with an incremental update mechanism. Subsequently, Baby & Wang (2020) extend the result to scenarios involving higher-order smoothness. However, their method requires storing the entire sequence and recalculating all wavelet coefficients for each new element, making it less suitable for online updates. We have addressed the challenge by proposing a novel streaming wavelet operator. Additionally, previous research focuses on online trend filtering, primarily one-dimensional estimation with squared loss, and our framework is suitable for the general online convex optimization setting. Our approach also introduces other technical improvements, such as removing recentering/padding operations for wavelets, with discussions presented in Appendix B.2.

Organization. Section 2 introduces the performance measure. Section 3 presents our general wavelet-based framework. Section 4 provides the applications to online label shift. Section 5 reports experiments. We conclude in Section 6. Due to page limits, we defer more empirical studies in Appendix A and related works in Appendix B. Appendix C contains related background for wavelet analysis and others. All the proofs are in Appendix D.

2. Performance Measure

Before introducing our proposed "underlying dynamic regret" measure, we start with two special variants of universal dynamic regret. Although these can be optimally optimized using single-layer algorithms, they are not ideally suited for the non-stationary online learning scenario.

The first one is the classical *static regret* (Hazan, 2016),

$$\mathbf{Reg}_T(\{f_t, \boldsymbol{\theta}\}_{t=1}^T) \triangleq \sum_{t=1}^T f_t(\boldsymbol{\theta}_t) - \min_{\boldsymbol{\theta}\in\Theta} \sum_{t=1}^T f_t(\boldsymbol{\theta}), \quad (2)$$

which compares the online learner's performance against the best fixed decision in hindsight. The second variant is the *worst-case dynamic regret* (Zhao & Zhang, 2021),

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \boldsymbol{\theta}_t^\star\}_{t=1}^T) \triangleq \sum_{t=1}^T f_t(\boldsymbol{\theta}_t) - \sum_{t=1}^T f_t(\boldsymbol{\theta}_t^\star), \quad (3)$$

where $\boldsymbol{\theta}_t^\star \in \arg\min_{\boldsymbol{\theta}\in\Theta} f_t(\boldsymbol{\theta})$ is a minimizer of the online function. Static regret minimization has been well-explored in the field of online learning and can be effectively optimized by, for example, the mirror descent framework (Nemirovskij & Yudin, 1983). For minimizing worst-case dynamic regret, prior art (Zhao & Zhang, 2021) demonstrates that a simple greedy strategy to select the last online function's minimizer as the current decision can achieve an optimal rate of $\mathcal{O}(P_T^\star)$, where $P_T^\star = \max_{\{\boldsymbol{\theta}_t^\star\}_{t=1}^T} \sum_{t=2}^T \|\boldsymbol{\theta}_{t-1}^\star - \boldsymbol{\theta}_t^\star\|_2$ is the path length.

[1] Concretely, our result achieves minimax optimality for exp-concave and strongly convex functions. Despite exhibiting suboptimality for convex functions, it is the best-known rate for single-layer models (with technical challenge discussed in Remark 5).

However, both static regret and worst-case dynamic regret are often not favorable in changing environments. Consider online supervised learning with $f_t(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}, \mathbf{z}_t)$, where $\ell : \Theta \times \mathcal{Z} \to \mathbb{R}$ is the loss function and $\mathbf{z}_t = (\mathbf{x}_t, y_t) \in \mathcal{Z}$ is a data sampled from distribution $\mathcal{D}_t$. Optimizing static regret apparently fails to adapt to the changing distributions. On the other hand, optimizing worst-case dynamic regret can result in severe *overfitting* to sample randomness: f_t merely provides an empirical approximation of the expected function $F_t(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z}_t \sim \mathcal{D}_t}[\ell(\boldsymbol{\theta}, \mathbf{z}_t)]$, while the expected one is our true optimization objective. Instead, optimizing universal dynamic regret (1) is more reasonable as it supports comparison to arbitrary comparator sequence, hence including the one with $\boldsymbol{\theta}_t^\dagger \in \arg\min_{\boldsymbol{\theta}\in\Theta} F_t(\boldsymbol{\theta})$, the best feasible solution tailored to the underlying distribution. Nevertheless, this optimization relies on a two-layer ensemble, leading to an evident computational overhead.

This work introduces a special form of universal dynamic regret, named as *underlying dynamic regret*, defined as

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) \triangleq \sum_{t=1}^T f_t(\boldsymbol{\theta}_t) - \sum_{t=1}^T f_t(\mathring{\mathbf{u}}_t), \tag{4}$$

where $\mathring{\mathbf{u}}_t \in \Theta$ is the ground-truth comparator characterizing the *underlying distribution* at round t. Besides, the learner can access an unbiased empirical estimation $\widetilde{\mathbf{u}}_t$ after making the prediction, formally defined in the *observation model*.

Assumption 1 (observation model)**.** *At iteration $t \in \{1, \ldots, T\}$, the learner observes $\widetilde{\mathbf{u}}_t$ satisfying $\mathbb{E}[\widetilde{\mathbf{u}}_t] = \mathring{\mathbf{u}}_t$, with a bounded variance of σ^2, i.e., $\mathbb{V}[\widetilde{\mathbf{u}}_t] = \frac{1}{d}\|\widetilde{\mathbf{u}}_t - \mathring{\mathbf{u}}_t\|_2^2 \le \sigma^2$.*

The observation model is sufficiently general to encompass many learning problems. A prominent example is the online label shift (OLS) problem (Wu et al., 2021; Bai et al., 2022), where the optimizer of the expected function is used as the comparator to avoid overfitting. In OLS, the comparator $\mathring{\mathbf{u}}_t$ (ground-truth label distribution) is empirically accessible via the unbiased estimator $\widetilde{\mathbf{u}}_t$ (empirical estimator), thus satisfying Assumption 1. More detailed elaboration is deferred to Section 4.2. Besides the OLS problem, the observation model is also applicable to online non-parametric regression (Baby & Wang, 2019) and online density ratio estimation (Zhang et al., 2023a) with the least square.

We note that the underlying dynamic regret problem shares similarities with the non-stationary stochastic optimization problem (Besbes et al., 2015) and the problem of dynamic regret minimization for Stochastically Extended Adversarial (SEA) model (Chen et al., 2023), where a stochastic loss function $\widetilde{f}_t$ satisfying $\mathbb{E}[\widetilde{f}_t(\mathbf{x})] = f_t(\mathbf{x})$, for all $\mathbf{x} \in \mathcal{X}$ is observed by the learner. All of these studies serve as *interpolation* between the worst-case and universal dynamic regret minimization problems by considering *stochastic feedback*. The main difference between our model and previous ones is that we consider a stochastic comparator instead of a stochastic function. By focusing on the specific structure of the stochastic comparator, we achieve a tight dynamic regret bound with a single-layer algorithm, while previous methods typically require a two-layer ensemble structure to handle non-stationary environments.

Non-stationarity Measure. A desired dynamic regret bound should scale with a certain non-stationarity measure. We introduce the *k-th order path length* defined as

$$P_T^k \triangleq T^k \|\boldsymbol{D}^{k+1}\mathring{\mathbf{u}}_{[1,T]}\|_1, \text{ for } k \ge 0, \tag{5}$$

to quantify the fluctuation of comparators, where $\mathring{\mathbf{u}}_{[1,T]} = [\mathring{\mathbf{u}}_1, \ldots, \mathring{\mathbf{u}}_T]^\top \in \mathbb{R}^{T\times d}$ is the matrix consisting of underlying comparators. Moreover, $\boldsymbol{D}^k \in \mathbb{R}^{(T-k)\times T}$ is the k-th order discrete difference matrix (Tibshirani, 2014), obtained recursively by applying $\boldsymbol{D}^i = \widetilde{\boldsymbol{D}}^1 \cdot \boldsymbol{D}^{i-1}\ \forall i \ge 2$ with $\widetilde{\boldsymbol{D}}^1$ being the $(T-i)\times(T-i+1)$ truncation of $\boldsymbol{D}^1$. For the first order case, $\boldsymbol{D}^1 = \mathbf{subdiag}(1, \ldots, 1) - \mathbf{I}_d$, where $\mathbf{I}_d$ is the identity matrix and $\mathbf{subdiag}(\cdot, \ldots, \cdot)$ is the subdiagonal located above the main diagonal. Consequently, when $k = 0$, we have $P_T^0 = \sum_{t=2}^T \|\mathring{\mathbf{u}}_{t-1} - \mathring{\mathbf{u}}_t\|_1$, which recovers the commonly used path length (Zhang et al., 2018).

Remark 1 (higher-order smoothness)**.** The higher the order k in (5), the smoother the comparators will be. For instance, in the case of a linearly varying comparator sequence, we observe that $P_T^1 = 0$ while $P_T^0 = \mathcal{O}(T)$. Hence, higher-order path length offers a more precise measure of non-stationarity, particularly when the underlying environments undergo changes that are beyond merely linear shifts. ¶

Remark 2 (naïve solution)**.** Given the observations of $\{\widetilde{\mathbf{u}}_t\}_{t=1}^T$, one might consider deploying OGD with step size $\eta \propto \sqrt{P_T^{\sim}/T}$, where $P_T^{\sim} = \sum_{t=2}^T \|\widetilde{\mathbf{u}}_t - \widetilde{\mathbf{u}}_{t-1}\|_1$. However, this will additionally introduce an $\mathcal{O}(T\sigma)$ cumulative error due to the sample randomness, only yielding an $\mathcal{O}(\sqrt{T(1+P_T^0)} + T\sigma)$ dynamic regret in terms of P_T^0, hence vacuous due to the linear dependence in T. ¶

3. Wavelet-based Framework

This section presents our general detection-restart framework based on the wavelet analysis for non-stationary online learning. We first introduce the detection module and then describe our designed streaming wavelet operator, followed by the dynamic regret analysis.

3.1. Detection Module Based on Wavelets

We propose a detection-restart based method, which restarts the learning model whenever the detected non-stationarity exceeds a certain threshold. Inspired by previous work in (online) trend filtering (Donoho & Johnstone, 1998; Tibshirani, 2014; Baby & Wang, 2019; 2020), our detection module is built upon *wavelets*. Below, we present details.

Algorithm 1: Detection Module by Wavelets

```
Input: Restart threshold $\gamma$; online algorithm $\mathcal{A}$.
Initialize: coefficient matrix $\widetilde{\boldsymbol{\alpha}} = \mathbf{0}$, time $s = 1$;
for $t = 1, \dots, T$ do
    Update coefficient matrix $\widetilde{\boldsymbol{\alpha}}_{[s,t]}$ as Sec 3.2;
    if $\|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s,t]})\|_{\mathrm{F}} > \gamma$ then
        Restart the online algorithm $\mathcal{A}$;
        Reset coefficient matrix $\widetilde{\boldsymbol{\alpha}} = \mathbf{0}$, set $s = t + 1$;
    Output the prediction $\boldsymbol{\theta}_t$ using $\mathcal{A}$;
    Suffer loss $f_t(\boldsymbol{\theta}_t)$, observe $\widetilde{\mathbf{u}}_t$, and update $\mathcal{A}$;
end
```

Algorithm 2: Streaming Wavelet Operator

```
Initialize: Wavelet coefficients $\widetilde{\boldsymbol{\alpha}} = 0$;
for $t = 1, \dots, |\mathcal{I}|$ do
    Update the binary indexed tree;
    $\textsc{Update}_{\boldsymbol{\alpha}}(t) = \emptyset$, $\textsc{Drop}_{\boldsymbol{\alpha}}(t) = \emptyset$;
    for $j = 0, \dots, \lceil \log_2 |\mathcal{I}| \rceil$ do
        add $\lfloor t/2^j \rfloor$ into $\textsc{Update}_{\boldsymbol{\alpha}}(t)$;
        add $\lfloor t/2^j \rfloor - 1$ into $\textsc{Drop}_{\boldsymbol{\alpha}}(t)$.
    end
    Update $\widetilde{\alpha}_i \in \textsc{Update}_{\boldsymbol{\alpha}}(t)$, delete $\widetilde{\alpha}_i \in \textsc{Drop}_{\boldsymbol{\alpha}}(t)$.
end
```

Considering an interval $[s, t] \subseteq [T]$, wavelet analysis decomposes the input signals (in our case, the empirically observed comparators $\{\widetilde{\mathbf{u}}_\tau\}_{\tau=s}^t$) into their time and frequency components. This process yields a series of wavelet coefficients constituting the *coefficient matrix* $\widetilde{\boldsymbol{\alpha}}_{[s,t]} = [\widetilde{\boldsymbol{\alpha}}_s, \dots, \widetilde{\boldsymbol{\alpha}}_{s+|\mathcal{I}|-1}]^\top \in \mathbb{R}^{|\mathcal{I}| \times d}$, where $|\mathcal{I}| = 2^{\lceil \log_2 (t-s) \rceil}$ is the length of coefficients. In this study, we adopt a classic wavelet construction, Cohen-Daubechies-Jawerth-Vial (CDJV) wavelets (Cohen et al., 1993a;b), to construct wavelet coefficients, which admits desirable properties such as orthogonality, vanishing moments, and compact support. Assuming all input signals are available offline, we can calculate wavelet coefficients in a standard way through matrix multiplication according to the textbook (Daubechies, 1992, Chapter 3), with an $\mathcal{O}(|\mathcal{I}|)$ computational complexity. However, in an online scenario, such calculations would incur an update cost *polynomial in* T at each round. For instance, it can be the case that the entire horizon is divided into $\mathcal{O}(\sqrt{T})$ periods, with each having a length of $|\mathcal{I}| = \mathcal{O}(\sqrt{T})$. In Section 3.2, we propose a *streaming wavelet operator* to update coefficients in an online manner.

Based on the coefficient matrix $\widetilde{\boldsymbol{\alpha}}_{[s,t]}$, we can separate the following two parts from the empirical comparators: (i) high-frequency and short-duration noises, and (ii) low-frequency, long-duration trends. By filtering out the noisy components, we thus approximately track the underlying comparators $\{\mathring{\mathbf{u}}_\tau\}_{\tau=s}^t$ and estimate the intensity of environmental changes. Technically, this can be realized by calculating the Frobenius norm $\|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s,t]})\|_{\mathrm{F}}$ as the criteria, where $\delta_\gamma : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ is the soft-threshold operator (Donoho & Johnstone, 1998) defined as $[\delta_\gamma(A)]_{i,j} = \mathbf{sign}(A_{i,j}) \cdot \max\{|A_{i,j}| - \gamma, 0\}$. Whenever the F-norm $\|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s,t]})\|_{\mathrm{F}}$ exceeds a predefined threshold $\gamma > 0$, the learner restarts the online algorithm $\mathcal{A}$. Furthermore, we note that the wavelets analysis can also accommodate the k-th order path length by employing the $(k+1)$-th order construction of CDJV wavelets. Due to page limits, we defer the details of wavelets to Appendix C.2.

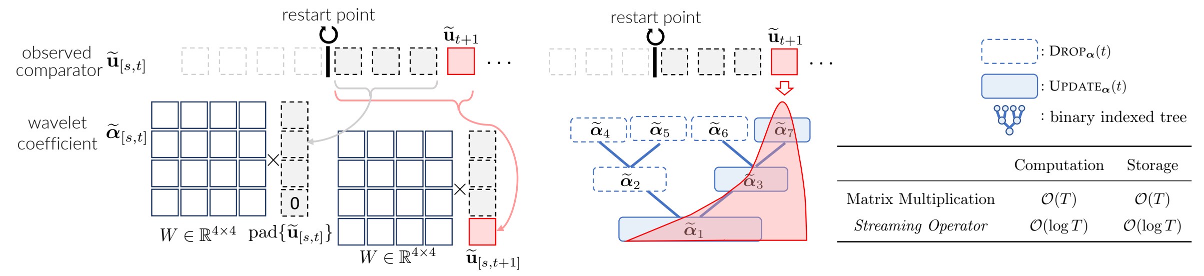
Algorithm 1 summarizes procedures of the detection module, which divides the time horizon into multiple piecewise-stationary intervals, enabling online algorithm to be performed within stationary environments by restarting. Provided that the update cost of wavelet coefficients is moderate (which we will ensure in Section 3.2), this detection-restart based method can be highly efficient due to its single-layer structure. In contrast, previous ensemble-based methods typically maintain $\mathcal{O}(\log T)$ base learners (Zhang et al., 2018; Zhao et al., 2020; Cutkosky, 2020), which substantially increases the computational complexity.

3.2. Streaming Wavelet Operator

In this part, we describe our designed procedure of efficiently calculating the wavelet coefficient matrix $\widetilde{\boldsymbol{\alpha}}_{[s,t]} = [\widetilde{\boldsymbol{\alpha}}_s, \dots, \widetilde{\boldsymbol{\alpha}}_{s+|\mathcal{I}|-1}]^\top \in \mathbb{R}^{|\mathcal{I}| \times d}$ for a given interval $[s, t] \subseteq [T]$, where $|\mathcal{I}| = 2^{\lceil \log_2 (t-s) \rceil}$ is the length of coefficients.

As shown in Figure 1(a), traditional methods calculate wavelet coefficients through the matrix multiplication $\widetilde{\boldsymbol{\alpha}}_{[s,t]} = W_{\mathcal{I}}^\top \cdot \mathrm{pad}\{\widetilde{\mathbf{u}}_{[s,t]}\}$, where $W_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{I}|}$ is the CDJV wavelet transformation matrix (the precise definition can be found in Appendix C.2.1), and $|\mathcal{I}| = 2^{\lceil \log_2 (t-s) \rceil}$ is the size of the wavelet transformation matrix, which is an integer of power of 2. The padding operator $\mathrm{pad}\{\cdot\}$ completes the sequence as a longer length of $|\mathcal{I}|$ and lets the extra padded elements be zero. Therefore, this calculation of coefficients requires storing all elements in the interval to calculate the coefficients, leading to an $\mathcal{O}(|\mathcal{I}|)$ computational and storage complexities. Note that this brings an update cost linear in T since in the worst case, $|\mathcal{I}| = \mathcal{O}(\mathrm{poly}(T))$. Even worse, this method needs to recalculate all coefficients each round upon encountering a new element, making it unsuitable for the online scenario.

To this end, we propose a *streaming wavelet operator*. We observe that wavelets decompose a sequence of signals via imposing a *convolution* operation on the inputs based on a set of orthogonal wavelet functions. Therefore, the arrival of a new element affects only a portion of coefficients. Consequently, we employ a *binary indexed tree* to organize coefficients. We first give a high-level intuition here: the binary indexed tree is employed to determine which part of


restart point
observed comparator $\widetilde{\mathbf{u}}_{[s,t]}$
$\widetilde{\mathbf{u}}_{t+1}$
wavelet coefficient $\widetilde{\boldsymbol{\alpha}}_{[s,t]}$
0
$W \in \mathbb{R}^{4\times 4}$
pad$\{\widetilde{\mathbf{u}}_{[s,t]}\}$
$W \in \mathbb{R}^{4\times 4}$
$\widetilde{\mathbf{u}}_{[s,t+1]}$
restart point
$\widetilde{\mathbf{u}}_{t+1}$
$\widetilde{\boldsymbol{\alpha}}_4$
$\widetilde{\boldsymbol{\alpha}}_5$
$\widetilde{\boldsymbol{\alpha}}_6$
$\widetilde{\boldsymbol{\alpha}}_7$
$\widetilde{\boldsymbol{\alpha}}_2$
$\widetilde{\boldsymbol{\alpha}}_3$
$\widetilde{\boldsymbol{\alpha}}_1$
: $\textsc{Drop}_{\boldsymbol{\alpha}}(t)$
: $\textsc{Update}_{\boldsymbol{\alpha}}(t)$
: binary indexed tree

	Computation	Storage
Matrix Multiplication	$\mathcal{O}(T)$	$\mathcal{O}(T)$
Streaming Operator	$\mathcal{O}(\log T)$	$\mathcal{O}(\log T)$

(a) traditional matrix computation

(b) streaming wavelet operator

Figure 1: Comparison of our streaming wavelet operator (b) with traditional wavelet computation (a). When encountering a new element, we lazily update only a portion of coefficients $\textsc{Update}_{\boldsymbol{\alpha}}(t)$ and drop outdated coefficients $\textsc{Drop}_{\boldsymbol{\alpha}}(t)$ using a binary index tree structure, thus reducing computational and storage complexities of updating wavelet coefficients from $\mathcal{O}(T)$ to $\mathcal{O}(\log T)$ at each round.

the coefficients is to be updated and which part is to be removed, based on their indexes in the tree. This tree structure allows for a selective, or "lazy update", where only specific portions of the coefficients are updated in each iteration. As illustrated in Figure 1(b), when the new element arrives, it only affects the wavelet coefficients $\widetilde{\boldsymbol{\alpha}}_1$, $\widetilde{\boldsymbol{\alpha}}_3$, and $\widetilde{\boldsymbol{\alpha}}_7$.

Consider a simple case with $k = 1$ and $d = 1$, where k is order of path length and d is dimension. The *bitwise right shift* operator is denoted as $\gg$. At round t, only coefficients $\textsc{Update}_{\boldsymbol{\alpha}}(t)$ are affected by the incoming element $\widetilde{\mathbf{u}}_t$:

$$\textsc{Update}_{\boldsymbol{\alpha}}(t) = \left\{\widetilde{\boldsymbol{\alpha}}_i \mid \mathbb{1}\{i = (t \gg j)\}, \exists j \in \left[\lfloor \log_2 t \rfloor\right]\right\}. \quad (6)$$

The formula indicates that the i-th coefficient is affected only when $\mathbb{1}\{i = (t \gg j)\}$ holds true. Consequently, we can arrange the coefficients using a binary indexed tree and only update coefficients in the set $\textsc{Update}_{\boldsymbol{\alpha}}(t)$ when encountering a new element $\widetilde{\mathbf{u}}_t$. This update ensures that, at each iteration, at most $\mathcal{O}(\log t)$ coefficients are updated. Additionally, for a d-dimensional signal, the complexity of updating each wavelet coefficient in $\textsc{Update}_{\boldsymbol{\alpha}}(t)$ is $\mathcal{O}(kd)$.

As described in Algorithm 1, the restart criterion is based on F-norm of wavelet coefficients, essentially the squared summation of elements within the matrix. As such, the F-norm of a sequence can be incrementally updated by combining the norm of the newly arrived element with the existing norm, and we only need to maintain the most recent $\lfloor \log_2 t \rfloor$ wavelet coefficients. Therefore, we remove the old coefficients and record their F-norm each time a wavelet coefficient becomes outdated. Formally, let $\textsc{Drop}_{\boldsymbol{\alpha}}(t)$ denote the set of outdated coefficients that no longer affected by new incoming elements at the round t,

$$\textsc{Drop}_{\boldsymbol{\alpha}}(t) = \left\{\widetilde{\boldsymbol{\alpha}}_i \mid \mathbb{1}\{(i \gg j) \& 1 = 1\}, \exists j \in \left[\lfloor \log_2 t \rfloor\right]\right\}, \quad (7)$$

where $\&$ denotes the bitwise AND operator. As a result, we maintain at most $\lceil \log_2 T \rceil$ wavelet coefficients, reducing the computational and storage complexities from $\mathcal{O}(dT)$ of previous matrix multiplication to $\mathcal{O}(d \log T)$ per round. For the more sophisticated cases of k-th order CDJV wavelets, we can apply the same idea and use binary indexed tree to maintain at most $\mathcal{O}(dk \log T)$ wavelet coefficients. The streaming wavelet operator is illustrated in Figure 1(b).

In summary, the streaming wavelet operator can be constructed by leveraging wavelet transform properties and data structures for online updates. This operator reduces the computational and storage complexities from the $\mathcal{O}(T)$ incurred by previous matrix computations to $\mathcal{O}(\log T)$ per round. Moreover, our operator demonstrates notable parallelism, making it suitable for practical online learning deployment on GPU facilities, as discussed in Appendix C.2.2.

Remark 3. We note that Baby & Wang (2019) introduce an efficient Haar wavelet update mechanism to handle the first-order smoothness, but extending it to higher-order cases remains challenging. Notably, even in the special case of first-order smoothness, our method also exhibits significant advantages in efficiency, mainly by eliminating their costly recentering and padding operations: we formally state that recentering is not necessary for streaming wavelet operator as it does not influence the computed coefficients, as detailed in Lemma 6 of Appendix D; besides, our operator performs an "implicit padding" strategy to omit yet-to-arrive elements in the online sequence by leveraging convolution operations, which implicitly complete the sequence as a longer length, thereby improving the efficiency. ¶

3.3. Theoretical Analysis for General Framework

In this part, we provide an analysis of regret as well as computational and storage complexity. Suppose $M - 1$ change points are identified by Algorithm 1, the entire time horizon can be thus decomposed into M intervals denoted by $\{\mathcal{I}_1, \ldots, \mathcal{I}_M\}$ with $\mathcal{I}_i = [s_i, e_i]$ for $i \in [M]$. Then we have $\mathbf{Reg}^{\mathbf{d}}_T(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) = \sum_{i=1}^M \mathbf{Reg}^{\mathbf{d}}_{\mathcal{I}_i}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=s_i}^{e_i})$. Therefore, it suffices to control the regret within each interval $\mathbf{Reg}^{\mathbf{d}}_{\mathcal{I}_i}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=s_i}^{e_i})$ and the total number of intervals M.

Let $|\mathcal{I}_i| = e_i - s_i$ be the length of the i-th interval, $P^k_{\mathcal{I}_i} \triangleq |\mathcal{I}_i|^k \|\boldsymbol{D}^{k+1} \mathring{\mathbf{u}}_{[s_i, e_i]}\|_1$ be the k-th order path length within the interval, and $C^k_{\mathcal{I}_i} = \sum_{t \in \mathcal{I}_i} \|\boldsymbol{\beta}^\top_{\mathcal{I}_i} \boldsymbol{\phi}_t - \mathring{\mathbf{u}}_t\|_1$ be the

k-th order comparator gap that measures the *higher-order smoothness* of the comparators, with a formal definition in Appendix D.1. The comparator gap $C^k_{\mathcal{I}_i}$ quantifies the smoothness of the underlying comparators, thereby reflecting the intensity of environmental change within the interval. A smaller $C^k_{\mathcal{I}_i}$ indicates a smoother comparator sequence $\mathring{\mathbf{u}}_t$ within the interval $\mathcal{I}_i$. The following theorem guarantees that the number of intervals is bounded, and environmental shift within each interval is small.

Theorem 1 (Guarantee for Detection-Restart Framework)**.** *Under Assumption 1, setting the threshold $\gamma = 4\sigma$ ensures that our detection-restart framework divides the entire time horizon into a maximum of $M \leq \widetilde{\mathcal{O}}(T^{\frac{1}{2k+3}}(P^k_T)^{\frac{2}{2k+3}})$ intervals with probability at least $1 - 2/T$. Furthermore, within each interval $\mathcal{I}_i$, $C^k_{\mathcal{I}_i} \leq \widetilde{\mathcal{O}}(|\mathcal{I}_i|^{\frac{k+2}{2k+3}}(P^k_{\mathcal{I}_i})^{\frac{1}{2k+3}})$, where $\widetilde{\mathcal{O}}(\cdot)$ omits the logarithmic factors in T.*

Furthermore, the online algorithm $\mathcal{A}$ is supposed to enjoy a favorable dynamic regret within the interval $\mathcal{I}_i$ for all $i \in [M]$, as formally described below.

Requirement 1. *An online algorithm $\mathcal{A}$ running over interval $\mathcal{I}_i = [s_i, e_i] \subseteq [T]$ is required to satisfy*

$$\sum_{t=s_i}^{e_i} f_t(\boldsymbol{\theta}_t) - \sum_{t=s_i}^{e_i} f_t(\mathring{\mathbf{u}}_t) \leq \mathcal{O}\Big(\sqrt{|\mathcal{I}_i|} + C^k_{\mathcal{I}_i}\Big). \tag{8}$$

Consequently, Requirement 1 essentially requires the online algorithm $\mathcal{A}$ to achieve a good dynamic regret guarantee in a relatively smoothed interval $\mathcal{I}$. Various online algorithms, such as online gradient descent (Zinkevich, 2003) and online Newton step (Hazan et al., 2007), meet Requirement 1, which we will validate later. Combining Theorem 1 and Requirement 1, we obtain the following regret guarantee.

Theorem 2 (Overall Dynamic Regret)**.** *With probability at least $1 - 2/T$, using the detection-restart framework described in Algorithm 1 with an online algorithm $\mathcal{A}$ satisfying Requirement 1 guarantees that*

$$\mathbf{Reg}^{\mathbf{d}}_T(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) \leq \widetilde{\mathcal{O}}\left(\max\left\{T^{\frac{k+2}{2k+3}}(P^k_T)^{\frac{1}{2k+3}}, \sqrt{T}\right\}\right).$$

Furthermore, for the exp-concave and the strongly convex functions, Requirement 1 can be further enhanced to $\sum_{t=s_i}^{e_i}(f_t(\boldsymbol{\theta}_t) - f_t(\mathring{\mathbf{u}}_t)) \leq \widetilde{\mathcal{O}}(1+\sum_{t\in\mathcal{I}_i}\|\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t - \mathring{\mathbf{u}}_t\|_1^2)$, thereby achieving an $\widetilde{\mathcal{O}}(T^{1/2k+3}(P^k_T)^{2/2k+3})$ dynamic regret, proved to be minimax optimal (Baby & Wang, 2020; 2023).

Remark 4 (optimality and higher-order smoothness)**.** For general convex functions, our result in Theorem 2 is suboptimal. For example, the optimal rate for 0-th order path length should be $\mathcal{O}(T^{1/2}(P^0_T)^{1/2})$, but we attain $\mathcal{O}(T^{2/3}(P^0_T)^{1/3})$. Despite this, we remind that the optimal rate is only achieved by ensemble-based methods (Zhang et al., 2018; Cutkosky, 2020), and our result is the best-known rate for the single-layer model. Note that previous best result is $\mathcal{O}(\sqrt{T \cdot P^0_T})$ achieved by OGD with step size $\eta = 1/\sqrt{T}$, and our method can simultaneously enjoy this rate as proved in Appendix D.3. Remark 5 will illustrate the difficulty of achieving optimal dynamic regret for convex functions. Furthermore, it is worth noting that our result is the first to attain dynamic regret with higher-order path length for convex functions, which is *new* to literature. ¶

Remark 5 (achieving optimality for convex functions)**.** While our result for convex functions is suboptimal, attaining an optimal rate of $\mathcal{O}(\sqrt{T \cdot P^0_T})$ (simply focusing on the $k = 0$ scenario) presents significant challenges. Technically, this problem is as difficult as addressing a major unresolved issue in non-stationary online learning: whether dynamic regret minimization can be reduced to strongly adaptive regret minimization, as highlighted in (Zhang, 2020, Section 5). Furthermore, we emphasize that even though the rate for this general OCO scenario may be suboptimal, applying it to our primary application, online label shift, can still yield optimal guarantees owing to the distinct structure of the OLS problem as shown in Section 4.2. ¶

Theorem 3 (Efficiency of Wavelets Update)**.** *The proposed streaming wavelet operator exhibits a computational complexity of $\mathcal{O}(kd\log T)$ per round, with a storage complexity at $\mathcal{O}(kd\log T)$. Here, k denotes the order of path length, and d represents the dimension of the online data sequence.*

Our streaming wavelet operator significantly improves efficiency compared with previous model ensemble methods. For clarity, we define several key terms: the computational complexity of updating a single model (C_{model}), obtaining an unbiased estimator (C_{esti}), and wavelet detection (C_{detect}). Typically, previous model ensemble-based methods incur a computational complexity of $C_{\text{model}} \times \log T$ due to the requirement of maintaining $\mathcal{O}(\log T)$ base learners. In contrast, our wavelet-based detection method maintains only one model, along with a multi-resolution detection/exploration with $\mathcal{O}(\log T)$ wavelet coefficients. Consequently, our wavelet-based detection method is much more efficient with a complexity of $C_{\text{model}} + C_{\text{esti}} + C_{\text{detect}}$, particularly when using complicated base models, such as overparametrized models in practice where C_{model} can be very large. Additionally, as proved in Theorem 3, C_{detect} is only $\mathcal{O}(kd\log T)$, and C_{esti} is often also very small, typically $\widetilde{\mathcal{O}}(d)$, as we will demonstrate with a concrete example of the OLS problem in Section 4. A summary table of efficiency improvements is provided in Appendix C.4, and empirical results in Section 5 further demonstrate the superior efficiency of our method.

4. Applications: Online Distribution Shift

In this section, we apply our detection framework to a specific application: OLS, and obtain novel efficient algorithms.

4.1. Problem Setup

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the feature space and $\mathcal{Y} = \{1, \dots, K\}$ denote the label space for the multi-class classification. There are two stages in the online label shift problem:

(i) In the *offline initialization stage*, the learner can access a number of offline labeled data $S_0 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_0}$ *i.i.d.* drawn from initial distribution $\mathcal{D}_0(\mathbf{x}, y)$, where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. The offline dataset is supposed to be sufficient ($|S_0| \to \infty$) to initialize a good model $h_0 : \mathcal{X} \to \mathcal{Y}$.

(ii) In the *online adaptation stage*, the learner needs to adapt her model $\widehat{h}_t$ for an *unlabeled* data stream with changing distributions. Specifically, in round $t \in [T]$, the learner receives a limited amount of unlabeled data $S_t = \{\mathbf{x}_n\}_{n=1}^{N_t}$, sampled *i.i.d.* from the distribution $\mathcal{D}_t(\mathbf{x})$. The goal is to minimize *expected risk* on $\mathcal{D}_t$ against the optimal classifier at each round, defined as

$$\mathbf{Reg}_T^{\mathbf{d}}(\{R_t, h_t^\star\}_{t=1}^T) \triangleq \sum_{t=1}^{T} R_t(\widehat{h}_t) - \sum_{t=1}^{T} R_t(h_t^\star), \quad (9)$$

where $R_t(h) \triangleq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}[\ell(h(\mathbf{x}), y)]$ is the expected risk, $\ell : \Delta_K \times \mathcal{Y} \to \mathbb{R}$ is the loss function, $h : \mathcal{X} \to \Delta_K$ is the predictive function. The optimal model of each round is denoted by $h_t^\star$, i.e., $h_t^\star \in \arg\min_{h\in\mathcal{H}} R_t(h)$. In OLS, label distribution $\mathcal{D}_t(y)$ changes over time, and class-conditional distribution $\mathcal{D}_t(\mathbf{x} \mid y)$ remains unchanged.

Reduction to Underlying Dynamic Regret Minimization. Our reduction involves constructing an unbiased distribution estimator. For OLS, we can upper bound Eq. (9) by the cumulative error of class prior $\sum_{t=1}^T \|\widehat{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_t\|_2$, where $\widehat{\boldsymbol{\mu}}_t$ is our predicted class prior and $\boldsymbol{\mu}_t = \mathcal{D}_t(y)$ is the ground-truth one. However, the underlying class prior $\boldsymbol{\mu}_t$ is unknown. To this end, Black Box Shift Estimation (BBSE) (Lipton et al., 2018) method is used to obtain an unbiased estimation $\widetilde{\boldsymbol{\mu}}_t$ with bounded variance, therefore satisfying Assumption 1, i.e., only $\widetilde{\mathbf{u}}_t = \widetilde{\boldsymbol{\mu}}_t$ and $f_t(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \widetilde{\boldsymbol{\mu}}_t\|_2$ are observed, while $\mathring{\mathbf{u}}_t = \boldsymbol{\mu}_t$ and $F_t(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \boldsymbol{\mu}_t\|_2$ are expected ones.

4.2. Adapting to Online Label Shift

This part applies our detection-restart framework for OLS.

① *Label Shift Estimation.* We begin with getting an unbiased estimation of the true class prior $\boldsymbol{\mu}_{y_t} = \mathcal{D}_t(y = j)$ to satisfy Assumption 1. To this end, we employ BBSE (Lipton et al., 2018) method to construct an estimator via offline data S_0 and unlabeled data S_t. Specifically, we first use the initial offline model h_0 to predict over unlabeled data S_t and get predicted labels $\widehat{y}_t$; and then we estimate label distribution as $\widetilde{\boldsymbol{\mu}}_{y_t} = C_0^{-1}\widetilde{\boldsymbol{\mu}}_{\widehat{y}_t}$, where $\widetilde{\boldsymbol{\mu}}_{\widehat{y}_t} \in \Delta_K$ with $[\widetilde{\boldsymbol{\mu}}_{\widehat{y}_t}]_j = {}^1\!/_{|S_t|} \sum_{\mathbf{x}\in S_t} [h_0(\mathbf{x})]_j$ is estimated class prior of the prediction $h_0(\mathbf{x})$, and $C_0 \in \mathbb{R}^{K\times K}$ is the confusion matrix with $[C_0]_{i,j} \triangleq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_0(\mathbf{x}|y=j)}\big[[h_0(\mathbf{x})]_i\big]$ being the classification rate that h_0 predicts samples from class i as j.

② *Verifying Assumption 1 for OLS.* BBSE ensures that the estimation $\widetilde{\boldsymbol{\mu}}_{y_t} = C_0^{-1}\widetilde{\boldsymbol{\mu}}_{\widehat{y}_t}$ satisfies $\mathbb{E}[\widetilde{\boldsymbol{\mu}}_{y_t}] = \boldsymbol{\mu}_{y_t}$, where $\boldsymbol{\mu}_{y_t} \triangleq C_0^{-1}\boldsymbol{\mu}_{\widehat{y}_t} = \mathcal{D}_t(y)$ is the ground-truth label distribution. Furthermore, given that the minimum singular value $\sigma_{\min}(C_0) = \Omega(1)$ is bounded away from zero. As a result, $\widetilde{\boldsymbol{\mu}}_{y_t}$ serves as an unbiased estimator for $\boldsymbol{\mu}_{y_t}$ with bounded variance of $1/\sigma_{\min}^2(C_0)$, which satisfies Assumption 1. Afterwards, we use $\boldsymbol{\mu}_t$ as shorthand for $\boldsymbol{\mu}_{y_t}$, and $\widetilde{\boldsymbol{\mu}}_t$ for $\widetilde{\boldsymbol{\mu}}_{y_t}$.

③ *Wavelet Detection for OLS.* The wavelet coefficients are calculated upon $\widetilde{\boldsymbol{\mu}}_{[s,t]}$ using our streaming wavelet operator as described in Section 3.2, where $\widetilde{\boldsymbol{\mu}}_{[s,t]}$ is the class prior estimated by BBSE within any interval $[s, t]$. The wavelet detection framework restarts the classifier when the F-norm of the wavelet coefficients exceeds the variance of BBSE's estimation, i.e., setting $\gamma = 4/\sigma_{\min}^2(C_0)$ in Algorithm 1.

③ *Combined with Online Algorithms.* In the following, we introduce two ways to combine detection-restart framework with previous online algorithms to adapt to OLS, i.e., combined with (i) reweighting-update and (ii) OGD-update.

(i) Combined with Reweighting-Update. Same as (Baby et al., 2023), we employ the predicted class prior to reweight the initial offline classifier to get the prediction. Note that

$$\mathcal{D}_t(y \mid \mathbf{x}) = \frac{\mathcal{D}_t(y)}{\mathcal{D}_t(\mathbf{x})}\frac{\mathcal{D}_0(\mathbf{x})}{\mathcal{D}_0(y)}\mathcal{D}_0(y \mid \mathbf{x}) \propto \frac{\mathcal{D}_t(y)}{\mathcal{D}_0(y)}\mathcal{D}_0(y \mid \mathbf{x}).$$

So we can use reweighting to get $h_t : \mathcal{X} \to \Delta_K$ as

$$[h_t(\mathbf{x})]_j = \frac{1}{Z(\mathbf{x})}\frac{[\widehat{\boldsymbol{\mu}}_t]_j}{\mathcal{D}_0(y=j)}[h_0(\mathbf{x})]_j, \ \forall j \in [K], \quad (10)$$

where $Z(\mathbf{x}) = \sum_{j=1}^K \frac{[\widehat{\boldsymbol{\mu}}_t]_j}{\mathcal{D}_0(y=j)}[h_0(\mathbf{x})]_j$ is the normalization factor. Then, we predict the class prior $\widehat{\boldsymbol{\mu}}_t$ by Online Newton Step (ONS) (Hazan et al., 2007). More details of OLS are deferred to Appendix C.3.1. Combining our detection framework with reweighting updates, we can obtain a new algorithm, *Wav-R*, which enjoys the following guarantee.

Theorem 4. *The reweighting-based update* (10) *satisfies Requirement 1, and using wavelet-based detection-restart framework (Algorithm 1) with this update as $\mathcal{A}$ ensures*

$$\mathbb{E}\Big[\mathbf{Reg}_T^{\mathbf{d}}(\{R_t, h_t^\star\}_{t=1}^T)\Big] \le \widetilde{\mathcal{O}}\Big(\max\{T^{\frac{k+2}{2k+3}}(P_T^k)^{\frac{1}{2k+3}}, \sqrt{T}\}\Big).$$

where $P_T^k = T^k\|\boldsymbol{D}^{k+1}\boldsymbol{\mu}_{[1,T]}\|_1$ is the k-th order path length.

When $k = 0$, Theorem 4 implies an $\mathcal{O}(T^{\frac{2}{3}}(P_T^0)^{\frac{1}{3}})$ rate, which is optimal for online label shift (Bai et al., 2022; Baby et al., 2023). Crucially, the new algorithm necessitates the maintenance of only a single classifier, leading to a significant improvement in both computational and storage complexities compared to ensemble-based methods. Besides, we also achieve dynamic regret for OLS under cases of higher-order smoothness, which is *new* to the literature.

(ii) Combined with OGD-Update. Apart from the reweighting, we can also combine our detection module with OGD update to handle the OLS problem. Following (Bai et al., 2022), we establish an unbiased risk estimator $\widehat{R}_t$ for the expected risk R_t with unlabeled data S_t and offline data S_0 by the risk rewriting technique. We specify the prediction in the form $h_t(\mathbf{x}) = h(\mathbf{w}_t, \mathbf{x})$ where $h : \mathcal{W} \times \mathcal{X} \to \Delta_K$ is the predictive function, and $\mathbf{w}$ is the model parameter. The loss function $\ell : \Delta^K \times \mathcal{Y} \to \mathbb{R}$ satisfies that $\ell(h(\mathbf{w}, \mathbf{x}), y)$ is convex in $\mathbf{w}$. In the rest, we use $R_t(\mathbf{w})$ to represent the expected risk of the classifier $h(\mathbf{w}, \cdot)$ on the distribution $\mathcal{D}_t$. Then, we have the following decomposition for R_t:

$$R_t(\mathbf{w}) = \sum_{j=1}^{K} [\boldsymbol{\mu}_t]_j \cdot R_t^j(\mathbf{w}) = \sum_{j=1}^{K} [\boldsymbol{\mu}_t]_j \cdot R_0^j(\mathbf{w}),$$

where R_0^j is the initial risk function for the j-th class. So the risk estimator becomes $\widehat{R}_t(\mathbf{w}) = \sum_{j=1}^{K} [\widetilde{\boldsymbol{\mu}}_t]_j \cdot R_0^j(\mathbf{w})$, where $\widetilde{\boldsymbol{\mu}}_t$ is the class prior estimated by BBSE. After that, we update the classifier by OGD

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}[\mathbf{w}_t - \eta_t \nabla \widehat{R}_t(\mathbf{w}_t)], \text{with } \eta_t = 1/\sqrt{t-s}. \quad (11)$$

Combining our detection-restart framework (with the order $k = 0$) with OGD update, we obtain *Wav-O*, which ensures:

Theorem 5. *The OGD-based update update (11) satisfies Requirement 1, and using wavelet-based detection-restart framework (Algorithm 1) with this update as $\mathcal{A}$ ensures*

$$\mathbb{E}\left[\mathbf{Reg}_T^{\mathbf{d}}(\{R_t, h_t^\star\}_{t=1}^T)\right] \le \widetilde{\mathcal{O}}\left(\max\{T^{\frac{2}{3}}(P_T^0)^{\frac{1}{3}}, \sqrt{T}\}\right),$$

where $P_T^0 = \sum_{t=2}^{T} \|\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t-1}\|_1$ is the path length.

Theorem 5 recovers the same optimal dynamic regret guarantee as in (Bai et al., 2022) and (Baby et al., 2023) ignoring the dimension factor. Crucially, our method utilizes a single classifier, offering a significant reduction in computational complexity compared to previous research (Bai et al., 2022). Specifically, the computational complexity decreases from $\mathcal{O}(N_0 d \log T)$ to $\mathcal{O}(N_0 d + \log T)$ each round. In addition, the number of costly projections is reduced from $\mathcal{O}(\log T)$ times to $\mathcal{O}(1)$. This is because the previous study (Bai et al., 2022) requires the computation of the gradient $\nabla \widehat{R}_t(\cdot)$ and the projection onto the feasible domain for $\mathcal{O}(\log T)$ base learners, while our algorithm maintains just one.

Remark 6 (broader applications)**.** Our wavelet-based detection-restart framework is very general and can be useful for broader online distribution shift adaptation problems, such as online label shift with new classes (Qian et al., 2023) and online covariate shift (Zhang et al., 2023a). More results will be included in the extended version.

5. Experiments

To further validate the effectiveness and efficiency of our proposal, we conduct experiments to evaluate over synthetic, benchmark, and real-world datasets across two scenarios: (i) general OCO scenario, and (ii) OLS problem. We compare with baselines and previous state-of-the-art methods.

Table 1: Avg error (%) of different algorithms in the general OCO scenarios. The best are emphasized in bold.

		`Linear`	`Square`	`Sine`
FIX		7.87±0.03	7.98±0.04	7.34±0.03
OGD		5.35±0.02	6.10±0.03	6.37±0.01
Reweight		6.08±0.01	6.45±0.02	6.87±0.02
FTFWH		5.27±0.02	6.52±0.01	6.36±0.02
ATLAS		5.44±0.02	5.65±0.03	**5.75±0.01**
Effi-Dyn		5.30±0.02	5.83±0.01	5.88±0.02
FLH-FTL		5.28±0.03	5.64±0.02	5.85±0.01
Wav-O	$k = 0$	5.30±0.01	5.67±0.03	6.21±0.02
	$k = 1$	**5.25±0.01**	5.61±0.02	6.39±0.03
	$k = 2$	5.47±0.03	5.58±0.03	5.92±0.02
Wav-R	$k = 0$	5.46±0.02	5.55±0.02	5.97±0.03
	$k = 1$	5.37±0.01	5.61±0.02	6.01±0.01
	$k = 2$	5.49±0.02	**5.52±0.04**	**5.75±0.02**

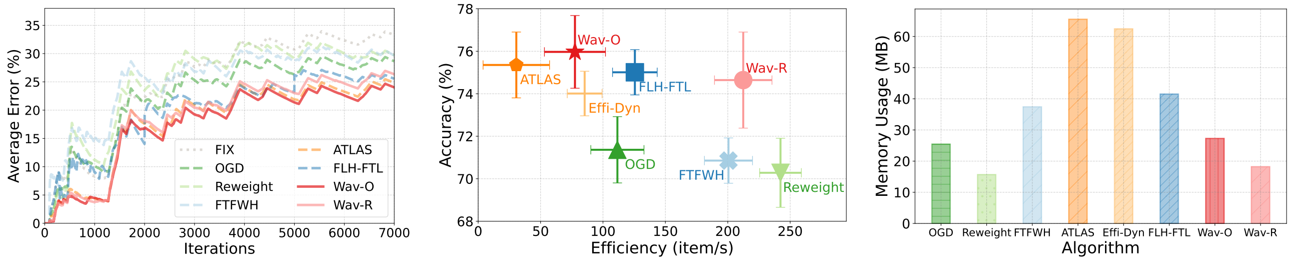
(i) Evaluation on General OCO Scenario. For the OCO scenario, we generate a changing comparator sequence $\{\mathring{\mathbf{u}}_t\}_{t=1}^T$, which represents the optimal decision for the underlying distribution at each round. However, the learner can only observe the empirical estimation $\{\widetilde{\mathbf{u}}_t\}_{t=1}^T$. We simulate three types of comparator sequences for the synthetic OCO data, including `Linear Shift`, `Square Shift`, and `Sine Shift` sequence. As shown in Table 1, our detection-restart framework achieves remarkable performances compared with previous state-of-the-art algorithms.

(ii) Evaluation on Online Label Shift. Table 2 presents the empirical results of various algorithms on the benchmark datasets. Here we consider two types of label distribution changes: the `Linear Shift` and the `Square Shift`. Our algorithms *Wav-O* and *Wav-R* outperform the baseline algorithms *OGD* and *Reweight*, respectively. This superior performance can be mainly attributed to our detection framework's ability to adaptively restart the classifier upon detecting environmental changes based on wavelets. Consequently, this enables the online algorithm to operate within stationary environments, leading to an enhancement in performance especially under higher-order label distribution shifts. Furthermore, our algorithms are highly competitive with previous online-ensemble-based methods *FTFWH*

We also conduct validations on real-world applications. The results on the locomotion dataset (Gjoreski et al., 2018) of OLS are presented in Figure 2(a). Combining our detection-restart framework with OGD and reweighting yields two new algorithms, *Wav-O* and *Wav-R*. It can be seen that they exhibit improvements over baselines *OGD* and *Reweight*. The prior arts for the OLS problem are *ATLAS* and *FLH-FTL* with ensemble structures, yet our algorithms are highly

Table 2: Average error (%) of different algorithms on five benchmark datasets of online label shift scenario, where *Wav-O* represents using our wavelet-based detection-restart framework with OGD, and *Wav-R* represents using our framework with reweighting. We take the order of wavelets as $k = 1$ for the `Linear Shift` and $k = 2$ for the `Square Shift`. We report the mean and standard deviation over five runs. The best algorithms are emphasized in bold. "∘" indicates the algorithm is significantly inferior to our algorithms by paired t-test at a 5% significance level. The online sample size is set as $N_t = 10$.

	`Linear Shift`							`Square Shift`						
	FIX	**OGD**	**Reweight**	**FTFWH**	**ATLAS**	**Wav-O**	**Wav-R**	**FIX**	**OGD**	**Reweight**	**FTFWH**	**ATLAS**	**Wav-O**	**Wav-R**
CIFAR10	∘20.89 ±0.13	15.92 ±0.13	∘18.25 ±0.45	∘ 17.32 ±0.15	15.75 ±0.12	**15.52** ±0.15	15.68 ±0.14	∘20.79 ±0.04	∘16.31 ±0.13	∘17.38 ±0.15	∘16.70 ±0.14	∘15.21 ±0.08	**14.72** ±0.11	15.55 ±0.13
CINIC10	∘34.56 ±0.24	∘27.31 ±0.21	∘32.42 ±2.55	∘ 28.55 ±0.12	∘26.44 ±0.21	**26.11** ±0.13	28.21 ±0.11	∘34.01 ±0.12	∘28.96 ±0.10	∘28.62 ±0.13	∘28.01 ±0.05	∘27.01 ±0.11	**26.12** ±0.05	26.65 ±0.13
EuroSAT	∘15.42 ±0.12	9.13 ±0.15	∘12.34 ±3.17	∘ 11.35 ±0.12	7.21 ±0.13	**7.15** ±0.12	7.25 ±0.12	∘14.19 ±0.15	7.33 ±0.11	∘ 8.88 ±0.09	∘10.19 ±0.12	**6.99** ±0.09	7.43 ±0.05	7.82 ±0.03
Fashion	∘11.35 ±0.05	7.98 ±0.06	8.15 ±0.05	**7.84** ±0.08	8.39 ±0.09	8.34 ±0.13	8.37 ±0.08	∘11.94 ±0.13	∘ 8.46 ±0.07	∘ 8.67 ±0.11	∘ 8.28 ±0.13	∘ 8.13 ±0.12	7.88 ±0.07	**7.32** ±0.07
MNIST	∘ 1.72 ±0.03	1.13 ±0.03	∘ 1.32 ±0.04	∘ 1.25 ±0.03	**1.07** ±0.05	1.09 ±0.02	1.13 ±0.03	∘ 1.83 ±0.08	∘ 1.17 ±0.07	∘ 1.36 ±0.08	∘ 1.17 ±0.07	1.05 ±0.02	1.12 ±0.04	**1.03** ±0.03



(a) error curve

(b) efficiency comparison

(c) storage comparison

Figure 2: (a) Comparison of overall performance on the real-world *online label shift application* with locomotion dataset (Gjoreski et al., 2018). We take the order $k = 0$ for our *Wav-O* and *Wav-R*. (b) Comparison of accuracy and efficiency (with mean and standard deviation over five runs). The closer to the top-right corner, the better the algorithm. (c) Comparison of the storage usage.

competitive even with a single layer. Compared to the previous state-of-the-art two-layer algorithm *Effi-Dyn*, we also achieve a better efficiency with one model alone.

Importantly, our algorithms exhibit significant superiority over the previous online-ensemble method, i.e., *ATLAS*, in both running time and storage usage, as shown in Figures 2(b) and 2(c). Specifically, our algorithms can process nearly five times more items per second while consuming approximately half the storage compared to *ATLAS*. Compared to the previous state-of-the-art two-layer algorithm *Effi-Dyn* which reduces the number of gradient queries and projections per round from $\mathcal{O}(\log T)$ to 1, we also achieve a better computational and storage cost with only a single-layer structure. This validates that our detection-restart framework is more efficient than ensemble-based methods.

6. Conclusion

In this paper, we introduced the *underlying dynamic regret* as the performance measure for non-stationary online learning. While being a particular form of the general dynamic regret notion, it is sufficient to encompass many real-world applications. To optimize this measure, we devised a novel method based on the adaptive restart strategy, equipped with an efficient wavelet-based detection to handle non-stationarity. This non-ensemble method provably achieves an almost optimal dynamic regret and demonstrates flexibility in handling higher-order smoothness in online data. As the main application, we applied the general framework to the problem of online label shift, leading to several new algorithms with optimal dynamic regret guarantees. They showed significant efficiency in computational and storage complexities compared to previous arts achieved by ensemble structures. Experiments further support our findings.

It is worth noting that both our wavelet-based detection method and previous model ensemble-based methods contain a computational complexity of $\mathcal{O}(\log T)$. But our method is more efficient in many cases, as it only maintains multiple wavelet coefficients instead of multiple model parameters. Indeed, to handle the non-stationarity, our method can be considered as an ensemble of multiple wavelet bases. This raises a question about the necessity of an additional computational overhead of $\mathcal{O}(\log T)$ compared to stationary algorithms, when handling non-stationary online environments with inherent uncertainty. Such a "lower-bound" argument would be an interesting future work to examine.

Acknowledgements

Yu-Yang Qian, Peng Zhao, and Zhi-Hua Zhou were supported by National Science and Technology Major Project (2022ZD0114802) and JiangsuSF (BK20220776). Yu-Jie Zhang was supported by Todai Fellowship and the Institute for AI and Beyond, UTokyo. Masashi Sugiyama was supported by Institute for AI and Beyond, UTokyo, and a grant from Apple, Inc. Any views, opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and should not be interpreted as reflecting the views, policies or position, either expressed or implied, of Apple Inc.

Impact Statement

This paper proposed a novel online learning method based on the adaptive restart strategy, equipped with an efficient wavelet-based detection to handle non-stationary online learning. This work advances the field of Online Learning. There are many potential societal consequences, none of which we feel must be specifically highlighted here.

References

Azoury, K. S. and Warmuth, M. K. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine learning*, 43:211–246, 2001.

Baby, D. and Wang, Y.-X. Online forecasting of total-variation-bounded sequences. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pp. 11069–11079, 2019.

Baby, D. and Wang, Y.-X. Adaptive online estimation of piecewise polynomial trends. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 20462–20472, 2020.

Baby, D. and Wang, Y.-X. Optimal dynamic regret in exp-concave online learning. In *Proceedings of the 34th Annual Conference on Computational Learning Theory (COLT)*, pp. 359–409, 2021.

Baby, D. and Wang, Y.-X. Second order path variationals in non-stationary online learning. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 9024–9075, 2023.

Baby, D., Zhao, X., and Wang, Y.-X. An optimal reduction of TV-denoising to adaptive online learning. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 2899–2907, 2021.

Baby, D., Garg, S., Yen, T.-C., Balakrishnan, S., Lipton, Z. C., and Wang, Y.-X. Online label shift: Optimal dynamic regret meets practical algorithms. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 65703–65742, 2023.

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

Besbes, O., Gur, Y., and Zeevi, A. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, 2015.

Beylkin, G., Coifman, R., and Rokhlin, V. Fast wavelet transforms and numerical algorithms. *Communications on Pure and Applied Mathematics*, 44(2):141–183, 1991.

Boudiaf, M., Müller, R., Ayed, I. B., and Bertinetto, L. Parameter-free online test-time adaptation. In *Proceedings of the 35th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8334–8343, 2022.

Chen, S., Tu, W.-W., Zhao, P., and Zhang, L. Optimistic online mirror descent for bridging stochastic and adversarial online convex optimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 5002–5035, 2023.

Chen, Y., Luo, H., Ma, T., and Zhang, C. Active online domain adaptation. *ICML Workshop on Lifelong ML*, 2020.

Cohen, A., Daubechies, I., Jawerth, B., and Vial, P. Multiresolution analysis, wavelets and fast algorithms on an interval. *Comptes rendus de l'Académie des sciences. Série 1, Mathématique*, 316(5):417–421, 1993a.

Cohen, A., Daubechies, I., and Vial, P. Wavelets on the interval and fast wavelet transforms. *Applied and Computational Harmonic Analysis*, 1(1):54–81, 1993b.

Cutkosky, A. Parameter-free, dynamic, and strongly-adaptive online learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 2250–2259, 2020.

Darlow, L. N., Crowley, E. J., Antoniou, A., and Storkey, A. J. Cinic-10 is not imagenet or cifar-10. *ArXiv preprint*, arXiv:1810.03505, 2018.

Daubechies, I. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, 1988.

Daubechies, I. *Ten Lectures on Wavelets*. SIAM, 1992.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database.

In *Proceedings of the 22nd IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Donoho, D. L. and Johnstone, I. M. Minimax estimation via wavelet shrinkage. *Annals of Statistics*, 26(3):879–921, 1998.

du Plessis, M. C. and Sugiyama, M. Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks*, 50:110–119, 2014.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Garg, S., Wu, Y., Balakrishnan, S., and Lipton, Z. C. A unified view of label shift estimation. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 3290–3300, 2020.

Gjoreski, H., Ciliberto, M., Wang, L., Morales, F. J. O., Mekki, S., Valentin, S., and Roggen, D. The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices. *IEEE Access*, 6:42592–42604, 2018.

Haar, A. *Zur Theorie Der Orthogonalen Funktionensysteme*. Georg August Universitat, 1909.

Hazan, E. Introduction to Online Convex Optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Helber, P., Bischke, B., Dengel, A., and Borth, D. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *Proceedings of the 17th IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 204–207, 2018.

Jadbabaie, A., Rakhlin, A., Shahrampour, S., and Sridharan, K. Online optimization : Competing with dynamic comparators. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 398–406, 2015.

Jain, V. and Learned-Miller, E. Online domain adaptation of a pre-trained cascade of classifiers. In *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 577–584, 2011.

Jézéquel, R., Gaillard, P., and Rudi, A. Efficient improper learning for online logistic regression. In *Proceedings of the 33th Annual Conference on Computational Learning Theory (COLT)*, volume 125, pp. 2085–2108, 2020.

Johnstone, I. M. *Gaussian Estimation: Sequence and Wavelet models*. 2017.

Krizhevsky, A., Hinton, G., et al. *Learning Multiple Layers of Features from Tiny Images*. Toronto, ON, Canada, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of IEEE*, 86(11):2278–2324, 1998.

Lim, H., Kim, B., Choo, J., and Choi, S. TTN: A domain-shift aware batch normalization in test-time adaptation. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2020.

Lipton, Z. C., Wang, Y.-X., and Smola, A. J. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 3128–3136, 2018.

Luo, H., Agarwal, A., Cesa-Bianchi, N., and Langford, J. Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pp. 902–910, 2016.

Mallat, S. *A Wavelet Tour of Signal Processing*. Elsevier, 1999.

Moon, J. H., Das, D., and Lee, C. S. G. Multi-step online unsupervised domain adaptation. In *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 41172–41576, 2020.

Nemirovskij, A. S. and Yudin, D. B. *Problem Complexity and Method Efficiency in Optimization*. A Wiley-Interscience Publication, 1983.

Nguyen, T. D., du Plessis, M. C., and Sugiyama, M. Continuous target shift adaptation in supervised learning. In *Proceedings of the 7th Asian Conference on Machine Learning (ACML)*, pp. 285–300, 2015.

Qian, Y.-Y., Bai, Y., Zhang, Z.-Y., Zhao, P., and Zhou, Z.-H. Handling new class in online label shift. In *Proceedings of the 23rd IEEE International Conference on Data Mining (ICDM)*, pp. 1283–1288, 2023.

Saerens, M., Latinne, P., and Decaestecker, C. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002.

Sugiyama, M., Suzuki, T., and Kanamori, T. Density-ratio matching under the bregman divergence: A unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64:1009–1044, 2012.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 9229–9248, 2020.

Tibshirani, R. J. Adaptive piecewise polynomial estimation via trend filtering. *Annals of Statistics*, 42(1):285–323, 2014.

Wu, R., Guo, C., Su, Y., and Weinberger, K. Q. Online adaptation to label distribution shift. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pp. 11340–11351, 2021.

Wu, R., Datta, S., Su, Y., Baby, D., Wang, Y.-X., and Weinberger, K. Q. Online feature updates improve online (generalized) label shift adaptation. *NeurIPS Workshop on Self-Supervised Learning: Theory and Practice*, 2024.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *ArXiv preprint*, arXiv:1708.07747, 2017.

Yang, T., Zhang, L., Jin, R., and Yi, J. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 449–457, 2016.

Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 819–827, 2013.

Zhang, L. Online learning in changing environments. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 5178–5182, 2020.

Zhang, L., Lu, S., and Zhou, Z.-H. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pp. 1330—-1340, 2018.

Zhang, L., Jiang, W., Yi, J., and Yang, T. Smoothed online convex optimization based on discounted-normal-predictor. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pp. 4928–4942, 2022.

Zhang, Y.-J., Zhao, P., and Zhou, Z.-H. A simple online algorithm for competing with dynamic comparators. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 390–399, 2020.

Zhang, Y.-J., Zhang, Z.-Y., Zhao, P., and Sugiyama, M. Adapting to continuous covariate shift via online density ratio estimation. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 29074–29113, 2023a.

Zhang, Z., Cutkosky, A., and Paschalidis, I. C. Unconstrained dynamic regret via sparse coding. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 74636–74670, 2023b.

Zhao, P. and Zhang, L. Improved analysis for dynamic regret of strongly convex and smooth functions. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control (L4DC)*, pp. 48–59, 2021.

Zhao, P., Zhang, Y.-J., Zhang, L., and Zhou, Z.-H. Dynamic regret of convex and smooth functions. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 12510–12520, 2020.

Zhao, P., Xie, Y.-F., Zhang, L., and Zhou, Z.-H. Efficient methods for non-stationary online learning. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pp. 11573–11585, 2022.

Zhao, P., Zhang, Y.-J., Zhang, L., and Zhou, Z.-H. Adaptivity and non-stationarity: Problem-dependent dynamic regret for online convex optimization. *Journal of Machine Learning Research*, 25(98):1–52, 2024.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pp. 928–936, 2003.

A. Additional Experiments

In this section, we present empirical results for the OCO scenario, and online label shift scenario, respectively.

A.1. Evaluation on General Online Convex Optimization Scenario

This subsection evaluates our algorithms in the online convex optimization scenario, with a particular emphasis on the effectiveness of our detection framework and its flexibility in accommodating the higher-order path length. We begin with an introduction of the experiment setup, followed by the contenders and the performance comparison in the OCO scenario.

Experiment Setup. For the OCO scenario, we generate a changing comparator sequence $\{\mathring{\mathbf{u}}_t\}_{t=1}^T$, which represents the optimal decision for the underlying distribution at each round. However, the learner can only observe the empirical estimation $\{\widetilde{\mathbf{u}}_t\}_{t=1}^T$. We simulate three types of comparator sequences for the synthetic OCO data, including the `Linear Shift` sequence, representing a gradual change in the underlying comparator following a linear pattern; the `Square Shift` sequence, where comparators fluctuate according to a quadratic pattern; and the `Sine Shift` sequence, where the underlying comparator periodically changes following a sinusoidal pattern. The data dimension is set as 12, and we repeat all experiments for five times and evaluate the contenders by the average over $T = 10,000$ rounds.

Contenders. We evaluate our algorithms against the following competitors.

- *FIX* predicts with the fixed initial classifier without any online updates.
- *OGD* (Zinkevich, 2003) is the online gradient descent (11) with the step size $\eta_t = 1/\sqrt{t}$ through the entire time horizon without restarting.
- *Reweight* Wu et al. (2021) is the reweighting update mechanism, which reweights the initial classifier following (10) through the entire time horizon without restarting.
- *FTFWH* (Wu et al., 2021) is short for Follow The Fixed Window History, which averages across previously estimated priors within a sliding window. In all experiments, we set the sliding window length as 100.
- *ATLAS* (Bai et al., 2022) is an online ensemble method, which maintains $\mathcal{O}(\log T)$ base learners, each performing online gradient descent with different step sizes and then uses the Hedge algorithm (Freund & Schapire, 1997) as the meta learner to combine the outputs.
- *Effi-Dyn* (Zhao et al., 2022) is the state-of-the-art two-layer ensemble algorithm that constructs a surrogate loss and domain, significantly reducing the number of projections and gradient queries from $\mathcal{O}(\log T)$ to only 1 per round. By integrating it with the OGD update mechanism, it becomes an effective version of *ATLAS*.
- *FLH-FTL* (Baby et al., 2023) initially transforms the adaptation problem into an online regression problem. Subsequently, it utilizes an ensemble-based approach that reweights the initial classifier to adapt to new environmental shifts.

Performance Analysis. As demonstrated in Table 1, we evaluate the performance of our algorithms (*Wav-O* and *Wav-R*) against other algorithms in the OCO setting with varying underlying comparator sequences $\{\mathring{\mathbf{u}}_t\}_{t=1}^T$. It should be noticed that our algorithms with $k = 1$ yield the best performance for the `Linear Shift` sequence, where k represents the order of the wavelet detection. Likewise, for the `Square Shift` sequence, our detection-restart framework with $k = 2$ has the best performance, which highlights that our detection-restart framework is flexible enough to capture the higher-order smoothness in online data. Particularly, under the higher-order environmental changes, our detection-restart framework achieves a more significant performance gain compared to the competitors. These results further demonstrate our detection-restart framework can accommodate the higher-order path length of the underlying comparator sequence.

A.2. Implementation Details

For the OCO scenario and the benchmark datasets in the OLS scenario, we simulate three types of environmental change patterns to encompass various non-stationary environments. For each case, the current distribution at round t is a mixture of two different constant distributions, i.e., $\boldsymbol{\mu}, \boldsymbol{\mu}' \in \Delta_K$ with a time-varying coefficient α_t, i.e., $\boldsymbol{\mu}_t = (1 - \alpha_t)\boldsymbol{\mu} + \alpha_t \boldsymbol{\mu}'$, where $\boldsymbol{\mu}_t$ denotes the current distribution at round t and α_t controls the non-stationarity and patterns. Specifically,

- `Linear Shift`: the parameter $\alpha_t = \frac{t}{T}$, which represents the gradual environmental change following a linear pattern.
- `Square Shift`: α_t switches between 1 and 0 following a quadratic pattern $\alpha_t = \sqrt{t/T}$.
- `Sine Shift`: $\alpha_t = \sin \frac{i\pi}{L}$ periodically changes following a sinusoidal pattern, where $i = t \bmod L$ and L signifies a given periodic length. By default, we set $L = \Theta(\sqrt{T})$ in the experiments.

Table 3: Average error (%) of different algorithms on the real-world *online label shift application*: locomotion dataset (Gjoreski et al., 2018), where *Wav-O* represents using our wavelet-based detection-restart framework with OGD, and *Wav-R* represents using our framework with reweighting. We report the mean and standard deviation over five runs.

	FIX	OGD	Reweight	FTFWH	ATLAS	Effi-Dyn	FLH-FTL	LAME	Wav-O			Wav-R		
									$k=0$	$k=1$	$k=2$	$k=0$	$k=1$	$k=2$
Locomotion	33.32 ±1.05	28.64 ± 1.67	29.72 ± 1.57	29.14 ± 1.12	24.64 ± 1.55	25.45 ± 2.35	24.83 ± 1.45	24.92 ± 1.23	24.03 ± 1.71	24.12 ± 1.54	**23.99** ± **1.65**	25.35 ± 2.23	25.41 ± 2.35	25.36 ± 2.23

We repeat all experiments five times to evaluate average error and standard deviations. Learning rates of algorithms are set according to theoretical guidelines. All experiments are executed on a computer with 2 CPUs, each having 32 cores.

A.3. Evaluation on Online Label Shift

In this part, we provide the details of the experiments in OLS. We compare the same seven contenders outlined in Appendix A.1, adding an additional contender: *LAME* (Boudiaf et al., 2022), which is a test-time adaptation (TTA) approach.

Datasets. We use various datasets to evaluate our algorithms in the context of the online label shift scenario, including five benchmark datasets and a real-world application: SHL dataset (Gjoreski et al., 2018) which aims to detect the locomotion of an object. The benchmark datasets are outlined as follows.

- **CIFAR10** (Krizhevsky et al., 2009): A classification dataset comprises 60, 000 color images distributed across ten classes: airplane, automobile, ship, truck, bird, cat, deer, dog, frog, and horse.
- **CINIC10** (Darlow et al., 2018): A tiny version of ImageNet (Deng et al., 2009) dataset, it contains images from both CIFAR10 and ImageNet and shares the same ten classes as CIFAR10.
- **EuroSAT** (Helber et al., 2018): A land cover classification dataset, EuroSAT includes 27, 000 satellite images from over 30 different European countries. The images span ten different categories: industrial, residential, annual crop, permanent crop, river, sea and lake, herbaceous vegetation, highway, pasture, and forest.
- **Fashion** (Xiao et al., 2017): A dataset includes 70, 000 grayscale fashion images divided among ten different classes: T-shirt, trouser, shirt and sneaker, pullover, dress, coat, sandal, bag, and ankle boot.
- **MNIST** (LeCun et al., 1998): A widely-used image dataset of handwritten digits, consisting of 70, 000 grayscale images of handwritten digits across ten different classes.

For the above five benchmark datasets, we utilize a finetuned ResNet34 (He et al., 2016) to extract image features. The images used to train the ResNet34 do not overlap with either the offline or online datasets. Besides, we also compare different algorithms on a real-world locomotion dataset:

- **Locomotion** (Gjoreski et al., 2018): A dataset that aims to distinguish the human locomotion in real-life. It comprises multi-modal sensor data (e.g., acceleration, gyroscope, magnetometer, orientation, gravity, pressure, altitude, and temperature) from a body-worn camera and four smartphones, carried simultaneously at typical body locations, along with corresponding human motion data and timestamps. We sample 30, 000 offline and 77, 000 online data samples from an 11-day period, which cover six classes: still, walking, running, bike, car, and bus. During the online update, the samples arrive in real chronological order, according to the timestamp. It is important to note that in this dataset, the distribution of human motion types changes over time, leading to the label shift occurring in the data stream.

Table 4: Average error (%) with different online sample size N_t. We take $N_t = 5, 10$ and 20 respectively.

CIFAR 10	$N_t = 5$	$N_t = 10$	$N_t = 20$
Wav-O	14.79 ± 0.15	14.72 ± 0.11	14.69 ± 0.10
Wav-R	15.54 ± 0.18	15.55 ± 0.13	15.32 ± 0.11

Sensitivity to Online Sample Size. We also analyze the sensitivity of our algorithm to the online sample size N_t. Specifically, we modify the online sample size to various values (i.e., 5, 10, and 20) for the benchmark dataset CIFAR10 under the `Square Shift`. This aims to evaluate how our method will perform under conditions where the empirical observation $\widetilde{\mathbf{u}}_t$ is less accurately estimated due to a reduced sample size, therefore increasing the variance and potentially violating Assumption 1. The results are listed as Table 4, showing that our algorithms do not exhibit a significant performance drop across different sample sizes. These results further indicate that our algorithms are flexible enough and can accommodate various scenarios, even if some assumptions are not satisfied.

Sensitivity Analysis of the order k. As detailed in Section 3.3, our wavelet detection-based method is flexible enough to capture *higher-order smoothness* in online data, i.e., it can accommodate complex non-stationary patterns beyond simple linear gradual changes. To examine the sensitivity of the order parameter, we conduct experiments with different orders of wavelet detection for our algorithms. Specifically, we employ $k = 0$, $k = 1$, and $k = 2$ orders of wavelet detection and evaluate their performance using the real-world locomotion dataset. As exhibited in Table 3, the overall performance among different orders does not exhibit substantial variance, suggesting that our algorithms are not particularly sensitive to the order parameter k. Therefore, the choice of order k is not sensitive and will not significantly impact the final results.

A.4. Wavelet Speedup

We also conduct modular analysis to evaluate the efficiency of our streaming wavelet operator. As demonstrated in Figure 3, the streaming wavelet operator is more efficient than the traditional matrix computation, processing more items per second in a data stream. This is because the operator only lazily updates a portion of wavelet coefficients using a binary index tree, while the traditional matrix computation needs to recompute all the wavelet coefficients at each round. When compared with the PyWT,[2] an open-source Python wavelet transformation package, our operator again demonstrates superior efficiency due to its lazy update property and parallelism. Therefore, the streaming wavelet operator is more suitable for the online learning scenario, where the online update is necessary.

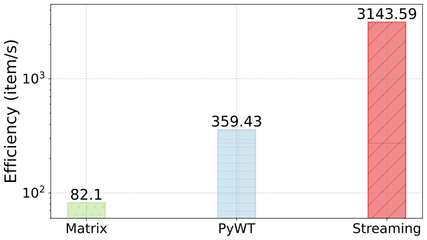


Figure 3: Efficiency comparison of the streaming wavelet operator with traditional matrix multiplication and the PyWT package.

B. Related Work

This section introduces related works to our paper, including the non-stationary online learning in Appendix B.1, the online trend filtering in Appendix B.2, and the online distribution shift in Appendix B.3.

B.1. Dynamic Regret Minimization

In this part, we briefly review the related works in dynamic regret minimization for online convex optimization. For a more comprehensive treatment, one may refer the latest work in (Zhao et al., 2024, Section 2.2) and (Zhang et al., 2023a, Appendix B). The classical performance measure for the OCO problem is the *static regret*. However, this measure may be overly optimistic and is not suitable in changing environments, where the optimal decision changes over time. To address this limitation, the paradigm of non-stationary online learning has been developed, which has seen significant progress over the years (Zinkevich, 2003; Besbes et al., 2015; Yang et al., 2016; Zhao & Zhang, 2021).

Worst-case Dynamic Regret. The worst-case dynamic regret (Jadbabaie et al., 2015; Yang et al., 2016; Zhang et al., 2020) aims to minimize the cumulative difference between the decision $\boldsymbol{\theta}_t$ and the minimizer $\boldsymbol{\theta}_t^\star$, defined in (3) and restated below

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \boldsymbol{\theta}_t^\star\}_{t=1}^T) = \sum_{t=1}^T f_t(\boldsymbol{\theta}_t) - \sum_{t=1}^T f_t(\boldsymbol{\theta}_t^\star),$$

where $\boldsymbol{\theta}_t^\star \in \arg\min_{\boldsymbol{\theta}\in\Theta} f_t(\boldsymbol{\theta})$ is the function minimizer at each round. However, as mentioned in Section 2, minimizing the worst-case dynamic regret can lead to severe *overfitting* to the sample randomness.

Universal Dynamic Regret. A more appropriate performance measure is the *universal dynamic regret* (Zinkevich, 2003), in the sense that it gives a universal guarantee that holds against *arbitrary* comparator sequence. Universal dynamic regret compares $\boldsymbol{\theta}_t$ with an arbitrary time-varying comparator sequence $\{\mathbf{u}_t\}_{t=1}^T$, formally defined in (1) and restated below

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathbf{u}_t\}_{t=1}^T) = \sum_{t=1}^T f_t(\boldsymbol{\theta}_t) - \sum_{t=1}^T f_t(\mathbf{u}_t).$$

This measure can recover the aforementioned worst-case dynamic regret (3) with comparators $\mathbf{u}_t = \boldsymbol{\theta}_t^\star \in \arg\min_{\boldsymbol{\theta}\in\Theta} f_t(\boldsymbol{\theta})$. In contrast to the worst-case dynamic regret, optimizing the universal dynamic regret is more reasonable, as it supports comparison to any feasible comparator sequence, hence including the one with $\boldsymbol{\theta}_t^\dagger \in \arg\min_{\boldsymbol{\theta}\in\Theta} F_t(\boldsymbol{\theta})$, the best feasible solution tailored to the underlying distribution.

[2] https://pypi.org/project/PyWavelets/

However, it is usually challenging to optimize the universal dynamic regret. The fundamental difficulty arises from the unknown level of environmental non-stationarity, which is manifested as the uncertainty of arbitrary comparators in the regret measure (1). To this end, recent works typically employ a two-layer *online ensemble* framework (Zhao et al., 2024) to optimize the measure, which maintains diverse multiple base learners and uses a meta learner to combine them to track the best one on the fly. For instance, Zhang et al. (2018) achieved the optimal dynamic regret of the convex function utilizing the following ensemble structure: the algorithm maintains $\mathcal{O}(\log T)$ base learners, each performing online gradient descent with varying step sizes; and then use the Hedge algorithm as the meta-learner, combining the outputs of the base learners. Cutkosky (2020) and Zhang et al. (2022) employed a sequential ensemble for non-stationary online learning, in which a series of base learners were sequentially ensembled, and their outputs were added up to obtain the final prediction. Subsequent works are devoted to achieving improved guarantees when the online functions are equipped with better properties such as smoothness (Zhao et al., 2020; 2024) and strong convexity or exponential concavity (Baby & Wang, 2021; 2023). Notably, albeit achieving favorable regret guarantees, the ensemble structure brings an evident computational overhead, especially when T is large. There are efforts devoted to improving the efficiency of model ensemble algorithms by reducing the per-round required number of gradient calculations or projections onto the feasible domain (Zhao et al., 2022). However, it remains unclear how to attain an *optimal* dynamic regret *without* an ensemble over multiple base learners.

Underlying Dynamic Regret. We investigate a special form of universal dynamic regret, defined in (4) and restated below:

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) = \sum_{t=1}^T f_t(\boldsymbol{\theta}_t) - \sum_{t=1}^T f_t(\mathring{\mathbf{u}}_t),$$

where $\mathring{\mathbf{u}}_t \in \Theta$ is the ground-truth comparator characterizing the *underlying distribution* at round t. We refer to this measure as *underlying dynamic regret*. Crucially, the ground-truth comparator is *unknown* to the learner, but she can access an empirical estimation $\widetilde{\mathbf{u}}_t$ with unbiasedness ($\mathbb{E}[\widetilde{\mathbf{u}}_t] = \mathring{\mathbf{u}}_t$) and bounded variance ($\mathbb{V}[\widetilde{\mathbf{u}}_t] \leq \sigma^2$) after making the prediction, as formulated in Assumption 1. In many applications of interest, we can construct an appropriate empirical estimation $\widetilde{\mathbf{u}}_t$ at each iteration to satisfy this assumption. We take OLS as a concrete example as verified in Section 4.2. Besides, Assumption 1 posits that the variance σ^2 is both bounded and known to the learner. Indeed, a bounded domain can ensure this, i.e., we have $\mathbb{V}[\widetilde{\mathbf{u}}_t] = \frac{1}{d}\|\widetilde{\mathbf{u}}_t - \mathring{\mathbf{u}}_t\|_2^2 \leq \Gamma^2/d$ where $\Gamma \triangleq \sup_{\boldsymbol{\theta},\boldsymbol{\theta}' \in \Theta}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2$ denotes the diameter of feasible domain.

Note that the underlying dynamic regret is positioned between the universal and worst-case dynamic regret. For example, in online supervised learning, the worst-case dynamic regret tracks the sequence $\{\boldsymbol{\theta}_t^\star\}_{t=1}^T$, the universal dynamic regret covers all possible sequences $\{\mathbf{u}_t\}_{t=1}^T$, while our proposed underlying dynamic regret only competes with the sequence $\{\boldsymbol{\theta}_t^\dagger\}_{t=1}^T$, where $\boldsymbol{\theta}_t^\dagger \in \arg\min_{\boldsymbol{\theta}\in\Theta} F_t(\boldsymbol{\theta})$ is the best feasible solution tailored to the underlying distribution. By taking expectation, we can easily achieve the dynamic regret guarantee on the expected function F_t: $\mathbb{E}[\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \boldsymbol{\theta}_t^\dagger\}_{t=1}^T)] = \mathbf{Reg}_T^{\mathbf{d}}(\{F_t, \boldsymbol{\theta}_t^\dagger\}_{t=1}^T) \triangleq \sum_{t=1}^T F_t(\boldsymbol{\theta}_t) - \sum_{t=1}^T F_t(\boldsymbol{\theta}_t^\dagger)$ if $\boldsymbol{\theta}_t$ is independent of f_t.

In our current framework, wavelet detection and model updating are conducted *after* making predictions, indicating that we do not utilize the current observed $\widetilde{\mathbf{u}}_t$ for the current model $\boldsymbol{\theta}_t$. It is noteworthy that for the detection, incorporating the latest data sample into the wavelet coefficients, which may contain noise due to sample randomness, could potentially compromise the unbiased nature of the detection process, thereby diminishing its effectiveness. Regarding model updating, we remark that by employing advanced improper online learning algorithms such as the Vovk-Azoury-Warmuth (VAW) forecaster (Azoury & Warmuth, 2001; Jézéquel et al., 2020), it might be possible to further improve the empirical behavior and theoretical guarantee (mainly for the constant dependency) for exponential concave functions and squared loss functions.

B.2. Online Trend Filtering

Another line of work considers the online trend filtering problem (Baby & Wang, 2019; 2020; Baby et al., 2021), which extends the classical offline trend filtering problem (Donoho & Johnstone, 1998; Tibshirani, 2014) to an online version. Specifically, in online trend filtering, the environment selects a sequence of underlying ground truth samples $\mathring{\mathbf{y}}_1, \ldots, \mathring{\mathbf{y}}_T \in \mathbb{R}$, while the learner only observes a noisy data sample $\widetilde{\mathbf{y}}_t \in \mathbb{R}$ at each round t, where $\widetilde{\mathbf{y}}_t = \mathring{\mathbf{y}}_t + Z$ with Z denoting sub-Gaussian noise with a variance of σ^2. The learner then denoises the observation to obtain her prediction. The goal is to minimize the cumulative squared loss $\sum_{t=1}^T \|\widehat{\mathbf{y}}_t - \mathring{\mathbf{y}}_t\|_2^2$, where $\widehat{\mathbf{y}}_t$ represents the learner's prediction for the t-th round. To this end, Baby & Wang (2019; 2020) explore methods based on wavelets to tackle this problem. The techniques of this paper take great inspiration from online trend filtering, particularly the contributions of Baby & Wang (2019; 2020). In the following, we discuss these line of work in detail, and highlight the key contribution of our work.

Computational Consideration. Baby & Wang (2019) implement the Haar Wavelet detection to achieve first-order smoothness. This methodology enabled them to streamline computations, given that the incremental updates are relatively straightforward for the Haar Wavelet. However, their method remains challenging to extend to handle higher-order smoothness. Zhang et al. (2023b) leverage the local property for the Haar wavelet for an efficient update, but their method does not consider the challenging higher-order cases. Baby & Wang (2020) introduce a method capable of managing higher-order smoothness. Although achieving optimal results for the online trend filtering problem, the method proposed by Baby & Wang (2020) encounters efficiency challenges: in their method, wavelets are updated in a mini-batch style due to the use of traditional wavelet implementations that require storing all elements of a sequence and recomputing all wavelet coefficients upon encountering new elements, which is unsuitable for the online update.

To address this, we carefully design a binary indexed tree to organize the k-th order CDJV wavelet coefficients and a lazy update mechanism, which we term as the *streaming wavelet operator*, as detailed in Section 3.2. The key technical contribution is the removal of the recentering operation and the novel organization of the wavelet coefficients by employing the binary index tree structure, which also performs an "implicit padding" to implicitly complete the sequence as a longer length of 2^n, where n is an integer, thus enabling the efficiently online update of high-order CDJV wavelets in a streaming fashion. Below, we first show that the recentering can be removed for the offline wavelet calculation, and then demonstrate that the implicit padding is effective for the detection module considered in this paper.

- *Removing recentering in convolution.* The recentering operation can be removed in the online operation, as the effect of the un-recentralization of the sequence can be canceled in the convolution operation with CDJV wavelets used in our streaming wavelet operator, that is, sliding the wavelet basis over the data sequence and compute the matrix multiplication at each position. Our Lemma 6 formally states that recenterlization will not affect the calculated wavelet coefficients.
- *Implicit padding.* Our proposed streaming wavelet operator essentially performs an "implicit padding" when encountering a sequence whose length is not a power of 2. Specifically, it omits yet-to-arrive elements in the online sequence by employing the convolution operation. This can be seen as an "implicit padding" to complete the sequence as a longer length of 2^n and let the coefficients of the extra padded element be zero. In Appendix 5 of Baby & Wang (2020), the authors elucidate the limitations of padding, suggesting that it can inflate the path length (aka, the "TV distance" in their work) of a sequence. However, in our work, we adopt an alternative perspective based on the wavelet coefficient analysis. Specifically, we first demonstrate that the F-norm of wavelet coefficients of the recentralized and padded sequence is identical to the F-norm of the unpadded original sequence. Furthermore, as we discussed before, the recentering can be removed. The statement is formally shown in Lemma 5. Consequently, Lemma 5 bridges the gap between the wavelet coefficients and the path length, and our implicit padding approach remains effective in this context, only leading to a minor gap of $\widetilde{\mathcal{O}}(\sqrt{|\mathcal{I}|})$. Still, we emphasize the key insight here is we utilize a binary tree structure to organize the wavelet coefficients, which enables us to perform the implicit padding.

Notably, even in the special case of first-order smoothness (Baby & Wang, 2019), our method also exhibits significant advantages in efficiency, mainly by eliminating their costly recentering and padding operations. Appendix D.4 theoretically illustrates how our streaming wavelet operator can leverage the binary tree structure to efficiently update the wavelet coefficients and enjoys a computational and storage complexity at $\mathcal{O}(\log T)$.

Loss Function. Prior trend filtering works (Baby & Wang, 2019; 2020) mainly focus on squared-loss regression cases where only 1-dimensional variables and squared loss is considered, while we build a wavelet detection-restart framework for the general OCO setting in Section 3.3. Although previous works of Baby & Wang (2019; 2020) can be extended to higher-dimension cases, its application in real-world scenarios, such as online distribution shift adaptation, for designing a detection module might be unsatisfactory. Specifically, Baby & Wang (2020) needs to run d parallel instances to handle a d-dimensional problem. This makes determining the restarting point for the downstream online algorithm very difficult, mainly because of inconsistencies in coefficients' norms across different dimensions. For example, at time t, the norms of coefficients might exceed the threshold in certain dimensions but not in others, making it hard to determine the overall restarting point for the entire algorithm. In contrast, we develop a general wavelet-based detection framework applicable to the broader online convex optimization cases. This requires a refined analysis of the wavelet detection compared to (Baby & Wang, 2019; 2020), including the introduction of the k-th order comparator gap as detailed in Section 3.3.

B.3. Online Distribution Shift Adaptation

The problem of distribution shift has received considerable attention in the offline setting (Saerens et al., 2002; Zhang et al., 2013; du Plessis & Sugiyama, 2014; Nguyen et al., 2015; Lipton et al., 2018; Garg et al., 2020), where testing and training data come from two distinct distributions. Recently, the more challenging setup of *online distribution shift* where distribution shifts as time evolves has attracted increasing attentions. Wu et al. (2021) make the first such attempt for the online label shift (OLS) problem where the label distribution changes over time, in which they constructed an unbiased risk estimator with the unlabeled data for model assessment and employed online gradient descent for model updating and achieve a static regret. However, in non-stationary environments, a fixed comparator can hardly perform well all the time, making the guarantee less attractive for OLS problems. To this end, the prior work of Bai et al. (2022) first introduces the dynamic regret measure for the OLS problem, then they propose an algorithm based on online ensemble structure and achieve a dynamic regret of $\mathcal{O}(T^{2/3}P_T^{1/3})$, where P_T is the path length of the label distribution shift. Specifically, they employ a total of $\mathcal{O}(\log T)$ base learners, each running with a different step size, and employ a meta-learner to combine the outputs of base learners to handle environment drifts.

Many other distribution shift problems have also been studied in the online setting. For example, Qian et al. (2023) investigate the online label shift problem with the existence of new classes, and Zhang et al. (2023a) initialize the study of the online continuous covariate shift problem. Besides, online domain adaptation methods (Jain & Learned-Miller, 2011; Chen et al., 2020) aim to adapt the offline model to align with online target domains, where distribution is different from the offline one, and online unsupervised domain adaptation approaches (Moon et al., 2020) have been proposed to adapt the offline model to target domains without labeled data.

Additionally, test-time adaptation (TTA) methods have been developed to adjust model outputs for online test domains in the absence of labeled test distribution data. These methods typically depend on the semantic content of visual or language data (Sun et al., 2020), or specific structures like batch normalization layers (Lim et al., 2020). Moreover, these methods are too general for the test data shift problems to capture the special structure of our investigated online label shift problem. We present an empirical comparison with a state-of-the-art TTA method, LAME (Boudiaf et al., 2022), which adapts the output of the model by employing a Laplacian regularization as a corrective term, as shown in Appendix A.3. Finally, while TTA methods are effective in refining model outputs, our primary focus is on model updating, and incorporating these TTA methods as plug-in modules alongside our algorithms can further enhance the overall performance.

C. Background: Smoothness, Wavelet Analysis, and Online Distribution Shift

This section introduces the preliminary knowledge for our work. We first present the higher-order smoothness in Appendix C.1, then introduce the wavelets in Appendix C.2, and finally online distribution shift adaptation in Appendix C.3, including two ways to adapt to the online label shift problem.

C.1. Higher-order Smoothness

In this work, we adopt the higher-order path length (Tibshirani, 2014) as the non-stationarity measure, which is defined in (5) and we restate it below:

$$P_T^k \triangleq T^k \|\boldsymbol{D}^{k+1}\mathring{\mathbf{u}}_{[1,T]}\|_1, \text{ for } k \geq 0,$$

where $\mathring{\mathbf{u}}_{[1,T]} = [\mathring{\mathbf{u}}_1, \ldots, \mathring{\mathbf{u}}_T]^\top \in \mathbb{R}^{T \times d}$ is the matrix consisting of underlying comparators, and $\boldsymbol{D}^k \in \mathbb{R}^{(T-k)\times T}$ is the k-th order discrete difference matrix (Tibshirani, 2014). For clarity, the first order matrix $\boldsymbol{D}^1 \in \mathbb{R}^{(T-1)\times T}$ is

$$\boldsymbol{D}^1 = \begin{bmatrix} -1 & 1 & 0 & \ldots & 0 & 0 \\ 0 & -1 & 1 & \ldots & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \ldots & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(T-1)\times T}. \tag{12}$$

By recursively applying $\boldsymbol{D}^i = \widetilde{\boldsymbol{D}}^1 \boldsymbol{D}^{i-1} \; \forall i \geq 2$ with $\widetilde{\boldsymbol{D}}^1$ being the $(T-i) \times (T-i+1)$ truncation of $\boldsymbol{D}^1$, we can obtain k-th order difference matrix $\boldsymbol{D}^k$. When $k = 0$, we get $P_T^0 = \|\boldsymbol{D}^1 \mathring{\mathbf{u}}_{[1,T]}\|_1 = \sum_{t=2}^T \|\mathring{\mathbf{u}}_t - \mathring{\mathbf{u}}_{t-1}\|_1$, which recovers commonly used path length (Zinkevich, 2003).

The higher-order path length (5) has drawn increased interest recently (Tibshirani, 2014; Baby & Wang, 2023). The advantage of higher-order path length over first-order path length stems from the enhanced flexibility of regularizing the comparator sequence, thereby accommodating a broader range of scenarios. The higher the order of the definition, the smoother the comparators will be. For instance, if the comparator sequence varies linearly, then its P_T^1 will be zero, while its $P_T^0 = \mathcal{O}(T)$ is much larger. Therefore, higher-order path length can provide a more accurate measure of non-stationarity in scenarios where the underlying distributions of environments exhibit higher-order smoothness.

C.2. Wavelet Analysis

Wavelet is a powerful mathematical tool for signal processing (Daubechies, 1992), and it works by decomposing a signal using a series of orthogonal basis functions, representing a signal in terms of its time and frequency components simultaneously. In this section, we first give an introduction to wavelets, especially the construction of *CDJV wavelets*. Then, we demonstrate the superiority of wavelets.

C.2.1. CDJV Wavelets

Introduction of Wavelets. A wavelet function, or simply a "wavelet", exhibits two essential properties: oscillation and short duration. These attributes highlight the wavelet's localization, allowing for the local representation of a signal in terms of its time and frequency components simultaneously. The concept of wavelets originated in 1910, when Haar (Haar, 1909) first introduced a piecewise constant wavelet function:

$$\psi(t) = \begin{cases} 1, & \text{if } 0 \leqslant t < 1/2; \\ -1, & \text{if } 1/2 \leqslant t < 1; \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Through dilations and translations of this wavelet function ψ, a set of orthonormal bases are generated, presented as

$$\left\{ \psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t - 2^j n}{2^j}\right) \right\}_{(j,n)\in\mathbb{Z}^2},$$

where j is a positive number that defines the scale (level) of the wavelet basis, and n is a positive number that represents the shift of the wavelet basis. Given a set of wavelet bases $\psi_{j,n}$, the wavelet transformation means projecting the signal f onto the wavelet bases. In other words, the wavelet transformation of any signal f can be obtained by its wavelet inner-product coefficients as follows:

$$\boldsymbol{\alpha} = \langle f, \psi_{j,n} \rangle = \int_{-\infty}^{+\infty} f(t) \cdot \psi_{j,n}(t)\, \mathrm{d}t, \tag{14}$$

which means that wavelet transformation can be seen as a *convolution* of the signal and wavelet bases.

For the discrete cases, the situation is similar to the continuous case as presented in (14). By discretely sampling the wavelet bases, one can use a wavelet transformation matrix to obtain wavelet coefficients. We offer an illustrative example to demonstrate the construction of the Haar wavelet transformation matrix $W \in \mathbb{R}^{T\times T}$. This requires sampling basis functions $\psi_{j,n}$ at points $\{i/T\}$, for $i \in [T]$, and scaling them by $T^{-1/2}$. For simplicity, we illustrate this process using $T = 4$. The discrete Haar transformation matrix is:

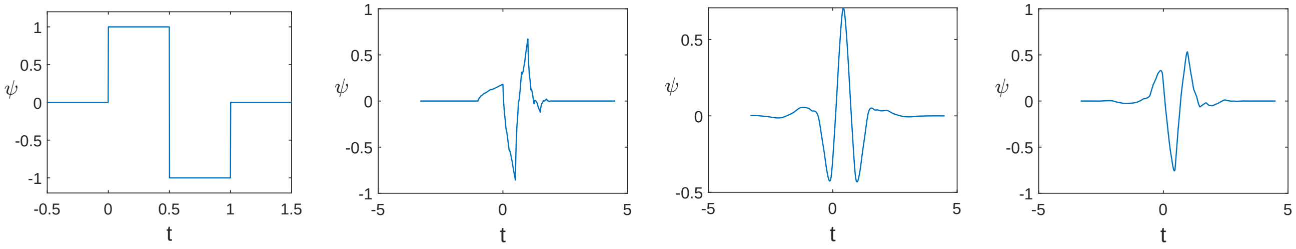
$$W = \begin{bmatrix} 1/2 & 1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & -1/2 & -1/2 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}, \tag{15}$$

and one can obtain the Haar wavelet coefficients by matrix multiplication $\boldsymbol{\alpha} = W^\top \cdot f$ (Daubechies, 1992; Mallat, 1999), where $f \in \mathbb{R}^T$ is the signal, $\boldsymbol{\alpha} \in \mathbb{R}^T$ is the Haar wavelet coefficients that we obtain.

We now provide the formal definition of wavelets (Daubechies, 1992). A wavelet is a function ψ in $L^2(\mathbb{R})$ such that the functions $\{\psi_{j,n}(t) = 2^{j/2}\psi(2^j t - n)\}$ form a set of orthonormal bases for $(j,n) \in \mathbb{Z}^2$. The wavelet transformation of a signal f derived by this wavelet function ψ corresponds to the wavelet inner-product coefficients, as illustrated in (14). Therefore, the wavelet transformation represents a convolution of the signal and wavelet bases. Derived by different constructions of wavelets, we can obtain various kinds of wavelet functions that can be employed in numerous scenarios. Commonly used wavelet functions include the Haar wavelet (Haar, 1909), Daubechies wavelet (Daubechies, 1992), Coiflet wavelet (Beylkin et al., 1991), and Symmlet wavelet (Daubechies, 1988) are illustrated in Figure 4.

Then, we introduce the 'order' of wavelets. We call that a wavelet function $\psi(\cdot)$ has order r if it has r vanishing moments. We give the definition of the vanishing moment of wavelets following Donoho & Johnstone (1998) and Johnstone (2017).

Definition 1 (vanishing moment)**.** *A wavelet function $\psi(x)$ is said to have r vanishing moments if $\int_0^1 x^i \psi(x) dx = 0$, for any $i = 0, \ldots, r-1$, which means the wavelet function ψ is orthogonal to all polynomials of degree at most $r-1$.*



(a) Haar Wavelet (b) Daubechies Wavelet (c) Coiflet Wavelet (d) Symmlet Wavelet

Figure 4: A graphical representation of different types of wavelets. (a) Haar wavelet (Haar, 1909), (b) Daubechies wavelet (Daubechies, 1992), (c) Coiflet wavelet (Beylkin et al., 1991), and (d) Symmlet wavelet (Daubechies, 1988).

Construction of CDJV Wavelets. In this study, we adopt a classic wavelet construction, Cohen-Daubechies-Jawerth-Vial (CDJV) wavelets (Cohen et al., 1993a;b), to construct wavelet functions, which enjoys desirable properties such as orthogonality, vanishing moments, and compact support. We summarize the *CDJV construction* (Johnstone, 2017; Donoho & Johnstone, 1998) as below, and more detailed descriptions can be found in Cohen et al. (1993a;b) and Johnstone (2017).

The CDJV wavelet functions contain three components: the interior part ψ^{int}, the left boundary part ψ^L, and the right boundary part ψ^R. One can obtain the CDJV wavelet coefficients using these three components of wavelet functions. Specifically, the signal is divided into three segments: the left edge, the right edge, and the interior. The left edge is used in conjunction with ψ^L, the right edge with ψ^R, and the interior part with ψ^{int}, respectively. The overall wavelet coefficients result from the concatenation of the coefficients derived from these three parts.

The construction of CDJV wavelets begins with the *Daubechies wavelet function* (Daubechies, 1992) ψ with r vanishing moments and minimal support $[-r+1, r]$ (the support of a wavelet function is a subset of the domain containing all the points where the function value is non-zero). For j such that $2^j \geq 2r$ and for $n = r, \ldots, 2^j - r - 1$, the interior wavelet functions $\psi^{\text{int}}_{j,n} = \psi_{j,n}$ have support entirely contained in $[0,1]$ and so are left unchanged. At the boundaries, for $n = 0, \ldots, r-1$, construct orthonormal functions ψ^L_n with support $[0, r+n]$ and ψ^R_n with support $[-r-n, 0]$, and set

$$\psi^{\text{int}}_{j,n}(t) = 2^{j/2}\psi^L_n\left(2^j t\right), \quad \psi^{\text{int}}_{j,2^j-n-1}(t) = 2^{j/2}\psi^R_n\left(2^j(t-1)\right), \tag{16}$$

where the functions ψ^L_n, ψ^R_n and ψ^{int}_n are finite linear combinations of scaled and translated versions of the original Daubechies wavelet function ψ, thereby retaining the same smoothness as ψ. As a result, the CDJV construction can provide a wavelet function of a given vanishing moment of r. We summarize the advantages of the CDJV construction as follows:

Proposition 2. *The CDJV construction with r vanishing moments satisfies*

1. *Let $l = \lceil \log 2r \rceil$. Then V_l contains polynomials of degree $\leq r-1$, where V_l is the space spanned by the wavelet functions $\{\psi_{l,n}, n = 1, \ldots, r-1\}$.*
2. *All wavelet bases $\psi^{\text{int}}_{j,n}$, $\psi^{\text{L}}_{j,n}$, and $\psi^{\text{R}}_{j,n}$ have r vanishing moments.*

To further clarify the CDJV wavelets, we present a special case of CDJV construction with 1 vanishing moment, which is the *Haar wavelets*, formally defined as (13). The transformation matrix for the first-order CDJV wavelet aligns with the Haar wavelet matrix, as depicted in (15), and we can get the corresponding CDJV wavelet coefficients using the matrix multiplication $\boldsymbol{\alpha} = W^\top \cdot f$, where $f \in \mathbb{R}^T$ is the signal, $\boldsymbol{\alpha} \in \mathbb{R}^T$ is the Haar wavelet coefficients that we obtain. For the more complicated k-th order CDJV wavelet, we can get the set of wavelet bases recursively by applying (16) based on a k-th order Daubechies wavelet function, and the corresponding wavelet transformation matrix can be obtained by sampling the basis functions $\psi_{j,n}$ at i/T, for $i \in [T]$.

Difference between Haar and CDJV wavelets. It is important to recognize that the Haar wavelet, known for its simplicity, represents a special case within the broader category of CDJV wavelets. The primary difference between Haar and CDJV wavelets lies in their basis functions: Haar wavelet bases are piece-wise constant functions with a maximum length of 2, whereas CDJV wavelets feature higher-order bases extending to a length of k. Consequently, the simplicity of Haar wavelets has facilitated the development of efficient computation and online updating techniques, as documented in (Baby & Wang, 2019) and (Zhang et al., 2023b). In contrast, the more intricate structure of CDJV wavelets presents significant challenges in achieving efficient computation. The efficient Haar wavelets update mechanism can not be directly extended for the CDJV wavelets. To this end, a binary tree is employed in our work to address this issue.

C.2.2. Superiority of Wavelets

In this section, we illustrate the superiority of the wavelets.

Multi-resolution Ability. The effectiveness of the wavelet detection in the non-stationary online learning scenario lies in its multi-resolution ability (Cohen et al., 1993b; Donoho & Johnstone, 1998; Johnstone, 2017), which allows for capturing both high-frequency, short-duration noises and low-frequency, long-duration trends in a signal. This allows for a more accurate and detailed representation of complex, non-stationary signals compared to traditional Fourier analysis. Besides, while previous STFT or FFT might be a simpler and faster method, its detection power and statistical properties are not yet clear, making it less suitable for deriving the dynamic regret problem. Therefore, the wavelets can capture both high-frequency, short-duration noises and low-frequency, long-duration trends in a signal. By decomposing the observed empirical comparator sequence $\{\widetilde{\mathbf{u}}_t\}$ using wavelets, we can denoise the sequence by filtering out the noisy components and thereby monitor the intensity of the ground truth environmental changes of $\{\mathring{\mathbf{u}}_t\}_{t=1}^{T}$.

Parallelism. The proposed streaming wavelet operator, as described in Section 3.2, exhibits exceptional parallelism, enabling concurrent updates at each round. Specifically, for a signal of length T, parallel updates can be performed across all $d\log(T)$ layers organized by the binary indexed tree and each dimension within the d-dimensional space. As a result, this offers advantageous properties for implementing streaming wavelet operators, especially in long time-duration and high-dimensional signals, making it suitable for practical online learning applications deployed on GPU facilities. Empirical evidence can be found in Appendix A.4.

C.3. Online Label Shift

Online label shift is a new problem setup drawing much attention in recent years (Wu et al., 2021; Bai et al., 2022; Baby et al., 2023). We first give a motivation example from Wu et al. (2021): consider a medical diagnosis model classifying whether a patient suffers from flu or hay fever (unlabeled data). Although the two diseases share similar symptoms throughout the year (same class-conditional distribution), one is far more prevalent than the other (different label distributions), depending on the season and whether an outbreak occurs. In this section, we describe the omitted details of applying our wavelet-based detection-restart framework to handle the online label shift problem. We first introduce the following lemma to show that the estimated label distribution by BBSE is unbiased towards the ground-truth label distribution.

Lemma 1. *The BBSE's estimation $\widetilde{\boldsymbol{\mu}}_t = C_0^{-1}\widetilde{\boldsymbol{\mu}}_{\widehat{y}_t}$ is unbiased towards the ground truth label prior $\widetilde{\boldsymbol{\mu}}_t$ if the initial data is sufficient such that we can obtain C_0.*

Proof. We rewrite the BBSE's estimation as $\widetilde{\boldsymbol{\mu}}_t = C_0^{-1}\widetilde{\boldsymbol{\mu}}_{\widehat{y}_t} = C_0^{-1}\frac{1}{|S_t|}\sum_{\mathbf{x}\in S_t} h_0(\mathbf{x})$. Taking expectations of both sides,

$$\mathbb{E}_{S_t\sim\mathcal{D}_t}\left[\widetilde{\boldsymbol{\mu}}_t\right] = \mathbb{E}_{S_t\sim\mathcal{D}_t}\left[C_0^{-1}\left(\frac{1}{|S_t|}\sum_{\mathbf{x}\in S_t} h_0(\mathbf{x})\right)\right] = \mathbb{E}_{S_t\sim\mathcal{D}_t}\left[C_0^{-1}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t}\left[h_0(\mathbf{x})\right]\right] = C_0^{-1}\boldsymbol{\mu}_{\widehat{y}_t} = \boldsymbol{\mu}_t,$$

which finishes the proof. □

C.3.1. Reweighting-based Update

The OGD update procedure described in (11) needs to store all the initial data to obtain $R_0^j(\cdot), \forall j \in \{1,\dots,K\}$, which may cause computational and storage burdens. To this end, another classifier update approach involves updating the classification model by reweighting the training data using the predicted class prior distribution $\widehat{\boldsymbol{\mu}}_t$ (Wu et al., 2021), which only needs to store the initial label prior $\mathcal{D}_0$ and the initial predictor h_0 to get the classifier $h_t : \mathcal{X} \to \mathcal{Y}$. As stated in (10), we can reweight the initial classifier h_0 to get the current classifier h_t, which is restated as below:

$$[h_t(\mathbf{x})]_j = \frac{1}{Z(\mathbf{x})}\frac{[\widehat{\boldsymbol{\mu}}_t]_j}{\mathcal{D}_0(y=j)}[h_0(\mathbf{x})]_j,\ \forall j \in [K],$$

with $Z(\mathbf{x}) = \sum_{j=1}^{K}\frac{[\widehat{\boldsymbol{\mu}}_t]_j}{\mathcal{D}_0(j)}[h_0(\mathbf{x})]_j$ being the normalization factor. Similarly, the comparator (Bayesian optimal classifier) $h_t^\star$ can be constructed by reweighting initial classifier h_0:

$$[h_t^\star(\mathbf{x})]_j = \frac{1}{Z(\mathbf{x})}\frac{[\boldsymbol{\mu}_t]_j}{\mathcal{D}_0(y=j)}[h_0(\mathbf{x})]_j,\ \forall j \in [K]. \tag{17}$$

We make the following assumption: for the class prior $\boldsymbol{\mu}_t$ and the initial classifier h_0, we have that $[\boldsymbol{\mu}_t]_j \geq \beta$ for all $j \in \{1, \dots, K\}$, and $[h_0(\mathbf{x})]_j \geq \alpha$ for all $\mathbf{x} \in \mathcal{X}$ and $j \in \{1, \dots, K\}$. In other words, the initial dataset should have a certain number of data for each class to avoid excessive class imbalance and unseen new class, which is a commonly used assumption in previous works (Wu et al., 2021; Bai et al., 2022). Given such an assumption, the loss function can have a bounded gradient, which is stated as follows.

Lemma 2 (Bounded Gradient, Lemma 11 & Lemma 12 of Baby et al. (2023))**.** *For commonly used loss functions such as Logistic loss and 0-1 loss, the gradient norm of the loss function is bounded by L, where L is a constant related to α and β, where $[h_0(\mathbf{x})]_j \geq \alpha$, and $[\boldsymbol{\mu}_t]_j \geq \beta$ for all $\mathbf{x} \in \mathcal{X}$, $t \in [T]$, and $j \in \{1, \dots, K\}$.*

In light of Lemma 2, the objective can be reformulated as selecting the reweighting vector $\widehat{\boldsymbol{\mu}}_t \in \Delta_K$ that minimizes:

$$\mathbf{Reg}_T^{\mathbf{d}}(\{R_t, h_t^\star\}_{t=s}^e) = \sum_{t=1}^T R_t(h_t) - \sum_{t=1}^T R_t(h_t^\star) \leq L \sum_{t=1}^T \|\widehat{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_t\|_2. \tag{18}$$

Therefore, for the reweighting updates, our algorithm focuses on estimating the class prior $\widehat{\boldsymbol{\mu}}_t \in \Delta_K$ at each time step to construct the classifier h_t.

However, optimizing the L_2-norm of the difference between the estimated and actual label prior directly is not optimal, given that the label prior is a probability distribution, and the L_2-norm does not effectively measure probability distributions. Additionally, the presence of out-of-distribution data can significantly distort the estimated label prior from the actual one, causing a large L_2-norm difference, which is undesirable. Besides, employing the previously discussed BBSE method, we can obtain an estimated class prior $\widetilde{\boldsymbol{\mu}}_t$ at each time step. However, the BBSE estimator $\widetilde{\boldsymbol{\mu}}_t$ may exhibit noise and high variance due to label shifts and the limited number of samples in S_t for $\mathcal{D}_t$, resulting in a lack of dynamic regret guarantees. As a result, inspired by the method proposed by Garg et al. (2020), we adopt Kullback-Leibler (KL) divergence to align the predicted label prior $\widehat{\boldsymbol{\mu}}_t$ and the ground truth label prior $\boldsymbol{\mu}_t$, defined as $\mathrm{KL}(\widehat{\boldsymbol{\mu}}_t \| \boldsymbol{\mu}_t) = \sum_{j=1}^K [\widehat{\boldsymbol{\mu}}_t]_j \log \frac{[\widehat{\boldsymbol{\mu}}_t]_j}{[\boldsymbol{\mu}_t]_j} + [\boldsymbol{\mu}_t]_j - [\widehat{\boldsymbol{\mu}}_t]_j$.

Lemma 3. *Let $\widehat{\boldsymbol{\mu}}_t \in \Delta_K$ be the predicted label prior, and the classifier h_t is updated by reweighting* (10)*, then for any interval $\mathcal{I} = [s, e] \subseteq [T]$ we have $\mathbf{Reg}_{\mathcal{I}}^{\mathbf{d}}(\{R_t, h_t^\star\}_{t=s}^e) \leq \mathcal{O}\left(\sqrt{|\mathcal{I}| \cdot \sum_{t=s}^e \mathrm{KL}(\widehat{\boldsymbol{\mu}}_t \| \boldsymbol{\mu}_t)}\right)$.*

Lemma 3 illustrates that minimizing the KL-Divergence between the predicted and ground-truth label prior serves as an upper bound for the expected risk. In the following, we consider how to minimize the KL-Divergence between the predicted $\widehat{\boldsymbol{\mu}}_t$ and ground-truth label prior $\boldsymbol{\mu}_t$. The proof of Lemma 3 can be found in Appendix D.7.

Online Newton Step with Dummy Feature. Through Lemma 3, we have correlated the expected risk minimization problem to the KL-Divergence matching problem between the predicted label prior $\widehat{\boldsymbol{\mu}}_t$ and the ground truth label prior $\boldsymbol{\mu}_t$. Given that the KL-divergence is an exp-concave function, we use Online Newton Step (ONS) algorithm (Hazan et al., 2007) to estimate the $\widehat{\boldsymbol{\mu}}_t$. This algorithm attains a logarithmic static regret when applied to exp-concave loss functions. Specifically, given an interval $\mathcal{I} = [s, e] \subseteq [T]$ starting at time s, for each $t \in \mathcal{I}$, ONS updates by

$$\widehat{\boldsymbol{\mu}}_{t+1} = \Pi_{\Delta_K}^{A_t}\left[\widehat{\boldsymbol{\mu}}_t - \frac{1}{\varepsilon} A_t^{-1} \nabla \widetilde{L}_t(\widehat{\boldsymbol{\mu}}_t)\right], \tag{19}$$

where the matrix $A_t = \lambda \mathbf{I} + \sum_{\tau=s}^t \nabla \widetilde{L}_s(\widehat{\boldsymbol{\mu}}_s) \nabla \widetilde{L}_s(\widehat{\boldsymbol{\mu}}_s)^\top$. Same as the standard ONS algorithm, we set the hyperparameters in ONS as $\varepsilon = \frac{1}{2\alpha}$ and $\lambda = \frac{1}{\varepsilon^2}$. In above, the projection function is defined as $\Pi_{\Delta_K}^{A_t}[\boldsymbol{\mu}_1] = \arg\min_{\boldsymbol{\mu} \in \Delta_K} \|\boldsymbol{\mu} - \boldsymbol{\mu}_1\|_{A_t}$ and Δ_K is the parameter space (simplex) of the class prior. Note that the learner cannot receive the ground-truth loss function L_t, but only observes an empirical estimation $\widetilde{L}_t$, which is formally defined as

$$\widetilde{L}_t(\boldsymbol{\mu}) = \sum_{j=1}^K (\partial(\psi_{\mathsf{KL}}[\boldsymbol{\mu}]_j)[\boldsymbol{\mu}]_j - \psi_{\mathsf{KL}}([\widetilde{\boldsymbol{\mu}}]_j)) - \sum_{j=1}^K \partial\psi_{\mathsf{KL}}([\boldsymbol{\mu}]_j)[\boldsymbol{\mu}_t]_j = \sum_{j=1}^K [\boldsymbol{\mu}]_j - \sum_{j=1}^K [\widetilde{\boldsymbol{\mu}}]_j \log([\boldsymbol{\mu}]_j),$$

where $\widetilde{\boldsymbol{\mu}}_t$ is class prior estimated by BBSE. It is easy to verify that the empirical loss $\widetilde{L}_t$ is unbiased to the ground-truth L_t.

To handle the higher-order path length, we construct a k-th order dummy feature $\boldsymbol{\phi}_t$ at each round, which is defined in (21). We combine this dummy feature with a linear predictor $\mathbf{w}_t$, and let the input of ONS algorithm be $\widehat{\boldsymbol{\mu}}_t = \mathbf{w}_t^\top \boldsymbol{\phi}_t$.

Table 5: Summary of applying the wavelet-based detection-restart framework to handle online label shift. Combining our framework with previous existing online updates, i.e., OGD and Reweighting update, yields two new algorithms and immediately achieve the optimal dynamic regret guarantee for the OLS problem.

	Risk Function	**Wavelet Detection**	**Online Algorithm $\mathcal{A}$**	**Dynamic Regret**
Wav-R (10)	Lipschitz	$(k+1)$-th order wavelets	Reweighting (Baby et al., 2023)	$\widetilde{\mathcal{O}}\left(\max\{T^{\frac{k+2}{2k+3}}(P_T^k)^{\frac{1}{2k+3}}, \sqrt{T}\}\right)$
Wav-O (11)	Convex	1-st order wavelets	OGD Update (Bai et al., 2022)	$\widetilde{\mathcal{O}}\left(\max\{T^{\frac{2}{3}}(P_T^0)^{\frac{1}{3}}, \sqrt{T}\}\right)$

We further assume the $[\widehat{\boldsymbol{\mu}}_t]_j = [\mathbf{w}_t^\top \boldsymbol{\phi}_t]_j > \alpha, \forall j \in [K]$. Combining our detection framework with the reweighting-based update, we can obtain the following dynamic regret bound as shown in Theorem 4. When $k = 0$, Theorem 4 implies an $\mathcal{O}(T^{\frac{2}{3}}(P_T^0)^{\frac{1}{3}})$ rate, which is optimal for online label shift (Bai et al., 2022; Baby et al., 2023). Crucially, the new algorithm necessitates the maintenance of only a single classifier, leading to a significant improvement in both the computational and storage complexities compared to ensemble-based methods. Specifically, the required number of projections onto the feasible domain decreases from $\mathcal{O}(\log T)$ times to just 1 time at each round, due to online ensemble methods typically require projecting a group of $\mathcal{O}(\log T)$ models at each round, while we only maintain one model. Moreover, storage complexity is reduced from $\mathcal{O}(d^2k^2 \log T)$ to $\mathcal{O}(d^2k^2 + dk \log T)$, as we maintain a set of wavelet coefficients instead of a group of base learners and therefore become much lightweight.

C.3.2. OGD-based Update

In this part, we establish the risk estimator as $\widehat{R}_t(\mathbf{w}) = \sum_{j=1}^K [\widetilde{\boldsymbol{\mu}}_t]_j \cdot R_0^j(\mathbf{w})$, where $\widetilde{\boldsymbol{\mu}}_t$ is the class prior estimated by BBSE. We introduce the following lemma to show that the predicted risk $\widehat{R}_t$ is unbiased with respect to the expected risk R_t.

Lemma 4 (Lemma 1 of Bai et al. (2022))**.** *The estimator $\widehat{R}_t(\mathbf{w})$ is unbiased to $R_t(\mathbf{w}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}[\ell(h(\mathbf{w},\mathbf{x}), y)]$, i.e., $\mathbb{E}_{S_t\sim\mathcal{D}_t}[\widehat{R}_t(\mathbf{w})] = R_t(\mathbf{w})$, for any $\mathbf{w} \in \mathcal{W}$ independent of S_t.*

We summarize the results of combining our wavelet-based detection-restart framework with previous existing online algorithms to handle OLS, as shown in Table 5.

We further note that our wavelet-based detection-restart framework holds potential for addressing the online generalized label shift (Wu et al., 2024), where the conditional distribution remains invariant given a feature extractor $\phi : \mathbb{R}^d \to \mathbb{R}^{d'}$, i.e., $\mathcal{D}_t(\phi(\mathbf{x}) \mid y) = \mathcal{D}_{t-1}(\phi(\mathbf{x}) \mid y)$, but $\mathcal{D}_t(y)$ varies over time. In this case, we can first extract the features and then apply our wavelet detection framework to detect the change of the class prior, which restarts the classifier when the environmental changes are detected. We leave the detailed analysis of this problem to future work.

C.4. Summary of the Improvement on Efficiency

In this part, we highlight and summarize our efficiency improvement, supported by both empirical and theoretical evidence.

Empirical evidence. Our experiments show our method's substantial efficiency improvement in various scenarios. Specifically, in our primary application (online label shift), Figures 2(b) & 2(c) illustrate that our detection-based approaches (*Wav-O* & *Wav-R*) achieve comparable or even slightly better performance to traditional ensemble-based methods (*ATLAS* & *FLH-FTL*) with *nearly 300% running time speedup* and *50% reduction in memory usage*.

Theoretical analysis. Let us define several key terms: computational complexity of updating a single model (C_{model}), obtaining an unbiased estimator (C_{esti}), and wavelet detecting (C_{detect}). We list a running complexity comparison in Table 6. While exhibiting a comparable computational complexity for the simple case (linear model with convex losses), our detection-based framework demonstrates remarkable improvements in more complicated and realistic scenarios. These include (i) exp-concave case, and (ii) general convex case used in online label shift.

As illustrated in Table 6, our wavelet detection algorithm speeds up significantly in many cases. This is achieved by maintaining a set of wavelet coefficients instead of a group of base models: we only need to update the model once, thus becoming more computationally efficient. Typically, those ensemble-based methods incur a computational complexity of $C_{\text{model}} \times \log T$ due to the requirement of maintaining $\mathcal{O}(\log T)$ base learners. In contrast, our wavelet-based detection method maintains only one model, along with a multi-resolution detection/exploration using $\mathcal{O}(\log T)$ wavelet coefficients. Consequently, our wavelet-based detection method is much more efficient, particularly when using complicated base models, such as overparametrized models in practice, where C_{model} can be very large.

Table 6: Computational complexities of our wavelet detection framework with model ensemble methods under different scenarios.

	Wav. Detect Framework	**Ensemble**	Empirical Speedup	Remark
General Case	$C_{\text{model}} \times 1$ $+ C_{\text{esti}} + C_{\text{detect}}$	$C_{\text{model}} \times \mathcal{O}(\log T)$	/	/
Linear model, convex loss	$d + d + d \log T$ $= \mathcal{O}(d \log T)$	$\mathcal{O}(d \log T)$	/	ensemble can only handle first-order path length
(i) Exp-concave	$d^2 + d + d \log T$ $= \mathcal{O}(d^2 + d \log T)$	$\mathcal{O}(d^2 \log T)$	≈200%, see Figure 2(b) *Wav-R* vs. *FLH-FTL*	/
(ii) Online Label Shift	$d + d + K \log T$ $= \mathcal{O}(d + K \log T)$	$\mathcal{O}(d \log T)$	≈300%, see Figure 2(b) *Wav-O* vs. *ATLAS*	K is # classes in OLS, $K \ll d$

We note that both our method and previous ensemble-based methods contain a computational complexity having $\mathcal{O}(\log T)$ dependency. But our method is more efficient in many cases, as it only maintains multiple wavelet coefficients instead of multiple model parameters, which can be more lightweight in many scenarios. Actually, this raises an interesting question about the necessity of an additional computational overhead of $\mathcal{O}(\log T)$ compared to the stationary algorithms, when handling non-stationary online environments with inherent uncertainty.

D. Proofs

This section provides the proofs of Section 3 and Section 4.

D.1. General Regret Analysis Recipe

In this part, we present a general analysis framework for our detection-restart framework. Suppose that there are $M-1$ change points detected by Algorithm 1. The entire time horizon can be thus decomposed into M intervals denoted by $\{\mathcal{I}_1, \ldots, \mathcal{I}_M\}$ with $\mathcal{I}_i = [s_i, e_i]$ for $i \in [M]$, and then we have $\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) = \sum_{i=1}^M \mathbf{Reg}_{\mathcal{I}_i}^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=s_i}^{e_i})$. Therefore, it suffices to control the regret within each interval $\mathbf{Reg}_{\mathcal{I}_i}^{\mathbf{d}}$ and the total number of intervals M.

Here we present a general recipe to bound $\mathbf{Reg}_{\mathcal{I}_i}^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=s_i}^{e_i})$ with respect to the k-th order path length. Previous work (Baby & Wang, 2023) constructed a dummy feature to address the special case of the second-order path length. In the general k-th order case, we need more effort. We decompose

$$\mathbf{Reg}_{\mathcal{I}_i}^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=s_i}^{e_i}) = \underbrace{\sum_{t=s_i}^{e_i} f_t(\boldsymbol{\theta}_t) - \sum_{t=s_i}^{e_i} f_t(\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t)}_{\texttt{static regret w.r.t. linear predictor}} + \underbrace{\sum_{t=s_i}^{e_i} f_t(\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t) - \sum_{t=s_i}^{e_i} f_t(\mathring{\mathbf{u}}_t)}_{\texttt{variation of comparator}}, \tag{20}$$

where the first term characterizes the regret of $\boldsymbol{\theta}_t$ with respect to a linearized comparator $\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t$, and the second term characterizes variations of the comparator sequence. In (20), $\boldsymbol{\beta}_{\mathcal{I}_i} \in \mathbb{R}^{(k+1)\times d}$ is the best *static linear predictor* within the interval $\mathcal{I}_i$, and $\boldsymbol{\phi}_t \in \mathbb{R}^{k+1}$ is the k-th order dummy feature within the interval $\mathcal{I}_i = [s_i, e_i]$ defined as follows:

$$\boldsymbol{\phi}_t = \Big[1, (t - s_i + k + 1), (t - s_i + k + 1)^2, \ldots, (t - s_i + k + 1)^k\Big]^\top, \tag{21}$$

where s_i is the starting point of the interval $\mathcal{I}_i$. Let $\boldsymbol{\Phi} \triangleq [\boldsymbol{\phi}_{s_i}, \ldots, \boldsymbol{\phi}_{e_i}]^\top \in \mathbb{R}^{|\mathcal{I}_i|\times(k+1)}$ be the matrix of dummy features, and $\mathring{\mathbf{u}}_{[s_i,e_i]} = [\mathring{\mathbf{u}}_{s_i}, \ldots, \mathring{\mathbf{u}}_{e_i}]^\top \in \mathbb{R}^{|\mathcal{I}_i|\times d}$, the best linear predictor $\boldsymbol{\beta}_{\mathcal{I}_i}$ is obtained by least-square regression: $\boldsymbol{\beta}_{\mathcal{I}_i} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{(k+1)\times d}} \sum_{t\in\mathcal{I}_i} \|\boldsymbol{\beta}^\top \boldsymbol{\phi}_t - \mathring{\mathbf{u}}_t\|_2^2 = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top \mathring{\mathbf{u}}_{[s_i,e_i]}$.

The intuition of constructing dummy features (21) is that the j-th element of $\boldsymbol{\phi}_t$ captures the j-th order difference between the optimal static comparator $\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t$ and the ground truth comparator $\mathring{\mathbf{u}}_t$. Therefore, by using a linear predictor $\boldsymbol{\beta}_{\mathcal{I}_i}$ combined with $\boldsymbol{\phi}_t$, the second term of (20) characterizes higher-order changes of $\mathring{\mathbf{u}}_t$. Consider a simple case when $k = 0$, then $\boldsymbol{\phi}_t = 1$ and $\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t = 1/|\mathcal{I}_i| \cdot \sum_{t=s_i}^{e_i} \mathring{\mathbf{u}}_t$ is the average of the comparators within interval $\mathcal{I}_i$ and remains unchanged within $\mathcal{I}_i$. We then formally define the *comparator gap* $C_{\mathcal{I}_i}^k$ as (22) and restated as below.

$$C_{\mathcal{I}_i}^k \triangleq \sum_{t\in\mathcal{I}_i} \|\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t - \mathring{\mathbf{u}}_t\|_1, \tag{22}$$

which measures the *higher-order smoothness* of the comparator sequence. The smaller the comparator gap $C_{\mathcal{I}_i}^k$ is, the smoother the comparator sequence will be. Consequently, Requirement 1 suggests that the online algorithm should perform well under stationary environments. Importantly, $C_{\mathcal{I}_i}^k$ establishes a connection between the wavelet coefficients and the path length of the comparator sequence, as further illustrated in Theorem 1.

D.2. Proof of Theorem 1

Proof. The proof of Theorem 1 contains two parts: controlling the number of intervals M, and controlling the comparator gap $C_{\mathcal{I}_i}^k$ within each interval $\mathcal{I}_i$.

Step 1. Bound the total number of Intervals M. We first prove that the total number of the intervals is bounded by $M \leq \widetilde{\mathcal{O}}(T^{\frac{1}{2k+3}}(P_T^k)^{\frac{2}{2k+3}})$. We first present the following lemma corresponding to the wavelet coefficients:

Lemma 5. *Let $\bar{\mathbf{u}}_t$ denote the centralized comparator sequence, i.e., $\bar{\mathbf{u}}_t = \mathring{\mathbf{u}}_t - \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t$ and $\bar{\mathbf{u}}_{[s,e]} = [\bar{\mathbf{u}}_s, \ldots, \bar{\mathbf{u}}_e]^\top$ within interval $\mathcal{I} = [s,e] \subseteq [T]$, then for the CDJV wavelet transformation matrix $W_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{I}|}$, the F-norm of wavelet coefficients $\bar{\boldsymbol{\alpha}}_{[s,e]}$ satisfies*

$$\|\bar{\boldsymbol{\alpha}}_{[s,e]}\|_{\mathrm{F}} \leq \mathcal{O}(\sqrt{|\mathcal{I}|} \cdot P_{\mathcal{I}}^k),$$

where $\bar{\boldsymbol{\alpha}}_{[s,e]}$ is the wavelet coefficients of the sequence $\bar{\mathbf{u}}_{[s,e]}$ calculated by transformation matrix W, $\boldsymbol{\beta}_{\mathcal{I}}$ and $\boldsymbol{\phi}_t$ are defined in (21), and $P_{\mathcal{I}}^k = |\mathcal{I}|^k \|\boldsymbol{D}^{k+1}\mathring{\mathbf{u}}_{[s,e]}\|_1$ is the k-th order path length in $\mathcal{I}$.

Proof of Lemma 5. The intuition is that due to the orthogonality of the wavelet transformation matrix $W_{\mathcal{I}}$, the wavelets allow for a connection between the F-norm (Frobenius norm) of the wavelet coefficient matrix and the path length of the original sequence. To proof Lemma 5, we carefully analyze the F-norm of the wavelet coefficients $\bar{\boldsymbol{\alpha}}_{[s,e]}$ as follows:

$$\begin{aligned}
\left\|\boldsymbol{\alpha}_{[s,e]}\right\|_{\mathrm{F}} &= \left\|W_{\mathcal{I}}^\top \cdot \mathrm{pad}\{\mathbf{u}_{[s,e]}\}\right\|_{\mathrm{F}} = \left\|W_{\mathcal{I}}^\top \cdot \overline{\mathrm{pad}\{\mathbf{u}_{[s,e]}\}}\right\|_{\mathrm{F}} \\
&= \|\overline{\mathrm{pad}\{\mathbf{u}_{[s,e]}\}}\,\|_{\mathrm{F}} = \|\,\left[\bar{\mathbf{u}}_{[s,e]}, -\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_{e+1}, -\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_{e+2}, \ldots, -\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_L\right]\,\|_{\mathrm{F}} \\
&\leqslant \left\|\bar{\mathbf{u}}_{[s,e]}\right\|_{\mathrm{F}} + \|[\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_{e+1}, \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_{e+2}, \ldots, \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_L]\|_{\mathrm{F}} \\
&\leqslant \left\|\bar{\mathbf{u}}_{[s,e]}\right\|_{\mathrm{F}} + \sqrt{|2\mathcal{I}|} \cdot |2\mathcal{I}|^k \cdot D \cdot d \leqslant \mathcal{O}\left(\sqrt{|\mathcal{I}|} \cdot (P_{\mathcal{I}}^k + 1)\right),
\end{aligned}$$

where $L = 2^{\lceil |e-s| \rceil}$ is the length of the padded sequence, D is the diameter of $\mathbf{u}_t$, d is the dimension, $\boldsymbol{\beta}_{\mathcal{I}}$ and dummy feature $\boldsymbol{\phi}_t$ are defined in Lemma 5. $\overline{\mathrm{pad}\{\mathbf{u}_{[s,e]}\}}$ is the shifted padded sequence, i.e., minusing $\mathrm{pad}\{\mathbf{u}_{[s,e]}\}$ by the centering value of original sequence $\mathbf{u}_{[s,e]}$. The first equality is due to the orthogonality of the wavelet transformation matrix W, the second is due to the recentering operation can be removed (Lemma 6), and the last inequality is due to the definition $P_{\mathcal{I}}^k = |\mathcal{I}|^k \|\boldsymbol{D}^{k+1}\mathring{\mathbf{u}}_{[s,e]}\|_1$ and the statement in Lemma 19 of Baby & Wang (2020). The boundness of $\boldsymbol{\beta}_{\mathcal{I}}$ is shown in Lemma 12, and $\|\boldsymbol{\phi}_t\| \leq |L|^k$ due to the definition of the dummy feature (20). Thus, we finish the proof of Lemma 5. □

We remark that Lemma 5 demonstrates that the "implicit padding" mechanism in our streaming wavelet operator does not affect the detection ability of our method, and it bridges the gap between the wavelet coefficients and the path length. We now introduce the following lemma concerning our proposed streaming wavelet operator:

Lemma 6. *Let $\bar{\boldsymbol{\alpha}}_{[s,e]}$ be the wavelet coefficients of the sequence $\bar{\mathbf{u}}_{[s,e]} = [\bar{\mathbf{u}}_s, \ldots, \bar{\mathbf{u}}_e]^\top$, where $\bar{\mathbf{u}}_t = \mathring{\mathbf{u}}_t - \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t$. Also, let $\boldsymbol{\alpha}_{[s,e]}$ denote the coefficients of $\mathring{\mathbf{u}}_{[s,e]} = [\mathring{\mathbf{u}}_s, \ldots, \mathring{\mathbf{u}}_e]^\top$ obtained by streaming wavelet operator, then, $\|\boldsymbol{\alpha}_{[s,e]}\|_{\mathrm{F}} = \|\bar{\boldsymbol{\alpha}}_{[s,e]}\|_{\mathrm{F}}$.*

Proof of Lemma 6. The only difference between $\boldsymbol{\alpha}_{[s,e]}$ and $\bar{\boldsymbol{\alpha}}_{[s,e]}$ lies in: $\boldsymbol{\alpha}_{[s,e]}$ is calculated from the sequence $\mathring{\mathbf{u}}_{[s,e]}$, while $\bar{\boldsymbol{\alpha}}_{[s,e]}$ is calculated from the centralized $\bar{\mathbf{u}}_{[s,e]}$. Recall the calculation of the streaming wavelet operator: the coefficients are calculated by a *convolution* of the elements in a signal with wavelet bases. Specifically, we adopt the $(k+1)$-th order bases constructed by CDJV wavelets which have a vanishing moment of k, as defined in Definition 1. Therefore,

$$\boldsymbol{\alpha}_{[s,e]} = \boldsymbol{\psi} \circledast \mathring{\mathbf{u}}_{[s,e]}; \quad \bar{\boldsymbol{\alpha}}_{[s,e]} = \boldsymbol{\psi} \circledast \bar{\mathbf{u}}_{[s,e]},$$

where $\circledast$ represents the convolution operator, and $\boldsymbol{\psi} \in \mathbb{R}^{|\mathcal{I}| \times d}$ is a CDJV wavelet basis, in which $[\boldsymbol{\psi}]_i = \psi(i)$ (where ψ is the wavelet function defined in Appendix C.2.1). As discussed in Appendix C.2, the wavelet transformation, represented by matrix multiplication using matrix W as in Section 3.2, is equivalent to the convolution operation with a wavelet basis $\boldsymbol{\psi}$. Therefore, for the wavelet basis $\boldsymbol{\psi}$ and a single wavelet coefficient $\boldsymbol{\alpha}_t$, we have

$$\begin{aligned}
\boldsymbol{\alpha}_t &= \sum_{i\in|\boldsymbol{\psi}|} [\boldsymbol{\psi}]_i \cdot \mathring{\mathbf{u}}_{t+|\boldsymbol{\psi}|-i} = \sum_{i\in|\boldsymbol{\psi}|} [\boldsymbol{\psi}]_i \cdot \mathring{\mathbf{u}}_{t+|\boldsymbol{\psi}|-i} - \sum_{j=0}^{k} \sum_{i\in|\boldsymbol{\psi}|} [\boldsymbol{\psi}]_i \cdot [\boldsymbol{\beta}_{\mathcal{I}}]_j \cdot (t+|\boldsymbol{\psi}|-i-s+k+1)^j \\
&= \sum_{i\in|\boldsymbol{\psi}|} [\boldsymbol{\psi}]_i \cdot \Big(\mathring{\mathbf{u}}_{t+|\boldsymbol{\psi}|-i} - \sum_{j=0}^{k} [\boldsymbol{\beta}_{\mathcal{I}}]_j \cdot (t+|\boldsymbol{\psi}|-i-s+k+1)^j\Big) \\
&= \sum_{i\in|\boldsymbol{\psi}|} [\boldsymbol{\psi}]_i \cdot \big(\mathring{\mathbf{u}}_{t+|\boldsymbol{\psi}|-i} - [\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t]_{t+|\boldsymbol{\psi}|-i}\big) = \sum_{i\in|\boldsymbol{\psi}|} [\boldsymbol{\psi}]_i \cdot \bar{\mathbf{u}}_{t+|\boldsymbol{\psi}|-i} = \bar{\boldsymbol{\alpha}}_t,
\end{aligned}$$

where $|\boldsymbol{\psi}|$ is the length of the wavelet basis $\boldsymbol{\psi}$. The second equality is due to Definition 1 in the discrete case that the $(k+1)$-th order CDJV wavelet has a vanishing moment of k. Thus, the second term multiplied by $\boldsymbol{\psi}$ will equal to zero. Consequently, by summing up all the coefficients to compute the F-norm, we get $\|\boldsymbol{\alpha}_{[s,e]}\|_{\mathrm{F}} = \|\bar{\boldsymbol{\alpha}}_{[s,e]}\|_{\mathrm{F}}$. Note that this proof holds for any length of the sequence. Therefore, we finish the proof of Lemma 6. □

By the above Lemma 5 and Lemma 6, we can bridge the gap between the norm of wavelet coefficients and the k-th order path length. However, the learner cannot obtain $\boldsymbol{\alpha}_{[s,e]}$ as it cannot have access to $\mathring{\mathbf{u}}_{[s,e]}$, but only the $\widetilde{\boldsymbol{\alpha}}_{[s,e]}$ which is calculated on the empirical sequence $\widetilde{\mathbf{u}}_{[s,e]} = [\widetilde{\mathbf{u}}_s, \ldots, \widetilde{\mathbf{u}}_e]^\top$. Therefore, we need to bound the difference between $\boldsymbol{\alpha}_{[s,e]}$ and $\widetilde{\boldsymbol{\alpha}}_{[s,e]}$:

Lemma 7 (Lemma 3 of Baby & Wang (2019))**.** *For all $\mathcal{I} = [s,e] \subseteq [T]$, let $\boldsymbol{\alpha}_{[s,e]}$ denote wavelet coefficients of the sequence $\mathring{\mathbf{u}}_{[s,e]} = [\mathring{\mathbf{u}}_s, \ldots, \mathring{\mathbf{u}}_e]^\top$, and $\widetilde{\boldsymbol{\alpha}}_{[s,e]}$ be its empirical version calculated on the empirical sequence $\widetilde{\mathbf{u}}_{[s,e]} = [\widetilde{\mathbf{u}}_s, \ldots, \widetilde{\mathbf{u}}_e]^\top$, setting the threshold $\gamma = 4\sigma$ in Algorithm 1, then with probability at least $1 - 2/T^3$,*

$$\|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s,e]})\|_{\mathrm{F}} \le \|\boldsymbol{\alpha}_{[s,e]}\|_{\mathrm{F}} + \widetilde{\mathcal{O}}(1),$$

where $\delta_\gamma : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is the soft-thresholding operator (Donoho & Johnstone, 1998; Johnstone, 2017) defined as $[\delta_\gamma(A)]_{i,j} = \mathbf{sign}(A_{i,j}) \cdot \max\{|A_{i,j}| - \gamma, 0\}$.

Now we are ready to prove that $M \le \mathcal{O}(T^{\frac{1}{2k+3}} (P_T^k)^{\frac{2}{2k+3}})$. By the restart rule in Algorithm 1,

$$4\sigma \le \|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s,e]})\|_{\mathrm{F}} \lesssim \|\boldsymbol{\alpha}_{[s,e]}\|_{\mathrm{F}} \lesssim \sqrt{|\mathcal{I}|} \cdot P_{\mathcal{I}}^k = |\mathcal{I}|^{k+1/2} \cdot \|\boldsymbol{D}^{k+1} \mathring{\mathbf{u}}_{[s,e]}\|_1,$$

where the $\lesssim$ ignores the constant of order k and logarithmic factors $\log T$. The first inequality is due to the restart rule in Algorithm 1 as we set $\gamma = 4\sigma$, the second inequality is due to Lemma 7, and the third inequality is due to Lemma 5 and Lemma 6. By summing all intervals $\{\mathcal{I}_1, \ldots, \mathcal{I}_M\}$ and employing a union bound, with probability at least $1 - 2/T$,

$$\sum_{i=1}^{M} \frac{4\sigma}{|\mathcal{I}_i|^{k+1/2}} \le \sum_{i=1}^{M} \|\boldsymbol{D}^{k+1} \mathring{\mathbf{u}}_{[s_i,e_i]}\|_1 = \|\boldsymbol{D}^{k+1} \mathring{\mathbf{u}}_{[1,T]}\|_1.$$

Besides, by applying Jensen's inequality for the convex function $f(x) = 1/x^{k+1/2}$ where $x > 0$,

$$4\sigma M^{\frac{2k+3}{2}} T^{\frac{-2k-1}{2}} \le \sum_{i=1}^{M} \frac{4\sigma}{|\mathcal{I}_i|^{k+1/2}} \lesssim \|\boldsymbol{D}^{k+1} \mathring{\mathbf{u}}_{[1,T]}\|_1.$$

Rearranging the term, we can get that

$$M \lesssim 4T^{\frac{2k+1}{2k+3}} \|\boldsymbol{D}^{k+1} \mathring{\mathbf{u}}_{[1,T]}\|_1^{\frac{2}{2k+3}} = 4T^{\frac{1}{2k+3}} \|T^k \boldsymbol{D}^{k+1} \mathring{\mathbf{u}}_{[1,T]}\|_1^{\frac{2}{2k+3}} = \mathcal{O}(T^{\frac{1}{2k+3}} (P_T^k)^{\frac{2}{2k+3}}),$$

which finishes the proof of the upper bound of M.

Step 2. Bound the comparator gap $C^k_{\mathcal{I}_i}$. We first recall the definition of the comparator gap: $C^k_{\mathcal{I}_i} = \sum_{t\in\mathcal{I}_i} \|\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t - \mathring{\mathbf{u}}_t\|_1$, where $\boldsymbol{\beta}_{\mathcal{I}_i} \in \mathbb{R}^{(k+1)\times d}$ is the best *static linear predictor* within the interval $\mathcal{I}_i$, and $\boldsymbol{\phi}_t \in \mathbb{R}^{k+1}$ is the k-th order dummy feature within interval $\mathcal{I}_i = [s_i, e_i]$ as defined in (21). Let $\boldsymbol{\Phi} \triangleq [\boldsymbol{\phi}_{s_i}, \dots, \boldsymbol{\phi}_{e_i}]^\top \in \mathbb{R}^{|\mathcal{I}_i|\times(k+1)}$ be the matrix of dummy features, and $\mathring{\mathbf{u}}_{[s_i,e_i]} = [\mathring{\mathbf{u}}_{s_i}, \dots, \mathring{\mathbf{u}}_{e_i}]^\top \in \mathbb{R}^{|\mathcal{I}_i|\times d}$, the best linear predictor $\boldsymbol{\beta}_{\mathcal{I}_i}$ is obtained by least square regression: $\boldsymbol{\beta}_{\mathcal{I}_i} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{(k+1)\times d}} \sum_{t\in\mathcal{I}_i} \|\boldsymbol{\beta}^\top\boldsymbol{\phi}_t - \mathring{\mathbf{u}}_t\|_2^2 = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\mathring{\mathbf{u}}_{[s_i,e_i]}$. Applying Cauchy-Schwarz inequality, we have

$$\sum_{t\in\mathcal{I}_i} \|\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t - \mathring{\mathbf{u}}_t\|_1 \le \sqrt{|\mathcal{I}_i| \cdot \sum_{t\in\mathcal{I}_i} \|\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t - \mathring{\mathbf{u}}_t\|_2^2}.$$

Next, we control the term $\sum_{t\in\mathcal{I}_i} \|\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t - \mathring{\mathbf{u}}_t\|_2^2$. By the orthogonality of CDJV wavelet transformation matrix W, we have

$$\begin{aligned}
&\sum_{t\in\mathcal{I}_i} \|\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t - \mathring{\mathbf{u}}_t\|_2^2 = \|\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\Phi}_{[s_i,e_i]} - \mathring{\mathbf{u}}_{[s_i,e_i]}\|_{\mathrm{F}}^2 = \|W\left(\boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\Phi}_{[s_i,e_i]} - \mathring{\mathbf{u}}_{[s_i,e_i]}\right)\|_{\mathrm{F}}^2 \\
&= \left\|\bar{\boldsymbol{\alpha}}_{[s_i,e_i]}\right\|_{\mathrm{F}}^2 = \left\|\boldsymbol{\alpha}_{[s_i,e_i]}\right\|_{\mathrm{F}}^2 \le 2\left\|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s_i,e_i]})\right\|_{\mathrm{F}}^2 + 2\left\|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s_i,e_i]}) - \boldsymbol{\alpha}_{[s_i,e_i]}\right\|_{\mathrm{F}}^2 \le 8\sigma^2 + 2\left\|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s_i,e_i]}) - \boldsymbol{\alpha}_{[s_i,e_i]}\right\|_{\mathrm{F}}^2,
\end{aligned} \tag{23}$$

where the first equality is due to the orthogonality of the CDJV wavelet bases, the second equality follows the definition of the wavelet transformation, and the fourth equality arises from Lemma 6. The first inequality is a consequence of $(a+b)^2 \le 2a^2+2b^2$, and the last inequality is due to the restart rule in Algorithm 1. We further introduce the following lemma, which is a modified version of Theorem 4 in Baby & Wang (2019), in order to bound the term $\|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s_i,e_i]}) - \boldsymbol{\alpha}_{[s_i,e_i]}\|_{\mathrm{F}}^2$:

Lemma 8 (Theorem 4 of Baby & Wang (2019))**.** *Consider an offline trend filtering problem $\widetilde{\mathbf{y}} = \mathring{\mathbf{y}} + Z$ (Donoho & Johnstone, 1998; Tibshirani, 2014), where $\mathring{\mathbf{y}} \in \mathbb{R}^{|\mathcal{I}_i|}$ is the underlying ground truth sequence, Z is a sub-Gaussian noise with variance σ^2, and $\widetilde{\mathbf{y}} \in \mathbb{R}^{|\mathcal{I}_i|}$ is the noise observation. Using Algorithm 1 and setting the threshold $\gamma = 4\sigma$, with probability at least $1 - 2/T^3$, the estimated wavelet coefficients $\widetilde{\boldsymbol{\alpha}}$ satisfies*

$$\left\|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s_i,e_i]}) - \boldsymbol{\alpha}_{[s_i,e_i]}\right\|_{\mathrm{F}}^2 \le 80 \cdot d\big(1 + \log T\big) \min_{\widehat{\mathbf{y}}} \max_{\mathring{\mathbf{y}}} \mathbb{E}\left[\|\widehat{\mathbf{y}} - \mathring{\mathbf{y}}\|_2^2\right].$$

Lemma 8 presents an intriguing insight: the gap between the estimated empirical wavelet coefficients $\widetilde{\boldsymbol{\alpha}}_{[s_i,e_i]}$ and the ground truth wavelet coefficients $\boldsymbol{\alpha}_{[s_i,e_i]}$ can be cast into the minimax rate of an *offline trend filtering* problem (Donoho & Johnstone, 1998; Tibshirani, 2014). Specifically, in offline trend filtering, an adversary selects a total of $|\mathcal{I}_i|$ underlying ground truth samples $\mathring{\mathbf{y}}_s, \dots, \mathring{\mathbf{y}}_e \in \mathbb{R}$, while the learner only observes noisy data samples $\widetilde{\mathbf{y}}_s, \dots, \widetilde{\mathbf{y}}_e \in \mathbb{R}$, each represented as $\widetilde{\mathbf{y}}_t = \mathring{\mathbf{y}}_t + Z$, with Z denoting sub-Gaussian noise with the variance of σ^2. The learner then denoises $\widetilde{\mathbf{y}}$ to obtain her prediction $\widehat{\mathbf{y}} \in \mathbb{R}^{|\mathcal{I}_i|}$. The learner's goal is to minimize the cumulative squared loss $\sum_{t\in\mathcal{I}_i} \|\widehat{\mathbf{y}}_t - \mathring{\mathbf{y}}_t\|_2^2$, where $\widehat{\mathbf{y}}_t$ represents the learner's prediction for the t-th data sample. We now introduce the following lemma to elucidate the minimax rate of the offline trend filtering problem.

Lemma 9 (Theorem 1 of Donoho & Johnstone (1998))**.** *Consider an offline trend filtering problem $\widetilde{\mathbf{y}} = \mathring{\mathbf{y}} + Z$, where $\mathring{\mathbf{y}} \in \mathbb{R}^{|\mathcal{I}_i|}$ is the underlying sequence whose path length is at most $P^k_{\mathcal{I}_i}$, Z is a sub-Gaussian noise with variance σ^2, and $\widetilde{\mathbf{y}} \in \mathbb{R}^{|\mathcal{I}_i|}$ is the noise observation. The minimax rate for the offline trend filtering problem is*

$$\min_{\widehat{\mathbf{y}}} \max_{\mathring{\mathbf{y}}} \mathbb{E}\left[\|\widehat{\mathbf{y}} - \mathring{\mathbf{y}}\|_2^2\right] = \widetilde{\mathcal{O}}\left(|\mathcal{I}_i|^{\frac{1}{2k+3}} (P^k_{\mathcal{I}_i})^{\frac{2}{2k+3}} \sigma^{\frac{2k+4}{2k+3}}\right).$$

We use Lemma 9 as a black box. Therefore, with probability at least $1 - 2/T$, we have

$$\begin{aligned}
C^k_{\mathcal{I}_i} &\le \sqrt{|\mathcal{I}_i| \cdot \left(8\sigma^2 + 2\left\|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s_i,e_i]}) - \boldsymbol{\alpha}_{[s_i,e_i]}\right\|_{\mathrm{F}}^2\right)} \\
&\lesssim \sqrt{|\mathcal{I}_i| \cdot \left(d\big(1 + \log T\big) \min_{\widehat{\mathbf{y}}} \max_{\mathring{\mathbf{y}}} \mathbb{E}\left[\left\|\widehat{\mathbf{y}} - \mathring{\mathbf{y}}\right\|_2^2\right] + 8\sigma^2\right)} \le \widetilde{\mathcal{O}}\sqrt{|\mathcal{I}_i| \cdot \left(|\mathcal{I}_i|^{\frac{1}{2k+3}} (P^k_{\mathcal{I}_i})^{\frac{2}{2k+3}}\right)} \\
&= \widetilde{\mathcal{O}}\sqrt{|\mathcal{I}_i| \cdot \left(|\mathcal{I}_i|^{\frac{1}{2k+3}} (|\mathcal{I}_i|^k \|\boldsymbol{D}^{k+1}\mathring{\mathbf{u}}_{[s_i,e_i]}\|_1)^{\frac{2}{2k+3}}\right)} = \widetilde{\mathcal{O}}\left(|\mathcal{I}_i|^{\frac{k+2}{2k+3}} (P^k_{\mathcal{I}_i})^{\frac{1}{2k+3}}\right),
\end{aligned} \tag{24}$$

where the $\lesssim$ ignores the constant of order k and logarithmic factors $\log T$. The second inequality is due to Lemma 8, and the third inequality is due to Lemma 9, which finishes the proof of Theorem 1. □

D.3. Proof of Theorem 2

Proof. We first decompose the dynamic regret as follows:

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) = \sum_{i=1}^M \left(\sum_{t\in\mathcal{I}_i} f_t(\boldsymbol{\theta}_t) - \sum_{t\in\mathcal{I}_i} f_t(\mathring{\mathbf{u}}_t) \right) \lesssim \sum_{i=1}^M \left(\sqrt{|\mathcal{I}_i|} + C_{\mathcal{I}_i}^k \right) \lesssim \sum_{i=1}^M \left(\sqrt{|\mathcal{I}_i|} + |\mathcal{I}_i|^{\frac{k+2}{2k+3}} (P_{\mathcal{I}_i}^k)^{\frac{1}{2k+3}} \right),$$

where the first inequality is due to the online algorithm $\mathcal{A}$ satisfying Requirement 1, the second inequality is due to Theorem 1. Besides, with probability at least $1 - 2/T$,

$$\begin{aligned} \sum_{i=1}^M \left(\sqrt{|\mathcal{I}_i|} + |\mathcal{I}_i|^{\frac{k+2}{2k+3}} (P_{\mathcal{I}_i}^k)^{\frac{1}{2k+3}} \right) &\le \sqrt{M} \left(\sum_{i=1}^M \left(\sqrt{|\mathcal{I}_i|} \right)^2 \right)^{1/2} + \left(\sum_{i=1}^M |\mathcal{I}_i| \right)^{\frac{k+2}{2k+3}} \left(\sum_{i=1}^M P_{\mathcal{I}_i} \right)^{\frac{1}{2k+3}} \\ &= \sqrt{TM} + T^{\frac{k+2}{2k+3}} (P_T^k)^{\frac{1}{2k+3}} \le \max \left(\sqrt{T \cdot T^{\frac{1}{2k+3}} (P_T^k)^{\frac{2}{2k+3}}}, \sqrt{T} \right) = \max \left\{ T^{\frac{k+2}{2k+3}} (P_T^k)^{\frac{1}{2k+3}}, \sqrt{T} \right\} \end{aligned}$$

where the first inequality is due to the Hölder inequality, and the second inequality is due to Theorem 1, which finishes the proof of Theorem 2. □

Optimality for Exp-concave and Strongly-convex Functions. For exp-concave and strongly convex functions, we can modify Requirement 1 to achieve the minimax optimal dynamic regret guarantee, which is illustrated in Requirement 2.

Requirement 2. *Suppose an online algorithm $\mathcal{A}$ is running over interval $\mathcal{I}_i = [s_i, e_i] \subseteq [T]$, it is required to satisfy*

$$\mathbf{Reg}_{\mathcal{I}_i}^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=s_i}^{e_i}) = \sum_{t=s_i}^{e_i} f_t(\boldsymbol{\theta}_t) - \sum_{t=s_i}^{e_i} f_t(\mathring{\mathbf{u}}_t) = \mathcal{O}\Big(1 + \sum_{t\in\mathcal{I}_i} \|\mathring{\mathbf{u}}_t - \boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t\|_1^2\Big).$$

We remark that Requirement 2 is easy to satisfy. For instance, when f_t is the squared loss, an online average algorithm suffices to meet this requirement. When f_t is an exponential concave function, an Online Newton Step (Hazan et al., 2007) algorithm can satisfy this requirement, as demonstrated in (27). Combining Theorem 1 and Requirement 2, we can obtain the following dynamic regret guarantee.

Theorem 6 (Overall Dynamic Regret for Exp-concave and Strongly Convex Functions)**.** *With probability at least $1 - 2/T$, using the detection-restart framework in Algorithm 1 with an online algorithm $\mathcal{A}$ satisfying Requirement 2 guarantees that*

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) = \widetilde{\mathcal{O}} \left(\max \left\{ T^{\frac{1}{2k+3}} (P_T^k)^{\frac{2}{2k+3}}, \sqrt{T} \right\} \right).$$

Proof of Theorem 6. We first decompose the dynamic regret as follows:

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) = \sum_{i=1}^M \left(\sum_{t\in\mathcal{I}_i} f_t(\boldsymbol{\theta}_t) - \sum_{t\in\mathcal{I}_i} f_t(\mathring{\mathbf{u}}_t) \right) \lesssim \sum_{i=1}^M \left(1 + \sum_{t\in\mathcal{I}_i} \|\mathring{\mathbf{u}}_t - \boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t\|_1^2 \right),$$

where the first inequality is due to the online algorithm $\mathcal{A}$ satisfying Requirement 2. For the second term $\sum_{t\in\mathcal{I}_i} \|\mathring{\mathbf{u}}_t - \boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t\|_1^2$. Following the proof of Theorem 1, we have

$$\sum_{t\in\mathcal{I}_i} \|\mathring{\mathbf{u}}_t - \boldsymbol{\beta}_{\mathcal{I}_i}^\top \boldsymbol{\phi}_t\|_1^2 \lesssim 8\sigma^2 + 2\left\|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s_i,e_i]}) - \boldsymbol{\alpha}_{[s_i,e_i]}\right\|_{\mathrm{F}}^2 \lesssim |\mathcal{I}_i|^{\frac{1}{2k+3}} (|\mathcal{I}_i|^k \|\boldsymbol{D}^{k+1} \mathring{\mathbf{u}}_{[s_i,e_i]}\|_1)^{\frac{2}{2k+3}} = |\mathcal{I}_i|^{\frac{1}{2k+3}} (P_{\mathcal{I}_i}^k)^{\frac{2}{2k+3}},$$

where the first inequality is due to (23), and the second inequality is due to (24). Therefore, with probability at least $1-2/T$,

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) \le \sum_{i=1}^M \left(1 + |\mathcal{I}_i|^{\frac{1}{2k+3}} (P_{\mathcal{I}_i}^k)^{\frac{2}{2k+3}} \right) \le M + T^{\frac{1}{2k+3}} (P_T^k)^{\frac{2}{2k+3}} \le \widetilde{\mathcal{O}} \left(\max \left\{ T^{\frac{1}{2k+3}} (P_T^k)^{\frac{2}{2k+3}}, \sqrt{T} \right\} \right),$$

where the second inequality is due to the Hölder inequality, and the third inequality is due to Theorem 1. Such a regret rate has been proved to be optimal for exponential concave and strongly convex functions (Baby & Wang, 2020; 2023). Therefore, we finish the proof of Theorem 6. □

Achieving Best Result of the Single-layer Model. We further prove that our method can simultaneously enjoy the previous best result of the single-layer model, i.e., $\mathcal{O}(\sqrt{T} \cdot P_T^0)$.

Proof of Achieving Best Result of the Single-layer Model. Considering we employ an OGD algorithm with $\eta_t = 1/\sqrt{t - s_i}$ within interval $\mathcal{I}_i = [s_i, e_i]$ as the online algorithm $\mathcal{A}$ in Algorithm 1, then we have

$$
\begin{aligned}
\sum_{t=s_i}^{e_i} f_t(\boldsymbol{\theta}_t) - \sum_{t=s_i}^{e_i} f_t(\mathring{\mathbf{u}}_t) &\leq \sum_{t=s_i}^{e_i} \frac{1}{2\eta_t}\Big(\|\mathring{\mathbf{u}}_t - \boldsymbol{\theta}_t\|_2^2 - \|\mathring{\mathbf{u}}_t - \boldsymbol{\theta}_{t+1}\|_2^2 \Big) + \sum_{t=s_i}^{e_i} \eta_t \|\nabla f_t(\boldsymbol{\theta}_t)\|_2^2 \\
&\leq \sum_{t=s_i+1}^{e_i} \frac{1}{2\eta_t}\Big(\|\mathring{\mathbf{u}}_t - \boldsymbol{\theta}_t\|_2^2 - \|\mathring{\mathbf{u}}_{t-1} - \boldsymbol{\theta}_t\|_2^2 \Big) + 4D^2 + \sum_{t=s_i}^{e_i} \eta_t G^2 \\
&\leq \sum_{t=s_i+1}^{e_i} \frac{1}{2\eta_t}\Big(\|\mathring{\mathbf{u}}_t - \mathring{\mathbf{u}}_{t-1}\|_2 \|\mathring{\mathbf{u}}_t - \boldsymbol{\theta}_t + \mathring{\mathbf{u}}_{t-1} - \boldsymbol{\theta}_t\|_2 \Big) + \Gamma^2 + \sum_{t=s_i}^{e_i} \eta_t G^2 \\
&\leq \frac{\Gamma}{2} \sum_{t=s_i+1}^{e_i} \frac{1}{2\eta_t} \|\mathring{\mathbf{u}}_t - \mathring{\mathbf{u}}_{t-1}\|_2 + \frac{\Gamma^2}{\eta_t} + G^2 \sum_{t=s_i}^{e_i} \eta_t \\
&\leq \frac{\Gamma\sqrt{d}}{2} (P_{\mathcal{I}_i}^0)\sqrt{|\mathcal{I}_i|} + \Gamma^2 \sqrt{|\mathcal{I}_i|} + G^2 \sqrt{|\mathcal{I}_i|} = \mathcal{O}\Big(\sqrt{|\mathcal{I}_i|}(P_{\mathcal{I}_i}^0 + 1) \Big),
\end{aligned}
$$

where $\Gamma \triangleq \sup_{\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2$ is the diameter of the decision space Θ, and G is the upper bound of the gradient of the loss function f_t. Summing across all the intervals, by setting the threshold $\gamma = 4T^{1/3}\sigma$, we have that $M \leq \widetilde{\mathcal{O}}((P_T^0)^{2/3})$ according to Theorem 1. Therefore, we have

$$
\begin{aligned}
\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) &\lesssim \sum_{i=1}^{M} \sqrt{|\mathcal{I}_i|} \cdot P_{\mathcal{I}_i}^0 + \sum_{i=1}^{M} \sqrt{|\mathcal{I}_i|} \leq \sqrt{\sum_{i=1}^{M} |\mathcal{I}_i| \cdot \sum_{i=1}^{M} (P_{\mathcal{I}_i}^0)^2} + \sum_{i=1}^{M} \sqrt{|\mathcal{I}_i|} \\
&\leq \sqrt{\sum_{i=1}^{M} |\mathcal{I}_i| \cdot \left(\sum_{i=1}^{M} P_{\mathcal{I}_i}^0 \right)^2} + \sqrt{T \cdot M} \leq \sqrt{T}(P_T^0 + 1),
\end{aligned}
$$

where the third inequality is due to $\sum_{i=1}^n x^2 \leq (\sum_{i=1}^n x)^2$, which finishes the proof. □

D.4. Proof of Theorem 3

Proof. Note that the CDJV wavelets comprise three components: (i) the left edge, (ii) the right edge, and (iii) the interior.

- The computation of wavelet coefficients for the left and right edges of CDJV wavelets involves matrix multiplication. Given that these edges encompass a transformation matrix of size $\mathcal{O}(kd)$, both computational and storage complexities are $\mathcal{O}(kd)$ per round.
- For the interior part, CDJV wavelets computation is essentially a convolution operation, see Eq. (14) for details. The wavelet coefficients are derived from convolving the input sequence with orthogonal wavelet bases. Therefore, the arrival of a new element impacts only a subset of coefficients ($\textsc{Update}_{\boldsymbol{\alpha}}(t)$). Besides, we only need to maintain the F-norm information for the detection module, allowing for the deletion of outdated coefficients ($\textsc{Drop}_{\boldsymbol{\alpha}}(t)$). Our wavelet operator utilizes a *binary indexed tree* to organize the coefficients. As outlined in Eq. (6) and (7), the coefficients requiring updates are in the set $\textsc{Update}_{\boldsymbol{\alpha}}(t)$, and those needing deletion are in $\textsc{Drop}_{\boldsymbol{\alpha}}(t)$, each with a size of $\mathcal{O}(\log T)$ per round. Additionally, updating a single coefficient incurs a computational and storage complexity of $\mathcal{O}(kd)$.
- Due to the benign property of CDJV wavelets, Lemma 6 formally states that the recenterlization will affect the calculated wavelet coefficients in our streaming wavelet operator, therefore we can remove the costly recentering operation which is necessary in previous methods. Besides, for the sequence that is not at a length of 2^k, our operator performs an "implicit padding" strategy to omit yet-to-arrive elements in the online sequence, which implicitly completes the sequence as a longer length. We adopt a perspective based on the wavelet coefficient analysis to show that how this "implicit padding" will not affect the norm of the wavelet coefficients, as elaborated in our Lemma 5.

Consequently, the overall computational and storage complexity of our streaming wavelet operator is $\mathcal{O}(kd \log T)$ each round. Here, k denotes the order of path length, and d represents the data dimension of the online sequence. □

D.5. Proof of Theorem 4

Proof. As outlined in Appendix C.3.1, we have reduced the regret of the expected risk R_t to the Kullback-Leibler (KL) divergence between the estimated label prior $\widehat{\boldsymbol{\mu}}_t$ and the underlying label prior $\boldsymbol{\mu}_t$, as illustrated in Lemma 3. Given that the KL-divergence is an exp-concave function, we further apply ONS (Hazan et al., 2007) to generate the estimated label prior $\widehat{\boldsymbol{\mu}}_t$ at each round t, where $\widehat{\boldsymbol{\mu}}_t = \mathbf{w}_t^\top \boldsymbol{\phi}_t$. In this part, we prove Theorem 4 by bounding the KL-divergence in Lemma 3, i.e.,

$$\begin{aligned}\sum_{t=s}^{e} \mathrm{KL}(\widehat{\boldsymbol{\mu}}_t \| \boldsymbol{\mu}_t) &= \sum_{t=s}^{e} \left(\sum_{j=1}^{K} [\widehat{\boldsymbol{\mu}}_t]_j \log \frac{[\widehat{\boldsymbol{\mu}}_t]_j}{[\boldsymbol{\mu}_t]_j} + [\boldsymbol{\mu}_t]_j - [\widehat{\boldsymbol{\mu}}_t]_j \right) \\ &= \sum_{t=s}^{e} \left(L_t^{\psi_{\mathsf{KL}}}(\widehat{\boldsymbol{\mu}}_t) - L_t^{\psi_{\mathsf{KL}}}(\boldsymbol{\mu}_t) \right) = \sum_{t=s}^{e} \left(L_t^{\psi_{\mathsf{KL}}}(\mathbf{w}_t^\top \boldsymbol{\phi}_t) - L_t^{\psi_{\mathsf{KL}}}(\boldsymbol{\mu}_t) \right),\end{aligned}$$

where the third equality is due to that, we use an ONS algorithm with dummy features as described in Appendix C.3.1, where we combine the dummy feature with a linear predictor $\mathbf{w}_t$, and let the input of ONS algorithm be $\widehat{\boldsymbol{\mu}}_t = \mathbf{w}_t^\top \boldsymbol{\phi}_t$. $\psi_{\mathsf{KL}}(x) = x \log x - x$ is the divergence function, and $L_t^{\psi_{\mathsf{KL}}}$ is the loss function induced by Bregman divergence as defined in (28). We first prove the following lemma to show that the reweighting-based update satisfies Requirement 1.

Lemma 10. *For any interval $\mathcal{I} = [s, e] \subseteq [T]$, the reweighting-update (19) and (10) running on the interval $\mathcal{I}$ ensures*

$$\mathbb{E}\left[\sum_{t \in \mathcal{I}} R_t(h_t) - \sum_{t \in \mathcal{I}} R_t(h_t^\star) \right] = \mathcal{O}(\sqrt{|\mathcal{I}|} + C_{\mathcal{I}}^k),$$

where $h_t^\star = \arg\min_{h \in \mathcal{H}} R_t(h)$ is the Bayesian optimal classifier defined in (17), and $C_{\mathcal{I}}^k = \sum_{t \in \mathcal{I}} \|\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t - \mathring{\mathbf{u}}_t\|_1$ is the k-order comparator gap defined in (22).

Proof of Lemma 10. Following (20), we decompose the regret of KL-divergence matching as follows:

$$L_t^{\psi_{\mathsf{KL}}}(\mathbf{w}_t^\top \boldsymbol{\phi}_t) - L_t^{\psi_{\mathsf{KL}}}(\boldsymbol{\mu}_t) = \underbrace{L_t^{\psi_{\mathsf{KL}}}(\mathbf{w}_t^\top \boldsymbol{\phi}_t) - L_t^{\psi_{\mathsf{KL}}}(\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t)}_{\texttt{term (a)}} + \underbrace{L_t^{\psi_{\mathsf{KL}}}(\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t) - L_t^{\psi_{\mathsf{KL}}}(\boldsymbol{\mu}_t)}_{\texttt{term (b)}}. \tag{25}$$

Then, we turn to analyze $\texttt{term}$ (a) and $\texttt{term}$ (b), respectively. In the rest of the proof, we use L_t as shorthand for $L_t^{\psi_{\mathsf{KL}}}$. We first decompose the $\texttt{term}$ (a) as follows:

$$\begin{aligned}\mathbb{E}\Big[L_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t) - L_t(\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t) \Big] \leq{}& \underbrace{\mathbb{E}\left[\langle \nabla L_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t) - \nabla \widetilde{L}_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t), \mathbf{w}_t^\top \boldsymbol{\phi}_t - \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t \rangle \right]}_{\texttt{term (a}_1\texttt{)}} \\ & + \underbrace{\mathbb{E}\left[\langle \nabla \widetilde{L}_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t), \mathbf{w}_t^\top \boldsymbol{\phi}_t - \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t \rangle - \frac{1}{\beta}(\nabla L_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t - \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t)) \right]}_{\texttt{term (a}_2\texttt{)}},\end{aligned}$$

where β is the lower bound of the label prior as defined in Lemma 2, and $\widetilde{L}_t : \Delta_K \to \mathbb{R}$ is the estimated KL-divergence using $\widetilde{\boldsymbol{\mu}}_t$ obtained by BBSE. Specifically,

$$\widetilde{L}_t(\boldsymbol{\mu}) = \sum_{j=1}^{K} (\partial(\psi_{\mathsf{KL}}[\boldsymbol{\mu}]_j)[\boldsymbol{\mu}]_j - \psi_{\mathsf{KL}}([\widetilde{\boldsymbol{\mu}}]_j)) - \sum_{j=1}^{K} \partial\psi_{\mathsf{KL}}([\boldsymbol{\mu}]_j)[\boldsymbol{\mu}_t]_j = \sum_{j=1}^{K} [\boldsymbol{\mu}]_j - \sum_{j=1}^{K} [\widetilde{\boldsymbol{\mu}}]_j \log([\boldsymbol{\mu}]_j).$$

Thus, $\widetilde{L}_t$ is an unbiased estimation of the ground truth L_t. For $\texttt{term}$ ($\texttt{a}_1$), we have that

$$\begin{aligned}\texttt{term (a}_1\texttt{)} &= \mathbb{E}_{1:e}\left[\langle \nabla L_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t) - \nabla \widetilde{L}_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t), \mathbf{w}_t^\top \boldsymbol{\phi}_t - \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t \rangle \right] \\ &= \mathbb{E}_{1:t-1}\left[\langle \nabla L_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t) - \mathbb{E}_t\left[\nabla \widetilde{L}_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t) \,\middle|\, 1 : t-1 \right], \mathbf{w}_t^\top \boldsymbol{\phi}_t - \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t \rangle \right] = 0,\end{aligned}$$

where the last equality is due to the unbiasedness of the risk estimator $\widetilde{L}_t$ such that $\nabla L_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t) = \mathbb{E}_t[\nabla \widetilde{L}_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t)|1:t-1]$. Thus, it is sufficient to analyze $\texttt{term (a}_2\texttt{)}$ to provide an upper bound for $\texttt{term (a)}$. For the $\texttt{term (a}_2\texttt{)}$, we have

$$\texttt{term (a}_2\texttt{)} = \mathbb{E}\left[\langle \nabla \widetilde{L}_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t), \mathbf{w}_t^\top \boldsymbol{\phi}_t - \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t \rangle - \frac{1}{\beta}(\nabla \widetilde{L}_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t - \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t))\right], \tag{26}$$

where we use the unbiasedness of the $\widetilde{L}_t$. The above term in (26) can be upper-bounded by the standard ONS analysis. We introduce the following lemma concerning the ONS algorithm:

Lemma 11 (Theorem 2 of Luo et al. (2016))**.** *Given that L_t is an exp-concave loss function, then the ONS algorithm enjoys the following regret bound for any comparator $\mathbf{w} \in \mathcal{W}$,*

$$\sum_{i=s}^{e} L_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t) - \sum_{i=s}^{e} L_t(\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t) \leq \frac{\lambda}{2}\|\boldsymbol{\beta}_{\mathcal{I}}\|_2^2 + \frac{d}{2\varepsilon}\log\left(1 + \frac{\varepsilon|\mathcal{I}|G^2}{d\lambda}\right),$$

where the λ and ε are the parameters of the ONS algorithm defined in (19).

It is easy to verify that $L_t(\mathbf{w}^\top \boldsymbol{\phi}_t)$ is an exp-concave function w.r.t. $\mathbf{w}$. To control the norm $\|\boldsymbol{\beta}_{\mathcal{I}}\|_2^2$, we have

Lemma 12 (Corollary 40 of Baby & Wang (2020))**.** *Let $\boldsymbol{\beta}_{\mathcal{I}}$ denote the best static linear predictor, which is obtained by least square regression: $\boldsymbol{\beta}_{\mathcal{I}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{(k+1)\times d}} \sum_{t \in \mathcal{I}} \|\boldsymbol{\beta}^\top \boldsymbol{\phi}_t - \mathring{\mathbf{u}}_t\|_2^2$, where $\boldsymbol{\phi}_t$ is the dummy feature defined in (21) and $\forall t \in [T]$, the 1-norm of $\mathring{\mathbf{u}}_t$ has an upper bound, then we have $\|\boldsymbol{\beta}_{\mathcal{I}}\|_2^2 = \mathcal{O}(1)$.*

Combining Lemma 12 with Lemma 11, we can therefore control the summation of $\texttt{term (a)}$ as

$$\sum_{t=s}^{e} L_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t) - \sum_{t=s}^{e} L_t(\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t) \leq \widetilde{\mathcal{O}}\left(d \log\left(1 + \frac{|\mathcal{I}|}{d}\right)\right).$$

For the $\texttt{term (b)}$, notice that for the KL-divergence, $\nabla^2 L_t(\mathbf{w}^\top \phi) = \nabla^2 L_t(\boldsymbol{\mu}) = \sum_{j=1}^{K} [\boldsymbol{\mu}_t]_j / [\boldsymbol{\mu}]_j^2 \leq K/\beta^2$, which implies it is a K/β^2-smooth function. By the smoothness of the loss function L_t, we decompose $\texttt{term (b)}$ as follows:

$$\begin{aligned}\sum_{t=s}^{e} L_t(\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t) - \sum_{t=s}^{e} L_t(\boldsymbol{\mu}_t) &\leq \sum_{t=s}^{e} \langle \nabla L_t(\boldsymbol{\mu}_t), \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t - \boldsymbol{\mu}_t \rangle + \sum_{t=s}^{e} \frac{K}{2\beta^2}\|\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t - \boldsymbol{\mu}_t\|_2^2 \\ &\leq \sum_{t=s}^{e} \langle \nabla L_t(\boldsymbol{\mu}_t), \boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t - \boldsymbol{\mu}_t \rangle + \sum_{t=s}^{e} \frac{K^{3/2}}{2\beta^2}\|\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t - \boldsymbol{\mu}_t\|_1^2.\end{aligned}$$

Due to the ground-truth label prior $\boldsymbol{\mu}_t$ is minimizer of L_t, the first term equals zero. Thus, summation of $\texttt{term (b)}$ satisfies $\sum_{t=s}^{e} L_t(\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t) - \sum_{t=s}^{e} L_t(\boldsymbol{\mu}_t) \leq \widetilde{\mathcal{O}}\left(\sum_{t=s}^{e} \|\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t - \boldsymbol{\mu}_t\|_1^2\right)$. Combining $\texttt{term (a)}$ and $\texttt{term (b)}$ in Eq. (25), we have

$$\sum_{t=s}^{e} L_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t) - \sum_{t=s}^{e} L_t(\boldsymbol{\mu}_t) \leq \widetilde{\mathcal{O}}\left(1 + \sum_{t=s}^{e} \|\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t - \boldsymbol{\mu}_t\|_1^2\right). \tag{27}$$

Following Proposition 3, we obtain

$$\begin{aligned}&\mathbb{E}\left[\mathbf{Reg}_{\mathcal{I}}^{\mathbf{d}}(\{R_t, h_t^\star\}_{t=s}^{e})\right] \leq \sum_{t=s}^{e} L\sqrt{\|\widehat{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_t\|_2^2} \leq \mathcal{O}\left(\sum_{t=s}^{e} \sqrt{\mathrm{KL}(\widehat{\boldsymbol{\mu}}_t \| \boldsymbol{\mu}_t)}\right) \leq \mathcal{O}\left(\sum_{t=s}^{e} \sqrt{(\texttt{term (a)})_t + \|\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t - \boldsymbol{\mu}_t\|_1^2}\right) \\ &\leq \mathcal{O}\left(\sum_{t=s}^{e} \left(\sqrt{(\texttt{term (a)})_t} + \|\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t - \boldsymbol{\mu}_t\|_1\right)\right) \leq \mathcal{O}\left(\sqrt{|\mathcal{I}|(\texttt{term (a)})_t} + \sum_{t=s}^{e} \|\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t - \boldsymbol{\mu}_t\|_1\right) = \widetilde{\mathcal{O}}(\sqrt{|\mathcal{I}|} + C_{\mathcal{I}}^k),\end{aligned}$$

where the first inequality is due to Lemma 2, the second inequality is due to Proposition 3, $(\texttt{term (a)})_t$ represents $L_t(\mathbf{w}_t^\top \boldsymbol{\phi}_t) - L_t(\boldsymbol{\beta}_{\mathcal{I}}^\top \boldsymbol{\phi}_t)$ at time t in the decomposition (25). Thus, we finish the proof of Lemma 10. □

Lemma 10 indicates that the reweighting updates (10) satisfy Requirement 1. Combining Lemma 10 and Theorem 2, we have the following guarantee for reweighting update:

$$\mathbb{E}\left[\mathbf{Reg}_T^{\mathbf{d}}(\{R_t, h_t^\star\}_{t=1}^{T})\right] \leq \widetilde{\mathcal{O}}\left(\max\{T^{\frac{k+2}{2k+3}}(P_T^k)^{\frac{1}{2k+3}}, \sqrt{T}\}\right),$$

which finishes the proof of Theorem 4. □

D.6. Proof of Theorem 5

Proof. We first introduce the following interval regret for the OGD updates.

Lemma 13. *For any interval $\mathcal{I} = [s, e] \subseteq [T]$, setting the step size as $\eta_t = 1/\sqrt{t-s}$, the OGD updates* (11) *running on the interval $\mathcal{I}$ ensures*

$$\mathbb{E}\left[\sum_{t\in\mathcal{I}} R_t(\mathbf{w}_t) - \sum_{t\in\mathcal{I}} R_t(\mathbf{w}_t^\star)\right] \leq \mathcal{O}\left(\sqrt{|\mathcal{I}|} + C_{\mathcal{I}}^0\right),$$

where $\mathbf{w}_t^\star = \arg\min_{\mathbf{w}\in\mathcal{W}} R_t(\mathbf{w})$ is the minimizer of the expected risk function, $C_{\mathcal{I}}^0 = \sum_{i=s}^{e} \|\boldsymbol{\mu}_{\mathcal{I}}^\star - \boldsymbol{\mu}_i\|_1$ is the first-order comparator gap defined in (22)*, in which $\boldsymbol{\mu}_{\mathcal{I}}^\star = \frac{1}{|\mathcal{I}|}\sum_{i=e}^{s} \boldsymbol{\mu}_i$ is the static label prior within interval $\mathcal{I}$.*

Proof of Lemma 13. Lemma 13 indicates that the OGD updates (11) satisfy Requirement 1. The proof is similar to (Bai et al., 2022, Theorem 1). We can decompose the regret bound into two parts by introducing a reference comparator $\mathbf{w}_{\mathcal{I}}^\star$ that is static within the interval $\mathcal{I}$ following the decomposition (20), taken as the single best decision over the interval, i.e., $\mathbf{w}_{\mathcal{I}}^\star = \arg\min_{\mathbf{w}\in\mathcal{W}} \sum_{t\in\mathcal{I}} R_t(\mathbf{w})$. We have

$$\mathbb{E}_{1:e}\left[\sum_{t=s}^{e} R_t(\mathbf{w}_t)\right] - \sum_{t=s}^{e} R_t(\mathbf{w}_t^\star) = \underbrace{\mathbb{E}_{1:e}\left[\sum_{t\in\mathcal{I}} R_t(\mathbf{w}_t)\right] - \sum_{t\in\mathcal{I}} R_t(\mathbf{w}_{\mathcal{I}}^\star)}_{\texttt{term (a)}} + \underbrace{\sum_{t\in\mathcal{I}} R_t(\mathbf{w}_{\mathcal{I}}^\star) - \sum_{t\in\mathcal{I}} R_t(\mathbf{w}_t^\star)}_{\texttt{term (b)}},$$

where $\mathbb{E}_{1:e}[\cdot]$ denotes the expectation taken over the random draw of dataset $\{S_t\}_{t=1}^{e}$. Then, we turn to analyze $\texttt{term (a)}$ and $\texttt{term (b)}$, respectively. We first show that $\texttt{term (a)}$ can be decomposed as

$$\texttt{term (a)} \leq \underbrace{\mathbb{E}_{1:e}\left[\sum_{t=s}^{e}\langle \nabla R_t(\mathbf{w}_t) - \nabla\widehat{R}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_{\mathcal{I}}^\star\rangle\right]}_{\texttt{term (a}_1\texttt{)}} + \underbrace{\mathbb{E}_{1:e}\left[\sum_{t=s}^{e}\langle \nabla\widehat{R}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_{\mathcal{I}}^\star\rangle\right]}_{\texttt{term (a}_2\texttt{)}},$$

which is due to the convexity of the risk function $R_t(\cdot)$. For $\texttt{term (a}_1\texttt{)}$, we have

$$\texttt{term (a}_1\texttt{)} = \sum_{t=s}^{e} \mathbb{E}_{1:t-1}\left[\left\langle \nabla R_t(\mathbf{w}_t) - \mathbb{E}_t\left[\nabla\widehat{R}_t(\mathbf{w}_t)\,\middle|\, 1:t-1\right], \mathbf{w}_t - \mathbf{w}_{\mathcal{I}}^\star\right\rangle\right] = 0,$$

where the last equality is due to the unbiasedness of the risk estimator $\widehat{R}_t$ such that $\nabla R_t(\mathbf{w}_t) = \mathbb{E}_t[\nabla\widehat{R}_t(\mathbf{w}_t) \mid 1:t-1]$. Thus, it is sufficient to analyze $\texttt{term (a}_2\texttt{)}$ to provide an upper bound for $\texttt{term (a)}$. For the $\texttt{term (a}_2\texttt{)}$, we use the standard OGD analysis with a static comparator $\mathbf{w}_{\mathcal{I}}^\star$ (Hazan, 2016), and can achieve the static regret within the interval $\mathcal{I}$: $\texttt{term (a}_2\texttt{)} = \mathcal{O}(\sqrt{e-s}) = \mathcal{O}(\sqrt{\mathcal{I}})$. For the $\texttt{term (b)}$, we decompose it as follows:

$$\begin{aligned}
\texttt{term (b)} &= \sum_{t\in\mathcal{I}}\left(\frac{1}{|\mathcal{I}|}\sum_{i=s}^{e} R_t(\mathbf{w}_{\mathcal{I}}^\star) - R_t(\mathbf{w}_t^\star)\right) \leq \sum_{t\in\mathcal{I}}\left(\frac{1}{|\mathcal{I}|}\sum_{i=s}^{e} R_t(\mathbf{w}_i^\star) - R_t(\mathbf{w}_t^\star)\right)\\
&= \sum_{t\in\mathcal{I}}\left(\frac{1}{|\mathcal{I}|}\sum_{i=s}^{e} R_t(\mathbf{w}_i^\star) - \frac{1}{|\mathcal{I}|}\sum_{i=s}^{e} R_i(\mathbf{w}_i^\star) + \frac{1}{|\mathcal{I}|}\sum_{i=s}^{e} R_i(\mathbf{w}_i^\star) - R_t(\mathbf{w}_t^\star)\right)\\
&\leq \sum_{t\in\mathcal{I}}\left(\frac{1}{|\mathcal{I}|}\sum_{i=s}^{e} R_t(\mathbf{w}_i^\star) - \frac{1}{|\mathcal{I}|}\sum_{i=s}^{e} R_i(\mathbf{w}_i^\star) + \frac{1}{|\mathcal{I}|}\sum_{i=s}^{e} R_i(\mathbf{w}_t^\star) - R_t(\mathbf{w}_t^\star)\right) \leq 2\sum_{t\in\mathcal{I}}\sup_{\mathbf{w}\in\mathcal{W}}\left|R_t(\mathbf{w}) - \frac{1}{|\mathcal{I}|}\sum_{i=s}^{e} R_i(\mathbf{w})\right|.
\end{aligned}$$

In the above, the first inequality is due to the optimality of $\mathbf{w}_{\mathcal{I}}^\star$ over the interval $\mathcal{I}$. The second inequality holds since $\mathbf{w}_i^\star \in \arg\min_{\mathbf{w}\in\mathcal{W}} R_i(\mathbf{w})$. According to the label shift condition, we can further upper bound the variation of the loss function by the variation of the class prior

$$\sup_{\mathbf{w}\in\mathcal{W}}\left|R_t(\mathbf{w}) - \frac{1}{|\mathcal{I}|}\sum_{i=s}^{e} R_i(\mathbf{w})\right| = \sup_{\mathbf{w}\in\mathcal{W}}\left|\sum_{k=1}^{K}\left([\boldsymbol{\mu}_t]_k - \frac{1}{|\mathcal{I}|}\sum_{i=s}^{e}[\boldsymbol{\mu}_i]_k\right)R_0^k(\mathbf{w})\right| \leq B\sum_{k=1}^{K}\left|[\boldsymbol{\mu}_t]_k - \frac{1}{|\mathcal{I}|}\sum_{i=s}^{e}[\boldsymbol{\mu}_i]_k\right| = B\|\boldsymbol{\mu}_t - \boldsymbol{\mu}_{\mathcal{I}}^\star\|_1,$$

in which $\boldsymbol{\mu}_{\mathcal{I}}^\star = \frac{1}{|\mathcal{I}|}\sum_{i=e}^{s} \boldsymbol{\mu}_i$ is the static label prior in interval $\mathcal{I}$. By summing up `term (b)`, we have

$$\sum_{t\in\mathcal{I}} R_t(\mathbf{w}_t) - \sum_{t\in\mathcal{I}} R_t(\mathbf{w}_t^\star) = \sqrt{|I|} + \sum_{t=s}^{e} B\left\|\boldsymbol{\mu}_t - \boldsymbol{\mu}_{\mathcal{I}}^\star\right\|_1 = \sqrt{|\mathcal{I}|} + B\cdot C_{\mathcal{I}}^0,$$

which finishes the proof of Lemma 13. □

Lemma 13 illustrates that OGD updates (11) satisfy Requirement 1. Therefore, we have

$$\mathbb{E}[\mathbf{Reg}_T^{\mathbf{d}}(\{R_t, h_t^\star\}_{t=1}^T)] \leq \sum_{i=1}^{M}\left(\sqrt{|\mathcal{I}_i|} + C_{\mathcal{I}}^0\right) = \mathcal{O}\left(\sum_{i=1}^{M}\left(\sqrt{|\mathcal{I}_i|} + |\mathcal{I}_i|^{2/3}(P_{\mathcal{I}_i}^0)^{1/3}\right)\right) \leq \mathcal{O}\left(\max\left\{T^{2/3}(P_T^0)^{1/3}, \sqrt{T}\right\}\right),$$

where the second inequality is by taking the $k=0$ in Theorem 2, which finishes the proof. This result achieves the optimal dynamic regret ignoring the dimension factor, which is the same as the previous result (Bai et al., 2022) but only requires maintaining a single classifier. □

D.7. Proof of Lemma 3

Proof. To prove Lemma 3, we first introduce a general Bregman divergence label margin matching framework, then instantiate it as the KL-divergence matching problem.

Bregman Divergence Label Margin Matching. Bregman divergence label margin matching (Sugiyama et al., 2012) is a general framework for density ratio estimation that unifies various models developed in one-step distribution shift. Specifically, let $\psi : \mathrm{dom}\,\psi \to \mathbb{R}$ be a differentiable and strictly convex function; the Bregman divergence measures distance (discrepancy) between two points $\mathcal{D}_\psi(a,b) \triangleq \psi(a) - \psi(b) - \nabla\psi(b)(a-b)$, where $\nabla\psi$ is the derivative of ψ.

Then, we measure the discrepancy between true label margin $\boldsymbol{\mu}_t$ and any label margin estimator $\widehat{\boldsymbol{\mu}}_t$ by the total Bregman divergence over K classes, that is, $\mathrm{TD}_\psi(\boldsymbol{\mu}_t, \widehat{\boldsymbol{\mu}}_t) = \sum_{j=1}^K \mathcal{D}_\psi\left([\boldsymbol{\mu}_t]_j, [\widehat{\boldsymbol{\mu}}_t]_j\right)$. A direct calculation shows $\mathrm{TD}_\psi(\boldsymbol{\mu}_t, \widehat{\boldsymbol{\mu}}_t) = L_t^\psi(\widehat{\boldsymbol{\mu}}_t) - L_t^\psi(\boldsymbol{\mu}_t)$, where the loss function is defined as

$$L_t^\psi(\boldsymbol{\mu}) = \sum_{j=1}^{K}(\nabla\psi([\boldsymbol{\mu}]_j)[\boldsymbol{\mu}]_j - \psi([\boldsymbol{\mu}]_j)) - \sum_{k\sim D_t(y)} \nabla\psi([\boldsymbol{\mu}]_j) = \sum_{j=1}^{K}(\nabla\psi([\boldsymbol{\mu}]_j)[\boldsymbol{\mu}]_j - \psi([\boldsymbol{\mu}]_j)) - \sum_{j=1}^{K}\nabla\psi([\boldsymbol{\mu}]_j)[\boldsymbol{\mu}_t]_j, \tag{28}$$

where $\boldsymbol{\mu}_t$ is the true label margin at round t. Suppose that ψ is μ-strongly convex, then we have the following proposition.

Proposition 3 (Theorem 1 of Zhang et al. (2023a))**.** *Let ψ be a μ-strongly convex function, for any label margin estimator sequence $\{\widehat{\boldsymbol{\mu}}_t\}_{t=1}^T$, the cumulative estimation error is bounded by*

$$\sum_{t=s}^{s}\|\widehat{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_t\|_2 \leq \sqrt{\frac{2|\mathcal{I}|}{\mu}\left(\sum_{t=s}^{e} L_t^\psi(\widehat{\boldsymbol{\mu}}_t) - \sum_{t=s}^{e} L_t^\psi(\boldsymbol{\mu}_t)\right)}. \tag{29}$$

Instantiation: KL-Divergence Matching Model. When the online loss function L_t^ψ is non-convex, it is generally intractable to conduct the online optimization, no matter whether to minimize the standard regret or the strengthened dynamic regret. Fortunately, the attained loss functions are convex or enjoy even stronger curvature when we choose suitable hypothesis space and divergence functions. To this end, we instantiate the reduction of Proposition 3 via the KL-Divergence matching model (Garg et al., 2020). When the divergence function is chosen as $\psi_{\mathsf{KL}}(x) = x\log x - x$, in this case we have $L_t^{\psi_{\mathsf{KL}}}(\boldsymbol{\mu}) = \sum_{j=1}^K [\boldsymbol{\mu}]_j - \sum_{j=1}^K [\log(\boldsymbol{\mu})]_j [\boldsymbol{\mu}_t]_j$, and therefore, we have $\mathrm{KL}(\boldsymbol{\mu}\|\boldsymbol{\mu}_t) = L_t^{\psi_{\mathsf{KL}}}(\boldsymbol{\mu}) - L_t^{\psi_{\mathsf{KL}}}(\boldsymbol{\mu}_t)$. It is easy to verify that $\nabla^2\psi_{\mathsf{KL}}(x) = 1/x$, and $\nabla^3\psi_{\mathsf{KL}}(x) = -1/x^2$. Therefore, ψ_{KL} is a $1/\beta$-strongly convex function given that the inputs have a lower bound β. Thus, by Proposition 3,

$$\mathbf{Reg}_{\mathcal{I}}^{\mathbf{d}}(\{R_t, h_t^\star\}_{t=s}^e) \leq \sqrt{2\beta|\mathcal{I}|\cdot\left(\sum_{t=s}^{e} L_t^{\psi_{\mathsf{KL}}}(\widehat{\boldsymbol{\mu}}_t) - \sum_{t=s}^{e} L_t^{\psi_{\mathsf{KL}}}(\boldsymbol{\mu}_t)\right)} = \mathcal{O}\left(\sqrt{|\mathcal{I}|\cdot\sum_{t=s}^{e}\mathrm{KL}(\widehat{\boldsymbol{\mu}}_t\|\boldsymbol{\mu}_t)}\right),$$

which finishes the proof of Lemma 3. □