Adapting to Online Label Shift with Provable Guarantees

Yong Bai[1*], **Yu-Jie Zhang**[2,1*], **Peng Zhao**[1], **Masashi Sugiyama**[3,2], **Zhi-Hua Zhou**[1†]
[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2] The University of Tokyo, Chiba, Japan
[3] RIKEN AIP, Tokyo, Japan

Abstract

The standard supervised learning paradigm works effectively when training data shares the same distribution as the upcoming testing samples. However, this stationary assumption is often violated in real-world applications, especially when testing data appear in an online fashion. In this paper, we formulate and investigate the problem of *online label shift* (OLaS): the learner trains an initial model from the labeled offline data and then deploys it to an unlabeled online environment where the underlying label distribution changes over time but the label-conditional density does not. The non-stationarity nature and the lack of supervision make the problem challenging to be tackled. To address the difficulty, we construct a new unbiased risk estimator that utilizes the unlabeled data, which exhibits many benign properties albeit with potential non-convexity. Building upon that, we propose novel online ensemble algorithms to deal with the non-stationarity of the environments. Our approach enjoys optimal *dynamic regret*, indicating that the performance is competitive with a clairvoyant who knows the online environments in hindsight and then chooses the best decision for each round. The obtained dynamic regret bound scales with the intensity and pattern of label distribution shift, hence exhibiting the adaptivity in the OLaS problem. Extensive experiments are conducted to validate the effectiveness and support our theoretical findings.

1 Introduction

One of the fundamental challenges for modern machine learning is the distribution shift [1, 2, 3, 4]. The learned model's testing performance would significantly drop when the distribution is different from the initial training distribution. More severely, in many real-world applications, testing data often come in an *online* fashion after deploying the trained model such that the underlying distribution might continuously change over time. Hence, it is necessary to develop learning methods to handle distribution shift in online and open environments [5]. Another practical concern is the *label scarcity* issue in real tasks, particularly those tasks emerging in online scenarios. For example, in the species monitoring task [6, 7], a learned model is deployed to detect species of wild animals. The data consist of received signals from sensors and hence are naturally in the streaming form. The data distribution of upcoming animals will change due to the variety of species across different geographic locations and seasons, and moreover, it is hard to gather the labels of streaming data in time.

Motivated by the above real demand, this paper is concerned with the following problem: how to design an algorithm that can adapt to non-stationary environments with a few labeled data or even unlabeled data observed at every time? In addition to getting empirical performance gain, the overall method is desired to have clear and strong theoretical guarantees. The problem is generally hard due

[*]Equal contribution.
[†]Correspondence to: Zhi-Hua Zhou <zhouzh@lamda.nju.edu.cn>

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

to the non-stationarity of online environments and the lack of supervision. As such, we investigate a simplified case with a focus on the specific change on the *label distribution*. We formalize the *online label shift* (OLaS) problem,[3] which consists of two stages, including offline initialization and online adaptation. Specifically, the learner collects labeled samples drawn independently from the initial distribution $\mathcal{D}_0(\mathbf{x}, y)$ where $\mathbf{x}$ and y denote the feature vector and its associated label, and trains an initial model following standard supervised learning methods. Subsequently, she needs to adapt it to an unknown non-stationary environment where the underlying label distributions change over time. Specifically, at time t, she receives a few *unlabeled* samples drawn from the current distribution $\mathcal{D}_t(\mathbf{x})$ and uses them to update the model. In OLaS problems, the essential environment change happens on the label distribution $\mathcal{D}_t(y)$ with the conditional $\mathcal{D}(\mathbf{x} \mid y)$ always remaining the same.

The label shift problem has been widely studied in the offline setting [8, 9, 10, 11, 12, 13, 14], but this is less explored in the more challenging online setup. One natural impulse is to handle OLaS by online learning techniques [15], but it is generally non-trivial due to the lack of supervision in the adaptation stage and also the non-stationarity issue. Wu et al. [16] made the first such attempt for OLaS, where they constructed an unbiased risk estimator with the unlabeled data for model assessment and used online gradient descent for model updating. Let T denote the number of rounds. They proved an $\mathcal{O}(\sqrt{T})$ regret bound, which measures the gap between the learner's decision and the best *fixed* decision in hindsight. However, in non-stationary environments, a single decision can hardly perform well all the time, which makes the guarantee less attractive for OLaS problems. Another technical caveat is that their theory relies on a vital assumption of the convexity of risk functions, which was not verified strictly. In fact, this assumption can hardly be satisfied as an operation to take the argument of the maximum is involved in its formulation of the risk estimator.

In this paper, we aim to develop an algorithm for adapting to online label shift with *provable* guarantees. To this end, we first reframe the construction of the unbiased risk estimator via risk rewriting techniques and prove that the estimator still enjoys benign theoretical properties, albeit with a potential *non-convex* behavior. Second, to handle the non-stationarity of the online stream, instead of using traditional regret as the performance measure, we employ *dynamic regret* to guide the algorithm design, which ensures the online algorithm is competitive with a clairvoyant who knows the online functions in hindsight and hence chooses the best decision of each round. To optimize such a strengthened measure, we propose a novel online ensemble algorithm building upon the risk estimator, consisting of a meta-algorithm running over a group of base-learners, each associated with a customized configuration. Our algorithm enjoys an $\mathcal{O}(V_T^{1/3}T^{2/3})$ dynamic regret, where $V_T = \sum_{t=2}^{T} \|\boldsymbol{\mu}_{y_t} - \boldsymbol{\mu}_{y_{t-1}}\|_1$ measures the variation of the label distributions, with $\boldsymbol{\mu}_{y_t}$ denoting the vector consisting of the class-prior probabilities at time t . Notably, the obtained dynamic regret upper bound is *minimax optimal* in terms of both T and V_T, and importantly, our algorithm does not require the unknown non-stationarity measure V_T as the input.

Furthermore, for many situations where online label shift contains some patterns such as periodicity or gradual change behavior, we present an improved algorithm to exploit such structures and achieve provably better guarantees. The key idea is to leverage historical information to serve as a hint for online updates. We prove an $\mathcal{O}(V_T^{1/3}G_T^{1/3}T^{1/3})$ dynamic regret, where G_T measures the reusability of historical information that is at most $\mathcal{O}(T)$ while could be much smaller in benign environments. As a benefit, the improved algorithm safeguards the $\mathcal{O}(V_T^{1/3}T^{2/3})$ worst-case optimality and meanwhile achieves great improvement in easier environments. Extensive experiments are conducted to evaluate our approach, which show the usefulness of meta-base structure in tackling non-stationarity and validate the effectiveness of other adaptive components.

Technical Contribution. Our method is not a direct application of existing non-stationary online learning methods [17, 18, 19], but rather requires in-depth technical innovations. First, the potentially *non-convex* risk estimator makes it hard to apply existing techniques of online convex optimization (OCO) [15], but fortunately, we prove the convexity of its expectation such that the OCO framework can still be used (see Remark 2). Second, to optimize the dynamic regret, we employ a meta-base structure to hedge the uncertainty of the *unknown* minimizer of the expected risk function at each round and convert the variation of expected risk minimizers to the intensity of label distribution drifts, a natural non-stationarity measure for the OLaS problem (see Remark 3). Third, earlier study showed adaptive dynamic regret bounds for convex and smooth functions [19], while the smoothness

[3] We use OLaS instead of OLS, since OLS often refers to "ordinary least squares".

assumption can hardly be satisfied in our case. We remove such a constraint by introducing an *implicit update*, which could be of independent interest for general OCO purposes (see Remark 4).

2 Problem Formulation

We focus on OLaS of multi-class classification with feature space $\mathcal{X} \subseteq \mathbb{R}^d$ and label space $\mathcal{Y} = [K] \triangleq \{1, \ldots, K\}$, where d is the dimension and K is the number of classes. Below we formulate setups of two stages of OLaS (offline initialization and online adaptation).

Problem Setup. In the *offline initialization stage*, the learner collects a number of labeled samples denoted by $S_0 = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N_0}$ drawn from the distribution $\mathcal{D}_0(\mathbf{x}, y)$ and then obtains a well-performed initial model $f_0 : \mathcal{X} \mapsto \mathcal{Y}$. In the *online adaptation stage*, data come in the streaming form *without* labels. The current model is deployed to predict the labels of online data and also to evolve adaptively. Specifically, at each round $t \in [T]$, the learner receives a small number of *unlabeled* data $S_t = \{\mathbf{x}_n\}_{n=1}^{N_t}$ drawn from the distribution $\mathcal{D}_t(\mathbf{x})$. The non-stationary nature indicates $\mathcal{D}_t \neq \mathcal{D}_{t'}$ in general for different $t, t' \in [T]$. OLaS considers the simplified case that essential changes come from label distributions and there are no new classes, formally described in the following condition.

Assumption 1 (Online Label Shift)**.** In the online label shift problem, the label distribution $\mathcal{D}_t(y)$ changes over time while the class-conditional distribution $\mathcal{D}_t(\mathbf{x} \,|\, y)$ is identical throughout the process for $t \in \{0, 1, \ldots, T\}$. Moreover, it holds that $\mathcal{D}_0(y) > 0$ for any $y \in \mathcal{Y}$.

Performance Measure. At round $t \in [T]$, the learner uses the information observed so far to make the prediction and also update the model $\mathbf{w}_t \in \mathcal{W}$, where $\mathcal{W}$ is a convex decision set with diameter $\Gamma \triangleq \sup_{\mathbf{w}, \mathbf{w}' \in \mathcal{W}} \|\mathbf{w} - \mathbf{w}'\|_2$. The goal is to ensure that the t-round model $\mathbf{w}_t$ generalizes well on the underlying distribution $\mathcal{D}_t$. Thus, the model's quality is evaluated by its risk defined as $R_t(\mathbf{w}) = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_t}[\ell(f(\mathbf{w}, \mathbf{x}), y)]$, where $f : \mathcal{W} \times \mathcal{X} \mapsto \mathbb{R}^K$ is the predictive function and $\ell : \mathbb{R}^K \times \mathcal{Y} \mapsto \mathbb{R}$ is any convex surrogate loss for classification such that $\ell(f(\mathbf{w}, \mathbf{x}), y)$ is convex in $\mathbf{w}$. We introduce two constants, $G \triangleq \sup_{(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}, \mathbf{w} \in \mathcal{W}} \|\nabla_{\mathbf{w}} \ell(f(\mathbf{w}, \mathbf{x}), y)\|_2$ and $B \triangleq \sup_{(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}, \mathbf{w} \in \mathcal{W}} |\ell(f(\mathbf{w}, \mathbf{x}), y)|$, as upper bounds of gradient norm and loss function value.

We use regret to examine the performance of online algorithms. In particular, *dynamic regret* [20, 17] is employed to compete the algorithm's performance with the best response at each round, defined as

$$\mathbf{Reg}_T^{\mathbf{d}} \triangleq \sum_{t=1}^{T} R_t(\mathbf{w}_t) - \sum_{t=1}^{T} \min_{\mathbf{w} \in \mathcal{W}} R_t(\mathbf{w}) = \sum_{t=1}^{T} R_t(\mathbf{w}_t) - \sum_{t=1}^{T} R_t(\mathbf{w}_t^*), \tag{1}$$

where $\mathbf{w}_t^* \in \arg\min_{\mathbf{w} \in \mathcal{W}} R_t(\mathbf{w})$ is the model (or one of the models) with the best generalization ability on the distribution $\mathcal{D}_t$. Notably, it is known that a sublinear dynamic regret is impossible in the worst case [17], so an upper bound of dynamic regret is desired to scale with a certain non-stationarity measure. A natural measure for OLaS would be the variation intensity of label distributions.

Remark 1 (Static regret vs. dynamic regret)**.** The classic measure for online learning is *static* regret, defined as $\mathbf{Reg}_T^{\mathbf{s}} = \sum_{t=1}^{T} R_t(\mathbf{w}_t) - \sum_{t=1}^{T} R_t(\mathbf{w}^*)$, where $\mathbf{w}^* \in \arg\min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} R_t(\mathbf{w})$ is the best *fixed* model in hindsight. The measure was adopted in the prior work of OLaS [16]. However, the measure is not suitable for OLaS, because it is too optimistic to expect a single fixed model to behave well over the whole process in changing environments. By contrast, minimizing dynamic regret facilitates the online algorithm with more adaptivity and robustness to non-stationary environments.

3 Proposed Approach

This section presents our approach for the OLaS problem, including the algorithms and theoretical guarantees. In the following, we respectively address the two central challenges of OLaS: the lack of supervision and the non-stationarity of online environments.

3.1 Unbiased Risk Estimator for Online Convex Optimization

OCO is a powerful and versatile framework for online learning problems, which enjoys both practical and theoretical appeals [21, 15]. Online Gradient Descent (OGD) [20] is one of the most fundamental

and powerful algorithms due to its light computational cost and sound regret guarantees. In the OLaS problem, recall that the learner's goal is to obtain a model sequence $\{\mathbf{w}_t\}_{t=1}^T$ enjoying low cumulative risk $\sum_{t=1}^T R_t(\mathbf{w}_t)$. Thus, suppose the model's risk $R_t(\mathbf{w}_t)$ is known at each round; then OGD simply updates the model by $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\left[\mathbf{w}_t - \eta \nabla R_t(\mathbf{w}_t)\right]$, where $\Pi_{\mathcal{W}}[\cdot]$ denotes the projection onto $\mathcal{W}$ and $\eta > 0$ is the step size. It is well-known that OGD guarantees the regret bound $\sum_{t=1}^T R_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^T R_t(\mathbf{w}) \leq \mathcal{O}(\sqrt{T})$ when risk function $R_t(\mathbf{w})$ is convex and step size is set as $\eta = \Theta(T^{-1/2})$ [15] (see Appendix B.1 for more details).

However, the expected risk function $R_t(\mathbf{w})$ is unknown in the current OLaS setup as it is defined over the underlying joint distribution $\mathcal{D}_t(\mathbf{x}, y)$. More severely, the online environments in the OLaS problem are fully *unlabeled*, which poses great challenges to apply the OCO framework. Indeed, the lack of supervision makes it hard to empirically assess the expected risk, not to mention ensuring the convexity. In the following, we establish an unbiased estimator $\widehat{R}_t(\mathbf{w})$ with unlabeled data S_t, which exhibits nice properties such that the OCO framework is still applicable for our purpose.

Unbiased Estimator under Label Shift. Inspired by the progress in offline label shift [9, 12, 13], we establish an unbiased risk estimator in OLaS for $R_t(\mathbf{w})$ with unlabeled data S_t and offline data S_0 by the risk rewriting technique. To this end, let $\boldsymbol{\mu}_{y_t} \in \Delta_K$ denote the label distribution vector with the k-th entry $[\boldsymbol{\mu}_{y_t}]_k \triangleq \mathcal{D}_t(y = k)$, then we have the following decomposition for the true risk:

$$R_t(\mathbf{w}) \triangleq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}[\ell(f(\mathbf{w}, \mathbf{x}), y)] = \sum_{k=1}^K [\boldsymbol{\mu}_{y_t}]_k \cdot R_t^k(\mathbf{w}) = \sum_{k=1}^K [\boldsymbol{\mu}_{y_t}]_k \cdot R_0^k(\mathbf{w}), \tag{2}$$

where $R_t^k(\mathbf{w}) \triangleq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t(\mathbf{x} \mid y=k)}[\ell(f(\mathbf{w}, \mathbf{x}), k)]$ is the risk of the model over the k-th label at round t. The second equality holds due to the law of total probability, and the third equality is by the label shift assumption that $\mathcal{D}_t(\mathbf{x} \mid y) = \mathcal{D}_0(\mathbf{x} \mid y)$ for any $t \in [T]$. Since the labeled offline data S_0 is always available, one can approximate $R_0^k(\mathbf{w})$ by its empirical version $\widehat{R}_0^k(\mathbf{w})$ with offline data S_0, where $\widehat{R}_0^k(\mathbf{w}) \triangleq \frac{1}{|S_0^k|}\sum_{\mathbf{x}_n\in S_0^k} \ell(f(\mathbf{w}, \mathbf{x}_n), k)$ and S_0^k is a subset of S_0 containing all samples with label k.

Therefore, the task is now to estimate the label distribution vector $\boldsymbol{\mu}_{y_t}$. To this end, we employ the Black Box Shift Estimation (BBSE) method [12] to construct an estimator via only offline data S_0 and unlabeled data S_t. Specifically, we first use the initial offline model f_0 to predict over the unlabeled data S_t and get predictive labels $\widehat{y}_t$, and next estimate the label distribution via solving the crucial equation $\boldsymbol{\mu}_{y_t} = C_{f_0}^{-1}\boldsymbol{\mu}_{\widehat{y}_t}$, where $\boldsymbol{\mu}_{\widehat{y}_t} \in \Delta_K$ is the distribution vector of the predictive labels $\widehat{y}_t$ and $C_{f_0} \in \mathbb{R}^{K\times K}$ is the confusion matrix with $[C_{f_0}]_{ij} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_0(\mathbf{x} \mid y=j)}[\mathbb{1}\{f_0(\mathbf{x}) = i\}]$ being the classification rate that the initial model f_0 predicts samples from class i as class j. We defer more details to Appendix B.2. As a result, through risk rewriting and prior estimation, we can construct the following estimator for the true risk $R_t(\mathbf{w})$:

$$\widehat{R}_t(\mathbf{w}) = \sum_{k=1}^K [\widehat{C}_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}]_k \cdot \widehat{R}_0^k(\mathbf{w}), \tag{3}$$

where $\widehat{C}_{f_0}$ and $\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}$ are empirical estimators of the confusion matrix and predictive label distribution vector using offline data S_0 and unlabeled data S_t only. Our constructed risk estimator enjoys the unbiasedness property, which plays a crucial role in the later algorithm design and theoretical analysis.

Lemma 1. *The estimator $\widehat{R}_t(\mathbf{w})$ in Eq. (3) is unbiased to $R_t(\mathbf{w}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}[\ell(f(\mathbf{w}, \mathbf{x}), y)]$, 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 the dataset S_t, provided C_{f_0} is invertible and the offline dataset S_0 has sufficient samples such that $\widehat{C}_{f_0} = C_{f_0}$ and $\widehat{R}_0^k(\mathbf{w}) = R_0^k(\mathbf{w})$, $\forall k \in \mathcal{Y}$.*

The proof of Lemma 1 can be found in Appendix D.1. Note that the sufficient sample assumption is introduced on the offline data S_0 to simplify the presentation. Indeed, we can further show $|\mathbb{E}_{S_t\sim\mathcal{D}_t}[\widehat{R}_t(\mathbf{w})] - R_t(\mathbf{w})| \leq \mathcal{O}(\sqrt{1/|S_0|})$ with high probability in Appendix D.2. Thus, the additional term is negligible when a large number of offline data is collected at the initial stage, which is easy to realize and also will not trivialize the online adaptation.

Remark 2 (Non-convexity issue)**.** Our risk estimator $\widehat{R}_t(\mathbf{w})$ can be *non-convex* as the estimated label distribution $[\widehat{C}_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}]_k$ might be negative in order to ensure the unbiasedness. Such a non-convex

behavior introduces a great challenge for applying the OCO framework. Fortunately, owing to its unbiasedness and the fact that the *expected* risk $R_t(\mathbf{w})$ is indeed convex, we can continue the following algorithm design and theoretical analysis building upon the constructed unbiased estimator.

OGD with Unbiased Estimator. Building upon the risk estimator in Eq. (3), we then deploy OGD and obtain our UOGD algorithm (abbreviated for "OGD with unbiased risk estimator"), namely,

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}[\mathbf{w}_t - \eta \nabla \widehat{R}_t(\mathbf{w}_t)]. \tag{4}$$

Despite the potential non-convexity of the risk estimator itself, we can still establish solid regret guarantees via the OCO framework due to the benign property that the risk estimator is unbiased and the expected risk is indeed convex. For example, UOGD provably enjoys an $\mathcal{O}(\sqrt{T})$ static regret. See the formal statement in Appendix D.3. We remark that our attained static regret already achieves the state-of-the-art theoretical understanding of OLaS, in the sense that previously the same bound can be only achieved with an additional unrealistic convexity assumption imposed over the algorithm [16], which UOGD does not require. Concretely, Wu et al. [16] assumed that *the risk estimator is convex* (in expectation), which is hard to theoretically verify since the estimator approximates the $0/1$-loss and involves an indicator function and an argmax operation due to the employed reweighting mechanism. By contrast, our estimator directly approximates the surrogate loss without reweighting and thus does not suffer from such limitations. Even if modifying their estimator to optimize a surrogate loss, the reweighting mechanism makes their method still hardly suitable for the OCO framework. More details are in Appendix C.3. In a nutshell, our constructed risk estimator enjoys nice properties, which are indispensable for the algorithm design and theoretical analysis.

3.2 Adapting to Non-stationarity of Online Label Shift

So far, an $\mathcal{O}(\sqrt{T})$ static regret has been established for OLaS; however, the guarantee is not appealing because static regret is not suitable for non-stationary online problems as discussed in Remark 1. We now introduce our method adapting to the non-stationarity with provable dynamic regret guarantees.

First, benefiting from the unbiasedness and expected convexity of our risk estimator, we prove that UOGD achieves a dynamic regret scaling with the label distribution drift V_T.

Theorem 1. *Under the same assumptions as Lemma 1, UOGD in Eq. (4) with step size η satisfies*

$$\mathbb{E}\big[\,\mathbf{Reg}_T^{\mathbf{d}}\,\big] \leq 2(KG^2/\sigma^2 + B^2)\eta T + \Gamma^2/\eta + 4(\Gamma+1)\sqrt{BV_T T/\eta} = \mathcal{O}\Big(\eta T + 1/\eta + \sqrt{(V_T T)/\eta}\Big), \tag{5}$$

where the constant $\sigma > 0$ denotes the minimum singular value of the invertible confusion matrix C_{f_0}. Moreover, $V_T = \sum_{t=2}^{T} \|\boldsymbol{\mu}_{y_t} - \boldsymbol{\mu}_{y_{t-1}}\|_1$ measures the intensity of the label distribution shift.

The proof of Theorem 1 can be found in Appendix E.1. The dynamic regret guarantee is obtained in a non-trivial way, and below we expand the technical innovations.

Remark 3 (Non-stationarity measure for OLaS)**.** For readers who are familiar with the literature, our result is reminiscent of the existing dynamic regret bound in the OCO studies [17, 22] on the surface; however, our result exhibits fundamental differences. The key caveat is that in our case the comparator $\mathbf{w}_t^*$ at each round is *not* the minimizer of the online function. Specifically, as the expected risk R_t is inaccessible, one has to work on the unbiased risk estimator $\widehat{R}_t$ and requires optimizing the *empirical* dynamic regret $\sum_{t=1}^{T} \widehat{R}_t(\mathbf{w}_t) - \sum_{t=1}^{T} \widehat{R}_t(\mathbf{w}_t^*)$. Importantly, $\mathbf{w}_t^* \in \arg\min_{\mathbf{w}\in\mathcal{W}} R_t(\mathbf{w})$ but $\mathbf{w}_t^* \notin \arg\min_{\mathbf{w}\in\mathcal{W}} \widehat{R}_t(\mathbf{w})$ in general. Although the empirical dynamic regret can be trivially bounded by $\sum_{t=1}^{T} \widehat{R}_t(\mathbf{w}_t) - \sum_{t=1}^{T} \min_{\mathbf{w}\in\mathcal{W}} \widehat{R}_t(\mathbf{w})$, the bound will then be loose and related to temporal variation of risk estimators [17, 22], making it hard to establish relationship to the natural non-stationarity measure of OLaS: V_T.[4] On the other hand, there exist studies benchmarking dynamic regret with other choices of comparators [20, 18], but the bounds scale with the consecutive variation of comparators $\sum_{t=2}^{T} \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_2$, whose relation to V_T is also unclear. To address the difficulty, drawing inspiration from [23], we decompose the expected dynamic regret into two parts: (i) the analysis of the first part relies on an in-depth analysis of the empirical dynamic regret of UOGD

[4] Besbes et al. [17] also considered a more general setting with noisy function value feedback. In such cases, the comparator is not the exact minimizer of the online function at each round, and their algorithm will require a *periodical restart* to deal with the non-stationarity. By contrast, ours does not require the restart in the algorithm.

Algorithm 1 ATLAS: base-algorithm

Input: step size $\eta_i \in \mathcal{H}$

let $\mathbf{w}_{1,i}$ be any point in $\mathcal{W}$
for $t = 2$ **to** T **do**
  construct the risk estimator $\widehat{R}_{t-1}$ as (3)
  update the model of base-learner by
  $\mathbf{w}_{t,i} = \Pi_{\mathcal{W}}[\mathbf{w}_{t-1,i} - \eta_i \nabla \widehat{R}_{t-1}(\mathbf{w}_{t-1,i})]$
  send $\mathbf{w}_{t,i}$ to the meta-algorithm
end for

Algorithm 2 ATLAS: meta-algorithm

Input: step size pool $\mathcal{H}$; learning rate ε

initialization: $\forall i \in [N], p_{1,i} = 1/N$
for $t = 2$ **to** T **do**
  receive $\{\mathbf{w}_{t,i}\}_{i=1}^N$ from base-learners
  update weight $\mathbf{p}_t \in \Delta_N$ according to
  $p_{t,i} \propto \exp(-\varepsilon \sum_{s=1}^{t-1} \widehat{R}_s(\mathbf{w}_{s,i}))$, $i \in [N]$
  predict final output $\mathbf{w}_t = \sum_{i=1}^N p_{t,i} \mathbf{w}_{t,i}$
end for

to track a specific sequence of piecewise-stationary comparators to avoid undesired variation of comparators; and (ii) the analysis of the second part is directly conducted on the *original* risk functions to attain a non-stationarity measure only related to the underlying label distribution shift.

From the upper bound in Eq. (5), we can observe that a proper step size tuning is crucial. Specifically, it can be verified that when the environment is near-stationary (more precisely, $V_T \leq \Theta(T^{-\frac{1}{2}})$), simply choosing $\eta = \Theta(1/\sqrt{T})$ ensures an $\mathcal{O}(\sqrt{T})$ dynamic regret, which is known to be minimax optimal even for the weaker measure of static regret [24]. Thus, in the following, we focus on the non-degenerated OLaS situation where $V_T \geq \Theta(T^{-\frac{1}{2}})$, and then the dynamic regret upper bound can be further simplified as $\mathcal{O}(\eta T + 1/\eta + \sqrt{(V_T T)/\eta})$. As a result, it is clear that UOGD can attain an $\mathcal{O}(V_T^{\frac{1}{3}} T^{\frac{2}{3}})$ dynamic regret by setting the step size optimally as $\eta = \Theta(T^{-\frac{1}{3}} V_T^{\frac{1}{3}})$, which is known to be *minimax optimal* due to the $\Omega(V_T^{\frac{1}{3}} T^{\frac{2}{3}})$ lower bound in [25, Theorem 6].

However, the optimal step size $\eta^* = \Theta(T^{-\frac{1}{3}} V_T^{\frac{1}{3}})$ crucially depends on the label distribution drift $V_T = \sum_{t=2}^T \|\boldsymbol{\mu}_{y_t} - \boldsymbol{\mu}_{y_{t-1}}\|_1$, which measures the non-stationarity and is unfortunately *unknown* to the learner. It is worth emphasizing that the problem cannot be addressed by standard adaptive step size tuning mechanisms in online learning literature, such as the doubling trick [26] or self-confident tuning [27], in that the variation quantity V_T cannot be empirically evaluated as it is defined over the underlying label distribution $\mathcal{D}_t(y)$ inaccessible to the learner. Even diving into the analysis of Theorem 1, the adaptive tuning requires the knowledge of *original* risk functions $\{R_t\}_{t=1}^T$, which are also unavailable. Intuitively, the hardness of such optimal step size tuning essentially comes from the uncertainty of the non-stationary label shifts.

To overcome the difficulty, inspired by recent advances in non-stationary online learning [18, 19], we propose an *online ensemble* algorithm for the OLaS problem called ATLAS (Adapting To LAbel Shift). Specifically, to cope with the uncertainty in the optimal step size tuning, we first design a pool of candidate step sizes denoted by $\mathcal{H} = \{\eta_1, \ldots, \eta_N\}$ such that η_* can be well approximated by at least one of those candidates, which will be configured later and here N is the number of candidate step sizes. Then, ATLAS deploys a two-layer structure by maintaining a group of base-learners, each associated with a candidate step size from the pool $\mathcal{H}$ and then employs a meta-algorithm to track the best base-learner. The main procedures are presented in Algorithm 1 (base-algorithm) and Algorithm 2 (meta-algorithm), and we describe the details below.

Base-algorithm. We parallelly run a group of instances of UOGD, each one is associated with a candidate step size in the step size pool $\mathcal{H}$. Formally, the i-th base-learner yields a sequence of base models $\{\mathbf{w}_{t,i}\}_{t=1}^T$, which is updated by $\mathbf{w}_{t,i} = \Pi_{\mathcal{W}}[\mathbf{w}_{t-1,i} - \eta_i \nabla \widehat{R}_{t-1}(\mathbf{w}_{t-1,i})]$ with $\eta_i \in \mathcal{H}$.

Meta-algorithm. The meta-algorithm aims to combine all the base-learners' decisions such that the final output is competitive with the decisions returned by the (unknown) base-learner associated with the best step size. To achieve this, we employ a weighted combination mechanism with the final model as $\mathbf{w}_t = \sum_{i=1}^N p_{t,i} \mathbf{w}_{t,i}$, where $\mathbf{p}_t \in \Delta_N$ is the weight vector with $p_{t,i}$ denoting the weight of the i-th base-learner. We use the classic Hedge algorithm [28] to update the weight vector, namely, $p_{t,i} \propto \exp\big(-\varepsilon \sum_{s=1}^{t-1} \widehat{R}_s(\mathbf{w}_{s,i})\big)$, where $\varepsilon > 0$ is the learning rate of the meta-algorithm that can be simply set as $\Theta(\sqrt{(\ln N)/T})$ without dependence on label shift quantity. Intuitively, the meta-algorithm puts larger weights on base-learners with a smaller cumulative estimated risk so that the overall models $\{\mathbf{w}_t\}_{t=1}^T$ can be competitive to the base-learner with the best performance.

ATLAS enjoys the following dynamic regret guarantee with the proof in Appendix E.2.

Theorem 2. *Set the step size pool as* $\mathcal{H} = \{\eta_i = \frac{\Gamma\sigma}{2G\sqrt{KT}} \cdot 2^{i-1} \mid i \in [N]\}$, *where* $N = 1 + \lceil \frac{1}{2}\log_2(1+2T) \rceil$ *is the number of base-learners.* ATLAS *ensures that* $\mathbb{E}[\mathbf{Reg}_T^{\mathbf{d}}] \leq \mathcal{O}(\max\{V_T^{\frac{1}{3}} T^{\frac{2}{3}}, \sqrt{T}\})$, *or simplified as* $\mathcal{O}(V_T^{\frac{1}{3}} T^{\frac{2}{3}})$ *for non-degenerated cases of* $V_T \geq \Theta(T^{-\frac{1}{2}})$.

Theorem 2 shows that ATLAS enjoys the same dynamic regret as UOGD with the (unknown) optimal step size, but unknown non-stationarity V_T is no more required in advance. Algorithmically, our online method maintains $N = \Theta(\log T)$ base-learners to achieve an optimal dynamic regret, which would be computationally acceptable given a logarithmic dependence on T.

3.3 More Adaptive Algorithm by Exploiting Label Shift Structures

ATLAS is equipped with a minimax optimal dynamic regret, which indicates that it can safeguard the optimal theoretical property in the worst case. Worst-case optimality serves as the "stress-testing" for the robustness to non-stationarity environments [29]. At the same time, more adaptive results beyond the worst-case analysis are also urgently desired, as improved performance is naturally expected in many easier situations when the shift admits specific patterns such as periodicity or gradual evolution.

To this end, we propose an improved algorithm called ATLAS-ADA with provably more adaptive guarantees. The key idea is to exploit the label shift patterns and reuse historical information to help the current online update [30]. We build on the framework of *optimistic online learning* [31, 32] by introducing a *hint* function $H_t : \mathcal{W} \mapsto \mathbb{R}$ to encode shift patterns from historical data, which serves as an estimation of the expected risk $R_t(\mathbf{w}_t)$. Below, we start with a given hint function and describe the usage, and finally elaborate on how to design $H_t(\cdot)$ guided by the theory.

Similar to ATLAS, the improved ATLAS-ADA also deploys a two-layer meta-base structure. The key difference lies in the usage of the hint function at both the base-level and meta-level.

Base-algorithm. Besides the gradient descent step as did in ATLAS, another update step related to the hint function $H_t(\cdot)$ is performed. Concretely, the i-th base-learner updates the parameters by

$$\widehat{\mathbf{w}}_{t,i} = \Pi_{\mathcal{W}}[\widehat{\mathbf{w}}_{t-1,i} - \eta_i \nabla \widehat{R}_{t-1}(\mathbf{w}_{t-1,i})], \quad \mathbf{w}_{t,i} = \arg\min_{\mathbf{w}\in\mathcal{W}} \eta_i H_t(\mathbf{w}) + 1/2 \cdot \|\mathbf{w} - \widehat{\mathbf{w}}_{t,i}\|_2^2, \quad (6)$$

where $\widehat{\mathbf{w}}_{t,i}$ is an intermediate output and $\mathbf{w}_{t,i}$ is the final returned model. When $H_t(\mathbf{w}) = 0$ (i.e., without a hint function), the above two-step update simply degenerates to the same UOGD update in the base-learner of ATLAS by noting that now $\mathbf{w}_{t,i} = \widehat{\mathbf{w}}_{t,i}$. In the general $H_t(\cdot)$ case, the second step in (6) is crucial and can be regarded as another descent towards the direction specified by the hint function. As a result, this will reduce the regret whenever the hint function is set appropriately to approximate well the next-round risk function, which will be clear in the regret bound presented later.

Meta-algorithm. The meta-algorithm is used to track the best base-learner, and the hint function $H_t(\cdot)$ is also necessary to be considered in the update to achieve the adaptivity. To this end, we inject the hint function as the loss evaluation of the meta-algorithm, and then the weight is updated by $p_{t,i} \propto \exp\left(-\varepsilon\left(\sum_{s=1}^{t-1} \widehat{R}_s(\mathbf{w}_{s,i}) + H_t(\mathbf{w}_{t,i})\right)\right)$ for all $i \in [N]$, where ε is the learning rate that can be set properly without dependence on V_T. The key distinction to the meta-algorithm of ATLAS is the additional loss $H_t(\mathbf{w}_{t,i})$ evaluated over the current local models $\{\mathbf{w}_{t,i}\}_{i=1}^N$ by the hint function.

The main procedures are presented in Algorithm 4 (base-algorithm) and Algorithm 5 (meta-algorithm). We have the following dynamic regret bound for ATLAS-ADA (proof in Appendix F.2).

Theorem 3. *Suppose the hint function* $H_t : \mathcal{W} \mapsto \mathbb{R}$ *is convex, satisfies* $\max_{\mathbf{w}\in\mathcal{W}} \|\nabla H_t(\mathbf{w})\|_2 \leq \max_{\mathbf{w}\in\mathcal{W}} \|\nabla \widehat{R}_t(\mathbf{w})\|_2$, *and is independent of current data* S_t. *Set the step size pool as* $\mathcal{H} = \{\eta_i = \frac{\Gamma\sigma}{\sqrt{\sigma^2 + 4G^2KT}} \cdot 2^{i-1} \mid i \in [N]\}$ *with* $N = 2 + \lceil \frac{1}{2}\log_2(3T(1 + 4G^2KT/\sigma)) \rceil$. ATLAS-ADA *ensures* $\mathbb{E}[\mathbf{Reg}_T^{\mathbf{d}}] \leq \mathcal{O}(V_T^{1/3} G_T^{1/3} T^{1/3})$, *where* $G_T = \sum_{t=1}^T \mathbb{E}[\sup_{\mathbf{w}\in\mathcal{W}} \|\nabla \widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2^2]$ *measures the reusability of historical information, depending on label shift patterns and hint function designs.*

In the worst case, G_T is at most $\mathcal{O}(T)$ given a bounded gradient norm $\|\nabla H_t(\mathbf{w})\|_2$, and thus the bound presented in Theorem 3 safeguards the same $\mathcal{O}(V_T^{1/3} T^{2/3})$ minimax bound as ATLAS. More importantly, when the hint function $H_t(\cdot)$ encodes beneficial information and is close to the risk function, the obtained bound can be substantially better than the minimax rate.

Remark 4 (Implicit update)**.** Problem-dependent dynamic regret was first presented in [19] for convex and smooth functions. However, their result critically relied on the *smoothness* condition, which is not satisfied in our OLaS case. Our key technical innovation is the *implicit update* in the second step of (6). The previous method required the gradient-descent type update $\mathbf{w}_{t,i} = \Pi_{\mathbf{w}\in\mathcal{W}}[\widehat{\mathbf{w}}_{t,i} - \eta_i \nabla H_t(\mathbf{w}_{t-1,i})]$, which can be deemed as an approximated optimization over the linearized loss $\langle \nabla H_t(\mathbf{w}_{t-1,i}), \mathbf{w}\rangle$. By contrast, we directly updates over the *original function* $H_t(\mathbf{w})$, hence called the "implicit" update [33, 34]. Albeit with slightly larger computational complexity (which will not be a barrier given a proper design of hint functions), our method enjoys the same dynamic regret *without* smoothness, which could be of independent interest for general OCO purposes.

Design of Hint Functions. The hint functions should minimize the reusability measure G_T to sharpen the dynamic regret as suggested by Theorem 3. Recall that $\widehat{R}_t(\mathbf{w}) = \sum_{k=1}^K [\widehat{\boldsymbol{\mu}}_{y_t}]_k \cdot \widehat{R}_0^k(\mathbf{w})$. Thus, a natural construction is $H_t(\mathbf{w}) = \sum_{k=1}^K [\boldsymbol{h}_{y_t}]_k \cdot \widehat{R}_0^k(\mathbf{w})$ parametrized by *hint priors* $\boldsymbol{h}_{y_t} \in \mathbb{R}^K$, which is used to estimate the class prior based on the past observed data $\{S_\tau\}_{\tau=1}^{t-1}$. Then, G_T satisfies the *bias-variance decomposition* (with bias term $\mathbb{E}[\|\boldsymbol{h}_{y_t} - \boldsymbol{\mu}_{y_t}\|_2^2]$ and variance term $\mathbb{E}[\|\boldsymbol{\mu}_{y_t} - \widehat{\boldsymbol{\mu}}_{y_t}\|_2^2]$):

$$G_T \leq KG^2 \sum_{t=1}^T \mathbb{E}\left[\|\boldsymbol{h}_{y_t} - \widehat{\boldsymbol{\mu}}_{y_t}\|_2^2\right] \leq 2KG^2 \sum_{t=1}^T \left(\mathbb{E}\left[\|\boldsymbol{h}_{y_t} - \boldsymbol{\mu}_{y_t}\|_2^2\right] + \mathbb{E}\left[\|\boldsymbol{\mu}_{y_t} - \widehat{\boldsymbol{\mu}}_{y_t}\|_2^2\right]\right).$$

Setting $\boldsymbol{h}_{y_t} = \boldsymbol{\mu}_{y_t}$ will make the upper bound tightest possible, though the underlying class prior $\boldsymbol{\mu}_{y_t}$ is not accessible in practice. So the design of the hint function is actually a task of approximating it with different parts of previous data guided by prior patterns. In the experiments, we design four hint functions by encoding different knowledge, including Forward Hint (*Fwd*), Window Hint (*Win*), Periodic Hint (*Peri*), and Online KMeans Hint (*OKM*). More details are in Appendix F.1.

4 Experiments

In this section, we conduct extensive experiments to validate the effectiveness of the proposed methods (ATLAS and ATLAS-ADA) and justify the theoretical findings. We begin this section with a brief introduction to the experimental setups (more details are deferred to Appendix A) and then present empirical results on the synthetic and real-world data, respectively.

Experiments Setup. We compare seven algorithms in various experimental configurations. The contenders include a baseline that predicts with the initial model directly (*FIX*), three OLaS algorithms proposed by the previous work [16] (*ROGD*, *FTH*, and *FTFWH*), and our proposals (*UOGD*, ATLAS, and ATLAS-ADA) with the logistic regression model. Besides, we simulate four types of label shift on synthetic and benchmark data to capture different distribution change patterns. Two of them, including `Sine Shift` ( ) and `Square Shift` ( ) change in a periodic pattern. The other two have no periodic structure but are introduced to capture different shift intensities. The underlying distribution changes slowly in the `Linear Shift` ( ) while changes fast in `Bernoulli Shift` ( ). We repeat all experiments for five times and evaluate the contenders by the average error $\frac{1}{T}\sum_{t=1}^T \frac{1}{|S_t|}\sum_{n=1}^{|S_t|} \mathbb{1}\left[f\left(\mathbf{w}, \mathbf{x}_n\right) \neq y_n\right]$ over $T = 10,000$ rounds.

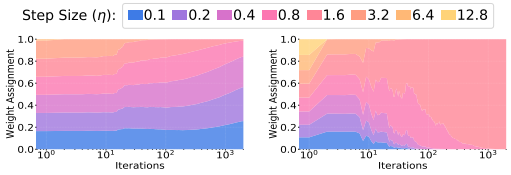
4.1 Illustrations on Synthetic Data

This subsection first compares all contenders on the synthetic data. Then, we further illustrate the effectiveness of our proposal by a closer look at the two key components, including a meta-base structure for step size search and a hint function for historical information reuse.

Overall Performance Comparison. Table 1 compares ATLAS with other methods when $N_t = 100$ samples are received at every iteration. Basically, *FIX* is inferior to the online methods, which shows the necessity of designing online algorithms for the OLaS problem. Besides, *UOGD* outperforms *ROGD*, the OGD algorithm running with the risk estimator proposed by [16]. The comparison demonstrates the empirical superiority of our estimator besides its benign theoretical properties. Moreover, the ATLAS surpasses almost all other methods in the four shift patterns. In particular, it achieves a significant advantage over *UOGD* (with step size $\eta = \Theta(T^{-1/2})$) when the environments change relatively fast (`Squ`, `Sin`, and `Ber`). The results justify our theoretical finding that the small step size $\Theta(T^{-1/2})$ suggested by the static regret analysis is unsuitable for the dynamic environments. Our method can better adapt to the changing environments by enjoying the dynamic regret guarantees.

Table 1: Average error (%) for different algorithms under various simulated shifts for the synthetic data. The best algorithms are emphasized in bold (paired t-test at a 5% confidence level).

	Lin	Squ	Sin	Ber
FIX	7.87±0.03	7.87±0.02	7.34±0.03	7.79±0.02
FTH	**4.70±0.02**	6.50±0.01	6.36±0.03	6.60±0.01
FTFWH	5.27±0.02	6.52±0.01	6.36±0.02	6.60±0.01
ROGD	6.08±0.01	7.11±0.01	6.87±0.02	6.40±0.01
UOGD	5.35±0.02	6.17±0.01	6.37±0.01	5.46±0.05
ATLAS	5.44±0.02	**4.27±0.02**	**5.75±0.01**	**4.04±0.07**



(a) Linear shift (b) Bernoulli shift

Figure 1: Weight assigned of the ATLAS algorithm for each step size along the learning process. Different colors are used to indicate different step sizes.

Table 2: Average error (%) for ATLAS-ADA with four hint functions under different sample sizes. The best one is emphasized in bold. Besides, • indicates a better result than ATLAS without hint (*None*).

Shift Type	Sample Size: 1					Sample Size: 10					Sample Size: 100				
	None	*Win*	*Peri*	*Fwd*	*OKM*	*None*	*Win*	*Peri*	*Fwd*	*OKM*	*None*	*Win*	*Peri*	*Fwd*	*OKM*
Lin	6.28 ± 0.21	• 5.89 ± 0.26	• 5.99 ± 0.29	• 6.01 ± 0.31	• **5.35** ± 0.31	5.61 ± 0.04	• 5.47 ± 0.04	• 5.43 ± 0.03	• 5.53 ± 0.05	• **5.42** ± 0.05	5.44 ± 0.02	5.44 ± 0.03	• **5.38** ± 0.02	• 5.40 ± 0.02	5.45 ± 0.03
Squ	6.03 ± 0.23	• 5.83 ± 0.24	• 5.27 ± 0.20	5.88 ± 0.23	• **5.07** ± 0.35	4.59 ± 0.02	4.69 ± 0.02	• 3.85 ± 0.04	• **3.72** ± 0.02	• 3.91 ± 0.03	4.27 ± 0.02	4.68 ± 0.02	• 3.39 ± 0.03	• **3.33** ± 0.03	• 3.46 ± 0.04
Sin	6.90 ± 0.22	• 6.58 ± 0.22	• 6.59 ± 0.25	• 6.43 ± 0.26	• **5.25** ± 0.22	6.12 ± 0.07	• 5.99 ± 0.06	• 5.83 ± 0.05	• **5.78** ± 0.05	• 5.86 ± 0.04	5.75 ± 0.01	5.78 ± 0.01	• 5.53 ± 0.00	• **5.48** ± 0.01	• 5.58 ± 0.00
Ber	5.55 ± 0.09	• 5.42 ± 0.11	• 5.43 ± 0.09	5.63 ± 0.16	• **4.69** ± 0.17	4.39 ± 0.10	4.45 ± 0.08	4.43 ± 0.10	• **3.66** ± 0.10	• 3.73 ± 0.06	4.04 ± 0.07	4.29 ± 0.06	4.26 ± 0.06	• **3.19** ± 0.07	• 3.45 ± 0.11

Effectiveness of Meta-Base Structure. One key component of our method is the meta-base structure to address the non-stationarity. To better illustrate its effectiveness, we visualize the weights $p_{t,i}$ assigned for each base-learner of ATLAS. As shown in Figure 5, the meta-algorithm can quickly assign larger weights to appropriate base-learners along the learning process. Specifically, Figure 1(a) illustrates the case of slowly changing environments, where more weights are assigned to the base-learners with small step sizes. In the fast-changing case, see Figure 1(b), larger step sizes are preferred. The results show that our algorithm can adaptively track the suitable step sizes according to the shift intensity of environments. Additional results for other shifts can be found in Figure 5.

Effectiveness of Using Hint Functions. Table 2 reports the performance of ATLAS-ADA with four different hint functions under sample sizes are $N_t = 1, 10$, and 100. All hint functions improve over vanilla ATLAS in most cases. When N_t is reasonably large, the Fwd hint, performing transductive learning with current unlabeled data, achieves the best performance. While, the OKM hint, which learns previous patterns by online K-means, is the best choice for a small sample size case ($N_t = 1$).

To illustrate how the hint function works, we further vary the buffer size of the Periodic Hint (*Peri*) on the Squ environment (), which shifts in a periodic length $L = 40$. As shown in Figure 2, when the buffer size matches the multiples of length L, *Peri* can significantly improve the vanilla ATLAS. Besides the improvement, our method is shown to achieve comparable performance with ATLAS even if the buffer size is misspecified. The result validates our theoretical guarantee of the safety of using hint functions (the readers can refer to the paragraph above Remark 4).

4.2 Comparisons on Real-world Data

We conduct experiments on real-world data, including six benchmark datasets (ArXiv, EuroSAT, MNIST, Fashion, CIFAR10, and CINIC10) and the SHL dataset [35] for the real-life locomotion detection task. Table 3 reports the averaged error of different algorithms, which shows a similar tendency as the results in the synthetic experiments. When the distribution changes rapidly (Ber), ATLAS and ATLAS-ADA outperform other contenders. Similar results are also observed in the Sine and Square shift, see Appendix A.1 for details. Even in a relatively stationary environment (Lin), our algorithms are comparable with the best algorithm (UOGD), which is specifically designed for stationary cases. The above results validate the adaptivity of the proposed algorithms.

Further, we highlight the results on the locomotion detection task. The task aims to distinguish six types of locomotion with sensor data from mobile phones. Figure 3(a) reports the averaged error of all contenders, which shows the superiority of our proposals ATLAS and ATLAS-ADA (with the *OKM* hint) over the entire time horizons. In addition, to further validate the adaptivity of our algorithm to the underlying environment regardless of the fast or slow changes, we simulate various shift

Table 3: Average error (%) of different algorithms on various real-world datasets (`Lin` and `Ber`). We report the mean and standard deviation over five runs. The best algorithms are emphasized in bold. "•" indicates the algorithms that are significantly inferior to ATLAS-ADA by the paired t-test at a 5% significance level. Here AT-ADA represents ATLAS-ADA (with *OKM*). The online sample size is set as $N_t = 10$.

	Lin							Ber						
	FIX	**FTH**	**FTFWH**	**ROGD**	**UOGD**	**ATLAS**	**AT-ADA**	**FIX**	**FTH**	**FTFWH**	**ROGD**	**UOGD**	**ATLAS**	**AT-ADA**
ArXiv	• 30.28 ±0.07	• 28.18 ±0.28	• 25.74 ±0.21	• 23.09 ±0.20	**21.04** ±0.11	• 22.10 ±0.09	21.28 ±0.09	• 30.63 ±0.20	• 27.69 ±0.13	• 28.50 ±0.19	• 24.82 ±0.11	• 21.53 ±0.68	• 21.11 ±0.70	**20.58** ±0.69
EuroSAT	• 14.06 ±0.09	• 11.16 ±0.11	• 9.78 ±0.12	• 12.56 ±3.16	**7.04** ±0.11	• 7.19 ±0.10	7.13 ±0.11	• 14.12 ±0.13	• 10.48 ±0.09	• 10.50 ±0.08	• 9.06 ±0.05	• 7.28 ±0.04	• 6.99 ±0.03	**6.91** ±0.05
MNIST	• 1.79 ±0.02	• 1.38 ±0.03	• 1.20 ±0.02	• 1.25 ±0.02	**1.06** ±0.02	**1.06** ±0.02	**1.06** ±0.02	• 1.81 ±0.05	• 1.29 ±0.03	• 1.34 ±0.03	• 1.33 ±0.03	• 1.12 ±0.02	**1.03** ±0.02	**1.03** ±0.02
Fashion	• 11.86 ±0.04	• 8.47 ±0.07	**7.84** ±0.06	8.18 ±0.07	7.95 ±0.08	• 8.36 ±0.07	8.04 ±0.08	• 11.85 ±0.09	• 8.48 ±0.11	• 8.69 ±0.10	• 8.72 ±0.08	• 8.23 ±0.12	• 7.91 ±0.12	**7.69** ±0.12
CIFAR10	• 20.77 ±0.12	• 17.36 ±0.14	15.77 ±0.12	• 18.45 ±0.47	**15.54** ±0.15	• 15.77 ±0.11	15.62 ±0.14	• 20.82 ±0.12	• 17.06 ±0.14	• 16.96 ±0.15	• 17.66 ±0.13	• 15.93 ±0.29	• 14.98 ±0.30	**14.80** ±0.29
CINIC10	• 33.98 ±0.22	• 28.85 ±0.10	• 26.87 ±0.13	• 32.54 ±2.59	**26.21** ±0.15	• 26.66 ±0.19	26.38 ±0.16	• 34.11 ±0.35	• 28.48 ±0.17	• 28.44 ±0.19	• 28.90 ±0.19	• 26.63 ±0.55	• 25.85 ±0.58	**25.63** ±0.60

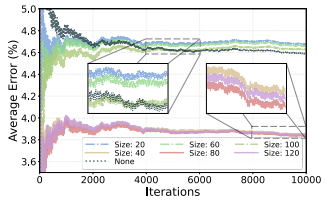
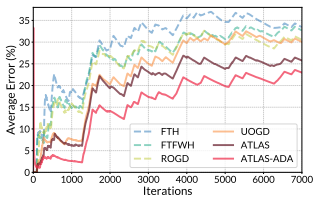
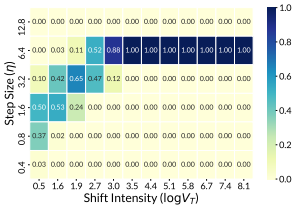


Figure 2: Average error of *Peri* hint with different buffer sizes. ATLAS-ADA enjoys a significant improvement with proper buffer sizes while still safeguarding similar performance as ATLAS even the size is misspecified.



(a) overall performance



(b) weight heatmap

Figure 3: (a) Overall performance comparison on the locomotion detection task. (b) The heatmap of the final round weights for each step size. A larger V_T implies a larger intensity of distribution shift. The darker color in the heatmap indicates the larger weight.

intensities by sampling the original data with different frequencies. Figure 3(b) shows the weight assignment of ATLAS-ADA for each step size in the final round. Our method automatically selects a larger step size for larger V_T while tracking a small step size in a relatively static scenario.

5 Conclusion

This paper proposed algorithms for online label shift with provable guarantees. We constructed an unbiased risk estimator without using any supervision at test time. Then, we designed novel online ensemble algorithms that automatically adapt to the non-stationary online label shift and enjoy problem-dependent dynamic regret. Our proposed ATLAS algorithm employed a meta-base structure to handle the non-stationarity and obtained an $\mathcal{O}(V_T^{1/3}T^{2/3})$ guarantee, and ATLAS-ADA further introduced hint functions to exploit the shift structure and obtained an improved $\mathcal{O}(V_T^{1/3}G_T^{1/3}T^{1/3})$ guarantee. Extensive experiments validated the effectiveness of the proposed algorithms.

Our study serves as a preliminary attempt to bridge the distribution change problem and online learning techniques by focusing on the label shift case. Considering a more general distribution change setting is an important future direction. Besides, our current algorithm is designed for the most challenging unlabeled scenario, and it is interesting to consider relaxed real-world demand where a few labels could be available in the learning process. Moreover, our obtained regret guarantees hold in expectation, and we will take high-probability bounds as the future work.

Acknowledgments

Yong Bai, Peng Zhao, and Zhi-Hua Zhou were supported by NSFC (61921006), JiangsuSF (BK20220776), and Collaborative Innovation Center of Novel Software Technology and Industrialization. Yu-Jie Zhang was supported by Todai Fellowship and the Institute for AI and Beyond, UTokyo. Masashi Sugiyama was supported by JST AIP Acceleration Research Grant Number JPMJCR20U3 and the Institute for AI and Beyond, UTokyo. Peng Zhao thanks Chen-Yu Wei for discussions on the conversation between function variation and gradient variation. The authors also thank Ruihan Wu and Yu-Yang Qian for their helpful discussions.

References

[1] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[2] Masashi Sugiyama and Motoaki Kawanabe. *Machine Learning in Non-stationary Environments: Introduction to Covariate Shift Adaptation*. The MIT Press, 2012.

[3] Thomas G. Dietterich. Steps toward robust artificial intelligence. *AI Magazine*, 38(3):3–24, 2017.

[4] Yoshua Bengio, Yann Lecun, and Geoffrey Hinton. Deep learning for AI. *Communication of ACM*, 64(7):58–65, 2021.

[5] Zhi-Hua Zhou. Open-environment machine learning. *National Science Review*, 9(8):nwac123, 2022.

[6] Alexander Gómez-Villa, Augusto Salazar, and Jesús Francisco Vargas-Bonilla. Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks. *Ecological Informatics*, 41:24–32, 2017.

[7] Mohammad Sadegh Norouzzadeh, Anh Nguyen, Margaret Kosmala, Alexandra Swanson, Meredith S. Palmer, Craig Packer, and Jeff Clune. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25):E5716–E5725, 2018.

[8] Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002.

[9] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 819–827, 2013.

[10] Marthinus Christoffel du Plessis and Masashi Sugiyama. Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks*, 50:110–119, 2014.

[11] Tuan Duong Nguyen, Marthinus Christoffel du Plessis, and Masashi Sugiyama. Continuous target shift adaptation in supervised learning. In *Proceedings of the 7th Asian Conference on Machine Learning (ACML)*, pages 285–300, 2015.

[12] Zachary C. Lipton, Yu-Xiang Wang, and Alexander J. Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 3128–3136, 2018.

[13] Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[14] Saurabh Garg, Yifan Wu, Sivaraman Balakrishnan, and Zachary C. Lipton. A unified view of label shift estimation. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 3290–3300, 2020.

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

[16] Ruihan Wu, Chuan Guo, Yi Su, and Kilian Q. Weinberger. Online adaptation to label distribution shift. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 11340–11351, 2021.

[17] Omar Besbes, Yonatan Gur, and Assaf J. Zeevi. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, 2015.

[18] Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 1330–1340, 2018.

[19] Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Dynamic regret of convex and smooth functions. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 12510–12520, 2020.

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

[21] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[22] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization : Competing with dynamic comparators. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 398–406, 2015.

[23] Yu-Jie Zhang, Peng Zhao, and Zhi-Hua Zhou. A simple online algorithm for competing with dynamic comparators. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 390–399, 2020.

[24] Jacob Abernethy, Peter L Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal strategies and minimax lower bounds for online convex games. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pages 415–423, 2008.

[25] Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. 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)*, pages 449–457, 2016.

[26] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

[27] Peter Auer, Nicolò Cesa-Bianchi, and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, 2002.

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

[29] Tim Roughgarden, editor. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2020.

[30] Peng Zhao, Le-Wen Cai, and Zhi-Hua Zhou. Handling concept drift via model reuse. *Machine Learning*, 109(3):533–568, 2020.

[31] Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Proceedings of the 25th Conference On Learning Theory (COLT)*, pages 6.1–6.20, 2012.

[32] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Proceedings of the 26th Conference On Learning Theory (COLT)*, pages 993–1019, 2013.

[33] Brian Kulis and Peter L. Bartlett. Implicit online learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 575–582, 2010.

[34] Nicolò Campolongo and Francesco Orabona. Temporal variability in implicit online learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 12377–12387, 2020.

[35] Hristijan Gjoreski, Mathias Ciliberto, Lin Wang, Francisco Javier Ordóñez Morales, Sami Mekki, Stefan Valentin, and Daniel Roggen. The university of Sussex-Huawei locomotion and transportation dataset for multimodal analytics with mobile devices. *IEEE Access*, 6: 42592–42604, 2018.

[36] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108, 2019.

[37] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 204–207, 2018.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[39] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[40] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747, 2017.

[41] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[42] Luke Nicholas Darlow, Elliot J. Crowley, Antreas Antoniou, and Amos J. Storkey. CINIC-10 is not ImageNet or CIFAR-10. arXiv:1810.03505, 2018.

[43] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[44] Shai Shalev-Shwartz. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[45] Jacob D. Abernethy, Peter L. Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal stragies and minimax lower bounds for online convex games. In *Proceedings of the 21st Conference On Learning Theory (COLT)*, pages 415–424, 2008.

[46] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[47] Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pages 57–68, 2008.

[48] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11:2973–3009, 2010.

[49] Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Class-prior estimation for learning from positive and unlabeled data. *Machine Learning*, 106(4):463–492, 2017.

[50] Harish G. Ramaswamy, Clayton Scott, and Ambuj Tewari. Mixture proportion estimation via kernel embeddings of distributions. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 2052–2060, 2016.

[51] Clayton Scott. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 838–846, 2015.

[52] Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *Advances in Neural Information Processing Systems 27 (NeurIPS)*, pages 703–711, 2014.

[53] Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1386–1394, 2015.

[54] Yu-Jie Zhang, Peng Zhao, Lanjihong Ma, and Zhi-Hua Zhou. An unbiased risk estimator for learning with augmented classes. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 10247–10258, 2020.

[55] Lijun Zhang, Tianbao Yang, Jinfeng Yi, Rong Jin, and Zhi-Hua Zhou. Improved dynamic regret for non-degenerate functions. In *Advance in Neural Information Processing Systems 30 (NIPS)*, pages 732–741, 2017.

[56] Lijun Zhang, Tianbao Yang, Rong Jin, and Zhi-Hua Zhou. Dynamic regret of strongly adaptive methods. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 5877–5886, 2018.

[57] Dheeraj Baby and Yu-Xiang Wang. Online forecasting of total-variation-bounded sequences. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 11069–11079, 2019.

[58] Xi Chen, Yining Wang, and Yu-Xiang Wang. Non-stationary stochastic optimization under $L_{p,q}$-variation measures. *Operations Research*, 67(6):1752–1765, 2019.

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

[60] Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, pages 7195–7201, 2016.

[61] Lijun Zhang, Shiyin Lu, and Tianbao Yang. Minimizing dynamic regret and adaptive regret simultaneously. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 309–319, 2020.

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

[63] Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Adaptivity and non-stationarity: Problem-dependent dynamic regret for online convex optimization. arXiv:2112.14368, 2021.

[64] Peng Zhao, Guanghui Wang, Lijun Zhang, and Zhi-Hua Zhou. Bandit convex optimization in non-stationary environments. *Journal of Machine Learning Research*, 22(125):1 – 45, 2021.

[65] Dheeraj Baby and Yu-Xiang Wang. Optimal dynamic regret in exp-concave online learning. In *Proceedings of the 34th Annual Conference on Learning Theory (COLT)*, pages 359–409, 2021.

[66] Peng Zhao, Yu-Xiang Wang, and Zhi-Hua Zhou. Non-stationary online learning with memory and non-stochastic control. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2101–2133, 2022.

[67] Lijun Zhang, Wei Jiang, Shiyin Lu, and Tianbao Yang. Revisiting smoothed online learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 13599–13612, 2021.

[68] Peng Zhao, Long-Fei Li, and Zhi-Hua Zhou. Dynamic regret of online markov decision processes. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 26865–26894, 2022.

[69] Mengxiao Zhang, Peng Zhao, Haipeng Luo, and Zhi-Hua Zhou. No-regret learning in time-varying zero-sum games. In *Proceedings of the 39th International Conference on Machine Learning (ICML))*, pages 26772–26808, 2022.

[70] Andrew Jacobsen and Ashok Cutkosky. Parameter-free mirror descent. In *Proceedings of the 35th Annual Conference on Learning Theory (COLT)*, pages 4160–4211, 2022.

[71] Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

[72] Daniel J. Hsu, Sham M. Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

[73] Wesam Barbakh and Colin Fyfe. Online clustering algorithms. *International Journal of Neural Systems*, 18(3):185–194, 2008.

[74] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, second edition, 2012.

[75] Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E. Schapire. Fast convergence of regularized learning in games. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 2989–2997, 2015.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Remark 4 for a discussion on computational efficiency and Section 5 for a discussion on high probability bound.

(c) Did you discuss any potential negative societal impacts of your work? [No] We cannot foresee a potential negative societal impact of this work.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [Yes]

   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix D- F.

3. If you ran experiments...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the supplemental material.

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A.2.

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes]

   (b) Did you mention the license of the assets? [Yes]

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Our code is in the supplemental material.

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] All the data are open source.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] None of the personally identifiable information or offensive content is involved.

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [No] We do not use crowdsourcing.

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [No] We do not use crowdsourcing.

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [No] We do not use crowdsourcing.

Supplementary Material for "Adapting to Online Label Shift with Provable Guarantees"

This is the supplemental material for the paper "Adapting to Online Label Shift with Provable Guarantees". The appendix is organized as follows.

- Appendix A: details of the experiments.
  - Appendix A.1: more numerical results on real-world datasets.
  - Appendix A.2: details of experimental setup.
  - Appendix A.3: additional experimental results.
- Appendix B: a brief introduction to OCO and BBSE for label shift.
  - Appendix B.1: review of the OCO framework.
  - Appendix B.2: details of the BBSE method.
- Appendix C: related work and discussion.
  - Appendix C.1: related work of the offline label shift problem.
  - Appendix C.2: related work of non-stationary online convex optimization.
  - Appendix C.3: comparison on the risk estimator of our method and the work [16].
- Appendix D: omitted proofs for Section 3.1.
  - Appendix D.1: proof of Lemma 1.
  - Appendix D.2: high-probability bound of the risk estimator.
  - Appendix D.3: static regret for UOGD algorithm.
- Appendix E: omitted proofs for Section 3.2.
  - Appendix E.1: proof of Theorem 1.
  - Appendix E.2: proof of Theorem 2.
  - Appendix E.3: useful lemmas in the analysis of Theorem 1 and Theorem 2.
- Appendix F: algorithm details for ATLAS-ADA and omitted proofs for Section 3.3.
  - Appendix F.1: algorithm details for ATLAS-ADA and the design of the hint function.
  - Appendix F.2: proof of Theorem 3.
  - Appendix F.3: useful lemmas in the analysis of Theorem 3.
- Appendix G: technical lemmas.

A Omitted Details for Experiments

In this section, we mainly supplement the omitted details in Section 4. We first present the omitted numerical results on benchmark datasets in Appendix A.1. Then we list the experiment setups in Appendix A.2, followed by additional experimental results in Appendix A.3.

A.1 More Numerical Results for Section 4.2

Table 4 presents the numerical results omitted in Section 4.2, which reports the average error of different algorithms on various real-world datasets. The results show that the ATLAS and ATLAS-ADA methods outperform other contenders on all tasks. The advantage is particularly significant when the underlying class prior changes fast. The empirical results validate that our method can effectively adapt to the non-stationary label shift.

A.2 Experiment Setup

This subsection describes details of experimental setups, including contenders, simulated shifts, and datasets. All our experiments are run on a machine with 2 CPUs (24 cores for each).

Table 4: Average error (%) of different algorithms on various real-world datasets (`Sin` and `Squ`). We report the mean and standard deviation over five runs. The best algorithms are emphasized in bold. "•" indicates the algorithms that are significantly inferior to ATLAS-ADA by the paired t-test at a 5% significance level. Here AT-ADA represents ATLAS-ADA (with *OKM*). The online sample size is set as $N_t = 10$.

	`Sin`							`Squ`						
	FIX	**FTH**	**FTFWH**	**ROGD**	**UOGD**	**ATLAS**	**AT-ADA**	**FIX**	**FTH**	**FTFWH**	**ROGD**	**UOGD**	**ATLAS**	**AT-ADA**
ArXiv	• 31.58 ±0.10	• 30.63 ±0.24	• 31.90 ±0.22	• 28.35 ±0.32	• 25.64 ±0.18	• 26.03 ±0.16	**25.08** ±0.12	• 30.35 ±0.06	• 26.72 ±0.39	• 28.05 ±0.20	• 24.44 ±0.17	• 21.96 ±0.07	• 21.36 ±0.06	**20.80** ±0.06
EuroSAT	• 13.62 ±0.13	• 10.90 ±0.03	• 10.96 ±0.02	• 9.68 ±0.08	• 8.03 ±0.06	• 8.03 ±0.06	**7.97** ±0.08	• 14.15 ±0.11	• 10.22 ±0.08	• 10.26 ±0.06	• 8.91 ±0.05	• 7.30 ±0.07	• 6.97 ±0.08	**6.81** ±0.06
MNIST	• 1.81 ±0.02	• 1.46 ±0.03	• 1.47 ±0.03	• 1.46 ±0.03	• 1.30 ±0.04	1.28 ±0.03	**1.27** ±0.03	• 1.79 ±0.04	• 1.26 ±0.03	• 1.28 ±0.04	• 1.32 ±0.04	• 1.13 ±0.03	• 1.04 ±0.02	**1.01** ±0.04
Fashion	• 11.77 ±0.11	9.37 ±0.15	9.39 ±0.14	• 9.75 ±0.12	9.36 ±0.07	• 9.44 ±0.04	**9.32** ±0.04	• 11.92 ±0.09	• 8.24 ±0.09	• 8.35 ±0.07	• 8.63 ±0.07	• 8.42 ±0.04	• 8.05 ±0.07	**7.73** ±0.05
CIFAR10	• 21.40 ±0.09	• 18.57 ±0.07	• 18.62 ±0.08	• 19.16 ±0.12	• 18.17 ±0.07	• 18.01 ±0.07	**17.89** ±0.05	• 20.77 ±0.08	• 16.67 ±0.12	• 16.72 ±0.12	• 17.40 ±0.11	• 16.29 ±0.09	• 15.18 ±0.07	**14.84** ±0.05
CINIC10	• 35.29 ±0.19	• 31.17 ±0.12	• 31.20 ±0.12	• 31.46 ±0.14	• 30.22 ±0.10	• 30.15 ±0.11	**30.06** ±0.15	• 33.99 ±0.16	• 27.99 ±0.09	• 28.08 ±0.08	• 28.58 ±0.09	• 27.00 ±0.14	• 25.94 ±0.13	**25.56** ±0.12

Contenders. In the experiments, we mainly compare seven online label shift algorithms, including:

- *FIX* predicts with the fixed initialized classifier without any online updates.
- *ROGD* is a variant of OGD algorithm proposed by Wu et al. [16]. The algorithm constructs the risk estimator with the $0/1$ loss and uses a re-weighting classifier for the model update. (Detailed description and the comparison with our risk estimator can be found in Appendix C.3).
- *FTH* is short for Following The History algorithm proposed by Wu et al. [16]. The method takes the class prior for every iteration t as the average of all previously estimated priors.
- *FTFWH* is a variant of FTH proposed as a compared method in experiments of Wu et al. [16]. The method takes an average across previously estimated priors within a sliding window. In all experiments, the length of the sliding window is set as 100.
- *UOGD*/ATLAS/ATLAS-ADA: The OLaS algorithms proposed in Section 3.

Following our theoretical results, the step size of *ROGD* and *UOGD* is set to be $\Gamma/(G\sqrt{T})$, where G can be estimated during training the initial model and Γ is given by the decision set domain. In the experiments, we choose the decision domain as a ball with a fixed diameter for each dataset. For a fair comparison, all contenders use the same decision domain $\mathcal{W}$ with the diameter Γ set according to the parameter norm of the initial offline model f_0. The settings of step size pool of ATLAS and ATLAS-ADA are guided by Theorem 2 and Theorem 3, respectively. We employ a multinomial logistic regression classifier for *UOGD*, ATLAS, and ATLAS-ADA.

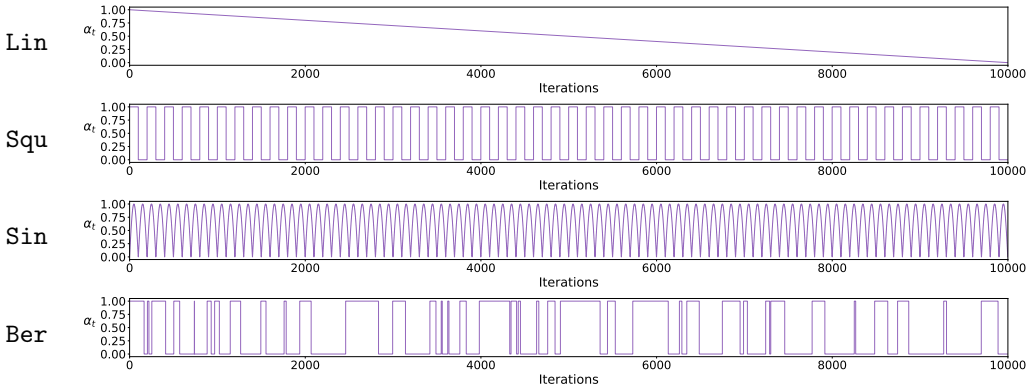


Figure 4: From top to bottom: `Linear Shift`, `Square Shift`, `Sine Shift`, `Bernoulli Shift`.

Simulated Shifts. We simulate four kinds of label shift patterns to capture different kinds of non-stationary environments. For each shift, the priors are a mixture of two different constant priors $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \Delta_K$ with a time-varying coefficient α_t: $\boldsymbol{\mu}_{y_t} = (1-\alpha_t)\boldsymbol{\mu}_1 + \alpha_t \boldsymbol{\mu}_2$, where $\boldsymbol{\mu}_{y_t}$ denotes the label distribution at round t and α_t controls the shift non-stationarity and patterns. We list the details in the following.

- `Linear Shift` (`Lin`): the parameter $\alpha_t = \frac{t}{T}$, which represents the gradually changed environments.
- `Square Shift` (`Squ`): the parameter α_t switches between 1 and 0 every $L/2$ rounds, where L is the periodic length. In the experiments, we set $L = \Theta(\sqrt{T})$ by default, which implies that the fluctuation of the class prior $V_T = \sum_{t=2}^{T} \|\boldsymbol{\mu}_{y_t} - \boldsymbol{\mu}_{y_{t-1}}\|_1$ is $\Theta(\sqrt{T})$. The `Square Shift` simulates a fast-changing environment with periodic patterns.
- `Sine Shift` (`Sin`): $\alpha_t = \sin \frac{i\pi}{L}$, where $i = t \bmod L$ and L is a given periodic length. In the experiments, we set $L = \Theta(\sqrt{T})$ by default. The `Sine Shift` also simulates a fast-changing environment with periodic patterns.
- `Bernoulli Shift` (`Ber`): At every iteration, we keep the $\alpha_t = \alpha_{t-1}$ with probability $p \in [0, 1]$ and otherwise set $\alpha_t = 1 - \alpha_{t-1}$. In the experiments, the parameter is set as $p = 1/\sqrt{T}$ by default, which implies $V_t = \Theta(\sqrt{T})$. The `Bernoulli Shift` simulates a fast-changing environment without periodic patterns.

Figure 4 demonstrates how α_t changes over time. We can observe that `Square Shift` and `Sine Shift` change in a periodic pattern while the others do not. Moreover, it can be seen that `Linear Shift` has simulated a much more tender class prior change than the other three shifts.

Datasets. We conduct experiments on synthetic data, six real-world datasets, and a real-life application of locomotion detection.

Synthetic data. There are three classes in the synthetic data, where the feature distribution of each class follows a Gaussian distribution. More specifically, for each instance $(\mathbf{x}_i, y_i)$ in the dataset S_t at round t, the label is generated from the discrete distribution defined by the given priors $\mathcal{D}_t(y)$, and the feature $\mathbf{x} \in \mathbb{R}^{12}$ is generated from the corresponding multivariate normal distributions $\mathcal{N}(\mu_y, \Sigma)$. We set $\boldsymbol{\mu}_1 = [1/3, 1/3, 1/3]$ and $\boldsymbol{\mu}_2 = [1, 0, 0]$ for simulated shifts.

Real-world datasets. We conduct experiments on six real-world datasets. For each dataset, we set $\boldsymbol{\mu}_1 = [1/K, 1/K, ..., 1/K]$ and $\boldsymbol{\mu}_2 = [1, 0, ..., 0]$ to generate the simulated shifts. Specifically, we include the following datasets.

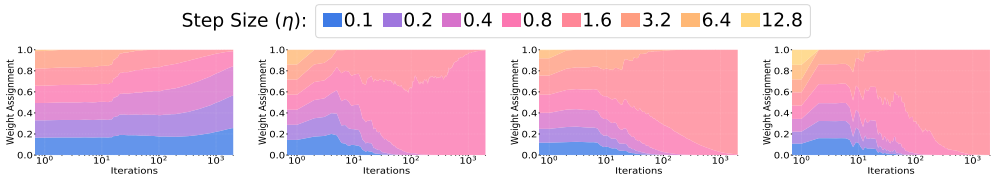
- **ArXiv** [5]**:** A paper classification dataset which contains metadata of the scholarly articles. We select 296, 708 papers from the computer science domain, which cover 23 classes, including cs.AI, cs.CC, cs.CL, cs.CR, cs.CV, cs.CY, cs.DB, cs.DC, cs.DM, cs.DS, cs.GT, cs.HC, cs.IR, cs.IT, cs.LG, cs.LO, cs.NE, cs.NI, cs.PL, cs.RO, cs.SE, cs.SI, and cs.SY. [6] We use a finetuned DistilBERT [36] to extract features from authors, titles and abstracts. The papers selected additionally to finetune the DistilBERT do not overlap with the offline and online datasets.
- **EuroSAT** [37]: A land cover classification dataset, which includes satellite images with the purpose of identifying the visible land use or land cover class. The dataset consists of 27, 000 satellite images from over 30 different European countries. It contains ten different classes, including industrial, residential, annual crop, permanent crop, river, sea and lake, herbaceous vegetation, highway, pasture, and forest. We use a finetuned ResNet [38] to extract features from the images of EuroSAT and the following four datasets. The images selected to train the ResNet also do not overlap with both the offline or online datasets.
- **MNIST** [39]: A widely-used image dataset of handwritten digits, which consists of 70, 000 grayscale images with 10 different classes.
- **Fashion** [40]: A dataset of 70, 000 grayscale fashion images, consisting of 10 different classes: T-shirt, trouser, shirt and sneaker, pullover, dress, coat, sandal, bag, and ankle boot.
- **CIFAR10** [41]: A dataset consists of 60, 000 color images in 10 classes, including airplane, automobile, ship, truck, bird, cat, deer, dog, frog, and horse.
- **CINIC10** [42]: A tiny ImageNet [43] dataset, which consists of images from CIFAR10 and ImageNet, and has the same ten classes as CIFAR10.

Real-life application. The real-life application is to distinguish human locomotion through the sensor data collected by the carry-on mobile phones[7]. The tabular data covers the sensor data (e.g., acceleration, gyroscope, magnetometer, orientation, gravity, pressure, altitude, and temperature) and

[5] www.kaggle.com/datasets/Cornell-University/arxiv

[6] See www.arxiv.org/archive/cs for the full name.

[7] www.shl-dataset.org



(a) `Linear shift` (b) `Square shift` (c) `Sine shift` (d) `Bernoulli shift`

Figure 5: Weight assigned of the ATLAS algorithm for each step size along the learning process. Different colors are used to indicate different step sizes.

the corresponding human motion and timestamp. We sample 30, 000 offline data and 77, 000 online data from 11 days, covering six classes, including still, walking, run, bike, car, and bus. During the online update, the online samples arrive in real chronological order based on the timestamp.

A.3 Additional Experimental Results

This part further reports the additional experimental results omitted in the main paper. Specifically, we show a supplement for the assigned weight visualization of ATLAS and a skyline to illustrate how the meta-algorithm tracks the proper base-algorithms.

Weight Assignment of ATLAS. In Figure 1, we show that slower `Linear Shift` is assigned relatively small step sizes, while faster `Bernoulli Shift` is assigned relatively large step sizes. Here, we replenish the experimental results for `Square Shift` and `Sine Shift`. As these two simulated shifts also change relatively fast like `Bernoulli Shift`, the assigned weights are also expected to be relatively large, which is in line with the experimental results in Figure 5(b) and 5(c).

Skyline of ATLAS-ADA. The purpose of this part is to answer the question of whether our proposed ATLAS-ADA can cope with online label shifts, regardless of how fast or slow the change is. To see this, we use the SHL dataset to simulate shifts with different non-stationarities, increasing from left to right side. We plot the final average errors of the major base-learners and ATLAS-ADA in Figure 6. Note that base-learners with much smaller or larger step sizes lead to worse results for all situations and are omitted for clarity. As shown in Figure 6, small step sizes achieve the best average errors for tender non-stationarity cases (left side) but fail for dramatic cases (right side), which is consistent with the theoretical results that the optimal step size scales with the non-stationarity measure V_T. Moreover, we find that the proposed ATLAS-ADA can always track the best base-learners, no matter tender cases or dramatic cases, which implies that we can safely employ it without knowing the change of the unknown environment in advance.

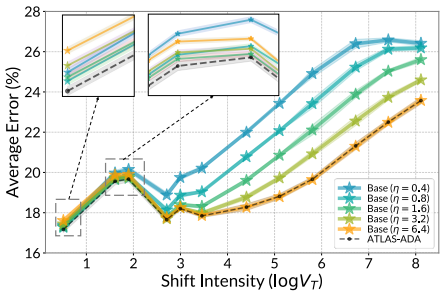


Figure 6: Skyline on the SHL dataset. The intensity of the label distribution shift increases from left to right. The black dashed line represents the average error of ATLAS-ADA, while the other lines represent that of different base-learners.

B Preliminaries

In this section, we first provide a brief introduction to the online convex optimization framework in Appendix B.1, and then in Appendix B.2 we present a detailed description of the BBSE method that is designed for the offline label shift problem, as well as its usage in constructing our risk estimator for the online label shift problem.

B.1 Brief Introduction to Online Convex Optimization

This section briefly introduces the online convex optimization framework. In many real-world tasks, data are often accumulated in a sequential way, which requires a learning paradigm to update the model in an online fashion. Taking the locomotion detection task as an example, each time, we

Algorithm 3 Online Gradient Descent (OGD)

Input: step size $\eta > 0$

1: Initialization: let $\mathbf{w}_{1,i}$ be any point in $\mathcal{W}$.
2: **for** $t = 1$ **to** T **do**
3: the learner plays $\mathbf{w}_t$ and observes loss $R_t(\mathbf{w}_t)$ and gradient $\nabla R_t(\mathbf{w}_t)$;
4: gradient-descent update:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}} \left[\mathbf{w}_t - \eta \nabla R_t(\mathbf{w}_t) \right].$$

5: **end for**

receive a few data collected by smartphone sensors and need to predict the motion type immediately. The sequential nature of the data makes it challenging to train a model offline using the standard machine learning methods. These circumstances can be modeled and handled with the Online Convex Optimization (OCO) framework [20, 15, 44].

The OCO framework can be viewed as a structured repeated game between a learner and the environment. During the learning process, the learner iteratively chooses decisions from a fixed convex decision set $\mathcal{W}$ based on the feedback from the environments. Let T denote the total number of game iterations, and the protocol of OCO is given as follows.

ONLINE CONVEX OPTIMIZATION FRAMEWORK

1: **for** $t = 1$ **to** T **do**
2: the learner chooses a decision $\mathbf{w}_t \in \mathcal{W}$, and the environments simultaneously choose a convex loss function $R_t : \mathcal{W} \mapsto \mathbb{R}$;
3: the learner suffers cost $R_t(\mathbf{w}_t)$ and observes certain information about R_t.
4: **end for**

The classic performance measure for the OCO algorithms is the (static) regret defined as

$$\mathbf{Reg}_T^{\mathbf{s}} = \sum_{t=1}^{T} R_t(\mathbf{w}_t) - \sum_{t=1}^{T} R_t(\mathbf{w}_*),$$

which compares the learner's prediction with the best single decision in hindsight $\mathbf{w}_* = \arg\min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} R_t(\mathbf{w})$. The static regret is a reasonable measure for stationary environments. But the performance measure could be too optimistic in non-stationary environments, where the underlying distribution changes over time, and the single best model could perform badly. Under such a case, a more suitable performance is the *dynamic regret* that competes with the online learner's performance with the best decisions for each round. A more detailed discussion and literature review for the dynamic regret can be found in Appendix C.2.

The OCO framework has received extensive studies over the decades. Among them, one of the most prominent algorithms is the Online Gradient Descent (OGD) algorithm [20], which takes a step in the opposite direction to the gradient of the previous risk at each iteration (See Algorithm 3 for the procedures). Despite its simplicity, OGD algorithm is powerful enough to handle a large family of online problems. Specifically, it enjoys a static regret of $\mathcal{O}(\sqrt{T})$ by setting the step size to be $\Theta(1/\sqrt{T})$, which was proven to be minimax optimal [45] for convex functions. Latter, tighter static bounds for loss functions with stronger curvature [46] and that can adapt to benign environments [47, 31, 32] were proposed. For more detailed reviews of the OCO framework, we refer the readers to the seminal books [44, 15].

B.2 BBSE Method for Online Label Shift

The Black Box Shift Estimation (BBSE) method [12] is a family of label shift algorithms that uses the confusion matrix of a given black-box classifier to estimate the label distributions from unlabeled samples. This part illustrates how to employ BBSE to the online label distribution estimation.

Recall that we have the labeled offline data S_0 and the unlabeled data S_t, and we need to estimate the label distribution $\boldsymbol{\mu}_{y_t}$. Note that $\boldsymbol{\mu}_{y_t}$ cannot be estimated directly due to labels of S_t being unavailable. Since S_0 is labeled, a possible way is to employ the labels of S_0. To this end, we introduce an initial

black-box model f_0 and have the following equation:

$$\begin{aligned}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t(\mathbf{x})}\left[\mathbb{1}\{f_0(\mathbf{x})=i\}\right] &= \sum_{j=1}^{K}[\boldsymbol{\mu}_{y_t}]_j \cdot \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t(\mathbf{x}\mid y=j)}\left[\mathbb{1}\{f_0(\mathbf{x})=i\}\right] \\ &= \sum_{j=1}^{K}[\boldsymbol{\mu}_{y_t}]_j \cdot \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_0(\mathbf{x}\mid y=j)}\left[\mathbb{1}\{f_0(\mathbf{x})=i\}\right], \end{aligned} \tag{7}$$

which holds for all $i \in [K]$. The first equality holds due to the law of total probability and the second equality is based on Assumption 1 that $\mathcal{D}_t(\mathbf{x} \mid y) = \mathcal{D}_0(\mathbf{x} \mid y)$ for any $t \in [T]$. (7) can be equivalently expressed in matrix notation, which is given by:

$$\boldsymbol{\mu}_{\widehat{y}_t} = C_{f_0}\boldsymbol{\mu}_{y_t}, \tag{8}$$

where $\boldsymbol{\mu}_{\widehat{y}_t} \in \Delta_K$ is the distribution vector of the black-box model's prediction under the distribution $\mathcal{D}_t(\mathbf{x})$, with the i-th entry $[\boldsymbol{\mu}_{\widehat{y}_t}]_i = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t(\mathbf{x})}\left[\mathbb{1}\{f_0(\mathbf{x})=i\}\right]$. $C_{f_0} \in \mathbb{R}^{K\times K}$ is the confusion matrix, whose (i,j)-th entry $[C_{f_0}]_{ij} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_0(\mathbf{x}\mid y=j)}[\mathbb{1}\{f_0(\mathbf{x})=i\}]$ is the classification rate that the initial model f_0 predicts samples from class i as class j. Here we assume the confusion matrix C_{f_0} is invertible. By solving (8), we have

$$\boldsymbol{\mu}_{y_t} = C_{f_0}^{-1}\boldsymbol{\mu}_{\widehat{y}_t}. \tag{9}$$

The LHS of (9) $\boldsymbol{\mu}_{y_t}$ is the label distribution we want. In the RHS, we note that the distribution $\mathcal{D}_t(\mathbf{x})$ and $\mathcal{D}_0(\mathbf{x} \mid y)$ is empirically accessible via the unlabeled data S_t and the labeled data S_0, so both C_{f_0} and $\boldsymbol{\mu}_{\widehat{y}_t}$ can be unbiasedly estimated with its empirical version. Specifically, C_{f_0} can be estimated with the offline labeled data S_0:

$$[\widehat{C}_{f_0}]_{ij} = \sum_{(\mathbf{x},y)\in S_0} \frac{\mathbb{1}\{f_0(\mathbf{x})=i \text{ and } y=j\}}{\mathbb{1}\{y=j\}}. \tag{10}$$

And $\boldsymbol{\mu}_{\widehat{y}_t}$ in the RHS can be estimated with the online data S_t, which is given by

$$[\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}]_j = \frac{1}{|S_t|}\sum_{\mathbf{x}\in S_t}\mathbb{1}\{f_0(\mathbf{x})=j\}. \tag{11}$$

With the above estimation, we construct the final estimator for the label distribution vector:

$$\widehat{\boldsymbol{\mu}}_{y_t} = \widehat{C}_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}.$$

Note that although BBSE has benign unbiasedness, it has limitations arising from the inverse operation, which might amplify slight errors in the confusion matrix. Other powerful unbiased estimators can also be applied to our OLaS method directly.

C Related Work and Discussion

This section introduces related works to our paper, including supervised learning with label shift in Appendix C.1, and online learning measures for non-stationary environments in Appendix C.2. Furthermore, in Appendix C.3, we clarify the main difference between Wu et al. [16] and our work.

C.1 Supervised Learning with Label Shift

Label shift is a typical scenario of learning with distribution change problem [1]. Most existing works have focused on the offline setting, where the label distribution varies from the source to target stages, but the label-conditional density remains the same. The main challenge of offline label shift problem lies in the estimation of the target label distribution. With such knowledge, the learner can train classifiers guaranteed to perform well over the target distribution. Saerens et al. [8] first introduced two kinds of solutions for label distribution estimation, including the confusion matrix-based method and the maximum likelihood estimation. Lipton et al. [12] provided a convergence analysis of the matrix-based method with black box models and Azizzadenesheli et al. [13] enhanced the matrix-based method by regularization. Another line of studies investigates the label distribution estimation problem via distribution matching [9, 10, 11, 14]. Garg et al. [14] further showed that the

matrix-based method can be interpreted from the distribution matching view. Beyond the label shift, the label distribution estimation with labeled data from the source domain and unlabeled data from the target domain is also known as the mixture proportion estimation problem [48, 49, 50], which has been wildly applied in learning with noisy label [51], positive and unlabeled learning [52, 53], and learning with unknown classes [54], etc.

The above methods only considered the scenario where the label distribution changes once, and sufficient unlabeled samples from the target distribution can be collected in advance. In many real-world applications, data are accumulated with time, and thus it is important to consider the online version of the label shift problem. Wu et al. [16] first introduced online label shift and considered the scenario where test samples come in sequence and label distribution changes with time. To address the OLaS problem, Wu et al. [16] proposed a risk estimator and used the vanilla OGD algorithm to learn over the online streams with label shift. They derived a static regret of order $\mathcal{O}(\sqrt{T})$, but the analysis crucially relied on the assumption of the convexity of the (expected) risk function, which is hard to be theoretically verified due to the use of 0/1-loss and the argmax operation. In contrast, by constructing an unbiased risk estimator with nice properties, our methods can adapt to dynamic environments with provable grantees. We provide more detailed discussions on the differences between the risk estimators in Appendix C.3.

C.2 Online Learning in Non-stationary Environments

This section first reviews existing results for non-stationary online convex optimization, including the worst-case dynamic regret and the universal dynamic regret. Then, we discuss the difference between our dynamic regret bounds and those in the previous works. We also present a brief introduction to readers unfamiliar with the OCO paradigm in Appendix B.1.

Worst-case Dynamic Regret. The rationale behind the static regret is that the best single decision could perform well over the time horizon. However, the underlying environments could change over the online learning process, making the static regret unsuitable. A better measure for the non-stationary environments is the worst-case dynamic regret $\mathbf{Reg}_T^{\mathbf{d}} = \sum_{t=1} R_t(\mathbf{w}_t) - \sum_{t=1}^T R(\mathbf{w}_t^*)$, which compares the learner's decision with the best decision at each iteration $\mathbf{w}_t^* = \arg\min_{\mathbf{w}\in\mathcal{W}} R_t(\mathbf{w})$ and has draw growing attention recently [20, 17, 22, 25, 55, 56, 57, 58, 23, 59].

It is well-known that a sublinear regret bound is impossible for the dynamic regret in the worst case unless imposing certain regularities on the non-stationarity of the environments. There are two commonly studied non-stationarity measures. The first one is the *path length* $P_T^* = \sum_{t=2}^T \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_2$, which reflects the fluctuation of the comparator sequence. When the loss functions are convex, Yang et al. [25] showed that OGD algorithm attains an $\mathcal{O}(\sqrt{T(1+P_T^*)})$ dynamic regret. The bound can be further improved to $\mathcal{O}(P_T^*)$ when the functions are convex and smooth, and the minimizer $\mathbf{w}_t^*$ lies in the interior of the feasible domain $\mathcal{W}$ [25], or the loss functions are strongly convex and smooth [60]. Another commonly studied non-stationarity measure is the *temporal variability* $V_T^f = \sum_{t=2}^T \sup_{\mathbf{w}\in\mathcal{W}} |f_t(\mathbf{w}) - f_{t-1}(\mathbf{w})|$, which quantifies the variation of the online functions. When the online functions are convex, Besbes et al. [17] showed an $\mathcal{O}(T^{2/3}{V_T^f}^{1/3})$ dynamic regret by the restarted OGD algorithm, and the bound can be further improved to $\mathcal{O}(T^{1/2}{V_T^f}^{1/2})$ for strongly convex functions. Then, the temporal variability bound was generalized to high-order deviations [57] and the p, q-norm [58]. There are also studies on the best-of-both-worlds bounds [22, 23, 59], where the path-length bound and temporal variability bound are achieved simultaneously by a single algorithm. For strongly convex and smooth functions, the best known result is $\mathcal{O}(\min\{P_T^*, S_T^*, V_T^f\})$ due to Zhao and Zhang [59], where $S_T^* = \sum_{t=2}^T \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_2^2$ is the squared path length.

Universal Dynamic Regret. The worst-case dynamic regret will sometimes be too pessimistic to guide the algorithm design as the function minimizer $\mathbf{w}_t^*$ can easily overfit to the noise [18]. To this end, the universal dynamic regret is proposed and defined as $\mathbf{Reg}_T^{\mathbf{d}}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=1}^T R_t(\mathbf{w}_t) - \sum_{t=1}^T R_t(\mathbf{u}_t)$, which supports the comparison to any feasible comparator sequence $\{\mathbf{u}_t\}_{t=1}^T$ with $\mathbf{u}_t \in \mathcal{W}$, and the measure has gained more and more attention [20, 18, 19, 61, 62, 63, 64, 65, 66, 67, 68, 69]. Zinkevich [20] showed that OGD can achieve an $\mathcal{O}((1+P_T)\sqrt{T})$ universal dynamic regret, where $P_T = \sum_{t=2}^T \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$ is the path length of the comparator sequence $\{\mathbf{u}_t\}_{t=1}^T$. Nevertheless,

the guarantee is far from the $\Omega(\sqrt{T(1+P_T)})$ minimax lower bound for convex functions as shown by Zhang et al. [18], who further developed an algorithm with an $\mathcal{O}(\sqrt{T(1+P_T)})$ universal dynamic regret and hence closed the gap [18]. Later, many studies are devoted to developing algorithms with tighter guarantees that can benefit from stronger curvature of loss functions [65], or can adapt to the benign environments [19, 63], or to the more challenging setting with bandit feedback [64].

Discussion on Our Dynamic Regret Bounds. Although there are many studies on non-stationary online learning, it is hard to apply existing methods to OLaS, and our regret bound is novel. First of all, although our performance measure shares seemingly similarity with the worst-case dynamic regret, the algorithm with such bounds is not suitable for the OLaS problem. The main reason is that existing methods require access to the loss function R_t for the model update. However, in the OLaS problem, the true loss function R_t is established on the inaccessible data distribution $\mathcal{D}_t$ and the learner can only approximate it with $\widehat{R}_t$ based on the empirical data. A direct application of algorithms with a worst-case dynamic regret can only lead to regret bounds defined over $\widehat{R}_t$ such that the learned model would suffer from severe overfitting on the sample noise. A few previous works [17, 25, 57] have attempted to develop methods to learn with the noisy loss function, but nevertheless the methods in [17, 25] require knowing the non-stationarity of the underlying distribution $\mathcal{D}_t$ ahead of time, which is generally unavailable.

Given that the comparators in our regret measure are *not* the optimizers of the loss functions, our dynamic regret measure is essentially a kind of *universal* dynamic regret rather than the worst-case dynamic regret. Our results differ from the existing universal dynamic regret bounds appearing in the literature by carefully exploiting the structure of online label shift. In particular, existing bounds always scale with the path length of compactors $P_T = \sum_{t=2}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$, whereas our dynamic regret can adapt to the variation of label distribution $V_T = \sum_{t=2}^{T} \|\boldsymbol{\mu}_{y_t} - \boldsymbol{\mu}_{y_{t-1}}\|_1$ directly, which is a more interpretable and suitable measure for OLaS.

From the technical side, the most related works to us are [18, 19] as all algorithms use the meta-base structure to achieve the dynamic regret bound. Besides the difference in the non-stationarity measure, our method improves the regret bound of Zhang et al. [18] for convex functions by replacing the dependence on T with the problem-dependent quantity G_T. The G_T quantity can be much smaller than T in benign environments, which implies a better adaptivity of our algorithm to underlying environments. The first dynamic regret bound with such adaptivity was achieved by Zhao et al. [19], but their work requires an additional smoothness assumption on the loss functions. We remove such a constraint via refined analyses for the meta-algorithm, which updates the weights with function values and an improved algorithm design for base-algorithm, which trains the model following an implicit update rule. The new finding might be of independent interest to the general online convex optimization. We note that a similar implicit update rule also appeared in a concurrent work [70, Section 5] for achieving problem-dependent *static* regret. Our dynamic regret is much more challenging, as we not only need an implicit update in base-algorithms but also need a refined analysis of the meta-algorithm, which is unique in online ensemble analysis of dynamic regret minimization.

C.3 Detailed Comparison with Wu et al. [16]

As mentioned in the main paper the work of Wu et al. [16] is the closest one to our proposal, who also proposed a risk estimator to cope with OLaS. In this part, we first review their result by restating their proposed risk estimator and then elaborate on the salient differences between their method and ours. The comparison demonstrates both theoretical and practical appeals of our proposed risk estimator in that it makes the OCO framework still applicable for OLaS.

Estimation of $0/1$ risk. The performance measure of Wu et al. [16] is slightly different from ours, where the authors aim to minimize the regret defined over the $0/1$ risk

$$\mathbf{Reg}_T^{01} = \sum_{t=1}^{T} R_t^{01}(g_t) - \min_{g \in \mathcal{G}} \sum_{t=1}^{T} R_t^{01}(g).$$

In above, the $0/1$ risk is defined as $R_t^{01}(g) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}[\mathbb{1}\{g(\mathbf{x}) \neq y\}]$ and $g : \mathcal{X} \mapsto \mathcal{Y}$ is the classifier whose definition will be clear latter. Denoting by $\boldsymbol{\mu}_{y_t}$ the class prior vector with

$[\boldsymbol{\mu}_{y_t}]_k = \mathcal{D}_t(y=k)$, the expected $0/1$ risk can be rewritten as

$$\begin{aligned} R_t^{01}(g) &= \sum_{k=1}^{K} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t(\mathbf{x}\mid y=k)} \left[\mathbb{1}\{g(\mathbf{x}) \neq k\}\right] \cdot [\boldsymbol{\mu}_{y_t}]_k \\ &= \sum_{k=1}^{K} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_0(\mathbf{x}\mid y=k)} \left[\mathbb{1}\{g(\mathbf{x}) \neq y\}\right] \cdot [\boldsymbol{\mu}_{y_t}]_k \,. \end{aligned} \tag{12}$$

In decomposition (12), the first term $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_0(\mathbf{x}\mid y=k)}[\mathbb{1}\{g(\mathbf{x}) \neq y\}]$ is established on the class-conditional distribution $\mathcal{D}_0(\mathbf{x}\mid y)$, which can be empirically approximated with the offline data S_0. The second term is the class prior at each iteration. One can estimate it with the BBSE estimator introduced in Section 3. As a consequence, Wu et al. [16] empirically approximated $R_t^{01}(g)$ by

$$\widehat{R}_t^{01}(g) = \sum_{k=1}^{K} [\widehat{\boldsymbol{\mu}}_{y_t}]_k \cdot \widehat{R}_{t,k}^{01}(g),$$

where $\widehat{\boldsymbol{\mu}}_{y_t} = C_{f_0}^{-1} \cdot \widehat{\boldsymbol{\mu}}_{\widehat{y}_t}$ is the estimated class prior returned by the BBSE model and $\widehat{R}_{t,k}^{01} = \frac{1}{|S_0^k|}\sum_{\mathbf{x}_n\in S_0^k} \mathbb{1}\{g(\mathbf{x}_n) \neq k\}$ is the model's empirical risk over S_0^k.

Reweighting classifier. To directly optimize the $0/1$ loss, Wu et al. [16] further restricted the model's structure by focusing on the reweighting algorithm. The basic idea is, the optimal classifier that minimizes $R_t^{01}(g)$ is $g_*(\mathbf{x}) = \arg\max_{k\in[K]} \mathcal{D}_t(y=k\mid\mathbf{x})$. Under the class shift condition, they rewrote $\mathcal{D}_t(y=k\mid\mathbf{x})$ as

$$\begin{aligned} \mathcal{D}_t(y=k\mid\mathbf{x}) &= \frac{\mathcal{D}_t(y=k)\mathcal{D}_t(\mathbf{x}\mid y=k)}{\mathcal{D}_t(\mathbf{x})} = \frac{\mathcal{D}_t(y=k)\mathcal{D}_0(\mathbf{x}\mid y=k)}{\mathcal{D}_t(\mathbf{x})} \\ &= \frac{\mathcal{D}_t(y=k)}{\mathcal{D}_0(y=k)}\frac{\mathcal{D}_0(\mathbf{x})}{\mathcal{D}_t(\mathbf{x})}\mathcal{D}_0(y=k\mid\mathbf{x}) \propto \frac{\mathcal{D}_t(y=k)}{\mathcal{D}_0(y=k)}\mathcal{D}_0(y=k\mid\mathbf{x}) \,. \end{aligned} \tag{13}$$

Since the $\mathcal{D}_0$ can be approximate by the offline data S_0, the leaner can train a probability model $g_0 : \mathcal{X} \mapsto \Delta_K$ on S_0 to approximate $\mathcal{D}_0(y\mid\mathbf{x})$. Then, by the relationship (13), one can construct a classifier $g_{\mathbf{p}} : \mathcal{X} \mapsto [K]$ parameterized by the reweighting parameter $\mathbf{p} \in \Delta_K$ to approximate the optimal classifier g_* by

$$g_{\mathbf{p}}(\mathbf{x}) = \arg\max_{y\in[K]} \frac{1}{Z(\mathbf{x})}\frac{[\mathbf{p}]_k}{\mathcal{D}_0(y=k)} \cdot [g_0(\mathbf{x})]_k, \tag{14}$$

where $Z(\mathbf{x}) = \sum_{k=1}^{K} \frac{[\mathbf{p}]_k}{\mathcal{D}_0(y=k)} \cdot [g_0(\mathbf{x})]_k$ is the normalization factor and the initial label distribution $\mathcal{D}_0(y=k)$ can be estimated with the offline data S_0. When g_0 provides a sufficiently accurate approximation of $\mathcal{D}_t(y\mid\mathbf{x})$, one can show that $g_* = \arg\min_{\mathbf{p}\in\Delta_K} R_t^{01}(g_{\mathbf{p}})$.

Comparisons. In summary, the differences between the risk estimator of Wu et al. [16] and ours are mainly in the following two aspects:

(i) **Loss function.** As shown in (12), Wu et al. [16] used the non-convex $0/1$ loss $\mathbb{1}\{g(\mathbf{x}) \neq y\}$ for evaluation. Theoretically, the non-convexity of $0/1$ loss makes it hard to confirm the convexity of the $0/1$ risk, which is the critical condition for the regret bound. Practically, the gradient of $0/1$ loss is always zero, which brings troubles to the optimization. By contrast, we use a convex surrogate loss in our risk estimator to address the above problems.

(ii) **Classifier.** To match the 0/1 loss function, Wu et al. [16] designed a classifier based on the reweighting algorithm, which directly updates the reweighting parameter $\mathbf{p}$ in a simplex. Theoretically, as shown in (14), there is a troublesome argmax operation in the formulation of the classifier, which makes it non-convex and then hurts the convexity of the entire risk. Practically, only adjusting the outputs of the initial classifier $f_0^{\mathbf{p}}$ by reweighting might limit the learning capability of the model, especially when the initial classifier does not perform well enough. By comparison, we employ a strictly convex classifier and update the whole parameters in a convex parameter space, which confirms the convexity of the risk and the flexibility to better adapt to the environmental changes. The classifier in (14) is hard to be convex even if we replace the 0/1 loss with convex surrogate loss (details elaborated below).

We finally remark that even if we modify the risk estimator of Wu et al. [16] by optimizing the convex surrogate loss rather than the $0/1$ loss, it remains unclear how to apply the OCO framework to this modified estimator. The critical challenge is that even if the argmax operation can be avoided, another renormalization process is required due to the reweighting mechanism, making it hard to confirm the convexity of online functions. By contrast, our risk estimator well fits the OCO framework and thus can achieve favorable dynamic regret guarantees with suitable online update rules.

D Omitted Details for Section 3.1

This section presents the omitted details for Section 3.1.

D.1 Proof of Lemma 1

Proof of Lemma 1. Recall the definition of the expected risk (2) as

$$R_t(\mathbf{w}) = \sum_{k=1}^{K} [\boldsymbol{\mu}_{y_t}]_k \cdot R_0^k(\mathbf{w}),$$

where $\boldsymbol{\mu}_{y_t}$ is the true class prior at iteration t with $[\boldsymbol{\mu}_{y_t}]_k = \mathcal{D}(y = k)$. Besides, the risk estimator (3) is defined by

$$\widehat{R}_t(\mathbf{w}) = \sum_{k=1}^{K} [\widehat{C}_{f_0}^{-1} \widehat{\boldsymbol{\mu}}_{\widehat{y}_t}]_k \cdot \widehat{R}_0^k(\mathbf{w}),$$

where $\widehat{\boldsymbol{\mu}}_{\widehat{y}_t} \in \Delta_K$ with $[\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}]_k = (1/|S_t|) \sum_{\mathbf{x} \in S_t} \mathbb{1}\{f_0(\mathbf{x}) = k\}$ is the estimated class prior of the prediction $f_0(\mathbf{x})$.

Given that the initial offline data has sufficient samples such that $\widehat{C}_{f_0} = C_{f_0}$ and $\widehat{R}_0^k(\mathbf{w}) = R_0^k(\mathbf{w})$ and the model $\mathbf{w} \in \mathcal{W}$ is independent of S_t, we have

$$\begin{aligned}
\mathbb{E}_{S_t \sim \mathcal{D}_t}[\widehat{R}_t(\mathbf{w})] &= \mathbb{E}_{S_t \sim \mathcal{D}_t} \left[\sum_{k=1}^{K} [\widehat{C}_{f_0}^{-1} \widehat{\boldsymbol{\mu}}_{\widehat{y}_t}]_k \cdot \widehat{R}_0^k(\mathbf{w}) \right] \\
&= \sum_{k=1}^{K} \left[\widehat{C}_{f_0}^{-1} \mathbb{E}_{S_t \sim \mathcal{D}_t} \left[\widehat{\boldsymbol{\mu}}_{\widehat{y}_t} \right] \right]_k \cdot \widehat{R}_0^k(\mathbf{w}) \\
&= \sum_{k=1}^{K} \left[\widehat{C}_{f_0}^{-1} \boldsymbol{\mu}_{\widehat{y}_t} \right]_k \cdot \widehat{R}_0^k(\mathbf{w}),
\end{aligned}$$

where the first equality is due to the definition of $\widehat{R}_t(\mathbf{w})$. The second inequality comes from the fact that $\widehat{C}_{f_0}$ and $\widehat{R}_0^k$ are independent of S_t. The last equality is a consequence of the unbiasedness of $\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}$, i.e., $\mathbb{E}_{S_t \sim \mathcal{D}_t}[\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}] = \boldsymbol{\mu}_{\widehat{y}_t}$, where $[\boldsymbol{\mu}_{\widehat{y}_t}]_k = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_t(\mathbf{x})}[\mathbb{1}\{f_0(\mathbf{x}) = k\}]$ is the expected class prior of the prediction $f_0(\mathbf{x})$.

Then, under the condition that $\widehat{C}_{f_0} = C_{f_0}$ and $\widehat{R}_0^k(\mathbf{w}) = R_0^k(\mathbf{w})$, we can further obtain

$$\mathbb{E}_{S_t \sim \mathcal{D}_t}[\widehat{R}_t(\mathbf{w})] = \sum_{k=1}^{K} \left[C_{f_0}^{-1} \boldsymbol{\mu}_{\widehat{y}_t} \right]_k \cdot R_0^k(\mathbf{w}) = \sum_{k=1}^{K} [\boldsymbol{\mu}_{y_t}]_k \cdot R_0^k(\mathbf{w}) = R_t(\mathbf{w}),$$

where the second equality is due to the relationship $\boldsymbol{\mu}_{y_t} = C_{f_0}^{-1} \boldsymbol{\mu}_{\widehat{y}_t}$ as introduced by (8). The last inequality comes from the definition of $R_t(\mathbf{w})$, which completes the proof. □

D.2 Concentration of Risk Estimator

In this section, we show that the risk estimator converges to the expected risk at the rate of $O(\sqrt{1/|S_0|})$ with high probability.

Lemma 2. *Let $\delta \in (0, 1/4]$. For any $\mathbf{w} \in \mathcal{W}$ independent of the dataset S_t, with probability at least $1 - (K+3)\delta$, the risk estimator $\widehat{R}_t(\mathbf{w})$ in Eq. (3) satisfies*

$$|\mathbb{E}_{S_t \sim \mathcal{D}_t}[\widehat{R}_t(\mathbf{w})] - R_t(\mathbf{w})| \leq \frac{4\sqrt{K}B}{\kappa^2\sigma^2}\left(\frac{\log(2K/\delta)}{|S_0|} + \sqrt{\frac{2\log(2K/\delta)}{|S_0|}}\right),$$

given that $|S_0| \geq (25\log(2K/\delta))/(\kappa\sigma)^2$, where $\kappa = \min_{k\in[K]} \mathcal{D}_0(y=k)$ and σ is the minimum singular value of C_{f_0}.

Proof of Lemma 2. For notation convenience, we denote by $\boldsymbol{R}_0 : \mathcal{W} \mapsto \mathbb{R}^K$ the vector risk function with its k-th entry $[\boldsymbol{R}_0(\mathbf{w})]_k = R_0^k(\mathbf{w})$ and its empirical version $\widehat{\boldsymbol{R}}_0 : \mathcal{W} \mapsto \mathbb{R}^K$ with $[\widehat{\boldsymbol{R}}_0(\mathbf{w})]_k = \widehat{R}_0^k(\mathbf{w})$. Under such a case, we can rewrite the risk estimator as $\widehat{R}_t(\mathbf{w}) = \widehat{\boldsymbol{\mu}}_{\widehat{y}_t}^\top \widehat{C}_{f_0}^{-1}\widehat{\boldsymbol{R}}_0(\mathbf{w})$ and the expected risk as $R_t(w) = \boldsymbol{\mu}_{\widehat{y}_t}^\top C_{f_0}^{-1}\boldsymbol{R}_0(\mathbf{w})$. Then, since S_t is independent of S_0, we have

$$\mathbb{E}_{S_t\sim\mathcal{D}_t}[\widehat{R}_t(\mathbf{w})] = \mathbb{E}_{S_t\sim\mathcal{D}_t}[\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}^\top \widehat{C}_{f_0}^{-1}\widehat{\boldsymbol{R}}_0(\mathbf{w})] = \mathbb{E}_{S_t\sim\mathcal{D}_t}[\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}^\top]\widehat{C}_{f_0}^{-1}\widehat{\boldsymbol{R}}_0(\mathbf{w}) = \boldsymbol{\mu}_{\widehat{y}_t}^\top \widehat{C}_{f_0}^{-1}\widehat{\boldsymbol{R}}_0(\mathbf{w}).$$

Let $W_{f_0} \in \mathbb{R}^K$ be the diagonal matrix with $[W_{f_0}]_{ii} = \mathcal{D}_0(y = i)$ and $[J_{f_0}]_{ij} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_0}[\mathbb{1}\{f_0(\mathbf{x}) = i \text{ and } y = j\}]$ be the confusion matrix defined over the joint distribution of $f_0(\mathbf{x})$ and y. Besides, denote by $[\widehat{W}_{f_0}]_{ii} = (1/|S_0|)\sum_{(\mathbf{x},y)\in S_0}\mathbb{1}\{y = i\}$ and $\widehat{J}_{f_0} = (1/|S_0|)\sum_{(\mathbf{x},y)\in S_0}\mathbb{1}\{f_0(\mathbf{x}) = i \text{ and } y = j\}$ the empirical versions. Definition (10) indicates that

$$J_{f_0} = W_{f_0}\cdot C_{f_0} \quad \text{and} \quad \widehat{J}_{f_0} = \widehat{W}_{f_0}\cdot\widehat{C}_{f_0}. \tag{15}$$

Under such a case, the empirical and expected risk can be further rewritten as

$$\mathbb{E}_{S_t\sim\mathcal{D}_t}[\widehat{R}_t(\mathbf{w})] = \boldsymbol{\mu}_{\widehat{y}_t}^\top \widehat{J}_{f_0}^{-1}\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w}) \quad \text{and} \quad R_t(\mathbf{w}) = \boldsymbol{\mu}_{\widehat{y}_t}^\top J_{f_0}^{-1}W_{f_0}\boldsymbol{R}_0(\mathbf{w}).$$

So, we have

$$\begin{aligned}
&\mathbb{E}_{S_t\sim\mathcal{D}_t}\left[\widehat{R}_t(\mathbf{w})\right] - R_t(\mathbf{w}) \\
&\leq \boldsymbol{\mu}_{\widehat{y}_t}^\top \widehat{J}_{f_0}^{-1}\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w}) - \boldsymbol{\mu}_{\widehat{y}_t}^\top J_{f_0}^{-1}W_{f_0}\boldsymbol{R}_0(\mathbf{w}) \\
&\leq \underbrace{\boldsymbol{\mu}_{\widehat{y}_t}^\top \widehat{J}_{f_0}^{-1}\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w}) - \boldsymbol{\mu}_{\widehat{y}_t}^\top J_{f_0}^{-1}\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w})}_{\texttt{term (a)}} + \underbrace{\boldsymbol{\mu}_{\widehat{y}_t}^\top J_{f_0}^{-1}\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w}) - \boldsymbol{\mu}_{\widehat{y}_t}^\top J_{f_0}^{-1}W_{f_0}\boldsymbol{R}_0(\mathbf{w})}_{\texttt{term (b)}}.
\end{aligned}$$

For term (a), with probability at least $1-\delta$, we have

$$\begin{aligned}
\texttt{term (a)} &= \boldsymbol{\mu}_{\widehat{y}_t}^\top (J_{f_0}^{-1} - \widehat{J}_{f_0}^{-1})\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w}) \\
&\leq \|\boldsymbol{\mu}_{\widehat{y}_t}\|_2 \cdot \|\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w})\|_2 \cdot \|\widehat{J}_{f_0}^{-1} - J_{f_0}^{-1}\|_2 \\
&\leq \sqrt{K}B\|\widehat{J}_{f_0}^{-1} - J_{f_0}^{-1}\|_2 \\
&\leq \frac{2\sqrt{K}B}{\kappa^2\sigma^2}\left(\frac{\log(2K/\delta)}{|S_0|} + \sqrt{\frac{2\log(2K/\delta)}{|S_0|}}\right),
\end{aligned}$$

where the second inequality is due to the Cauchy-Schwarz inequality and the property of the operator norm. The third inequality holds since $\|\boldsymbol{\mu}_{\widehat{y}_t}\|_2 \leq 1$ and $\|\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w})\|_2 \leq \|\widehat{W}_{f_0}\|_2 \cdot \|\widehat{\boldsymbol{R}}_0(\mathbf{w})\|_2 \leq \sqrt{K}B$. The last inequality is due to Lemma 4.

Next, we analyze the term (b).

$$\begin{aligned}
\texttt{term (b)} &= \boldsymbol{\mu}_{\widehat{y}_t}^\top J_{f_0}^{-1}(\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w}) - W_{f_0}\boldsymbol{R}_0(\mathbf{w})) \\
&\leq \|\boldsymbol{\mu}_{\widehat{y}_t}\|_2 \cdot \|J_{f_0}^{-1}\|_2 \cdot \|\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w}) - W_{f_0}\boldsymbol{R}_0(\mathbf{w})\|_2 \\
&\leq \frac{1}{\kappa\sigma}\|\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w}) - W_{f_0}\boldsymbol{R}_0(\mathbf{w})\|_2,
\end{aligned}$$

where the third inequality is due to the fact that $\|\boldsymbol{\mu}_{\widetilde{y}_t}\|_2 \leq 1$ and $\|J_{f_0}^{-1}\|_2 \leq \|W_{f_0}^{-1}C_{f_0}^{-1}\|_2 \leq 1/(\kappa\sigma)$. The term $\|\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w}) - W_{f_0}\boldsymbol{R}_0(\mathbf{w})\|_2$ can be further decomposed as

$$\begin{aligned}\|\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w}) - W_{f_0}\boldsymbol{R}_0(\mathbf{w})\|_2 &= \|\widehat{W}_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w}) - W_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w})\|_2 + \|W_{f_0}\widehat{\boldsymbol{R}}_0(\mathbf{w}) - W_{f_0}\boldsymbol{R}_0(\mathbf{w})\|_2 \\ &\leq \|\widehat{\boldsymbol{R}}_0(\mathbf{w})\|_2 \cdot \|\widehat{W}_{f_0} - W_{f_0}\|_2 + \|W_{f_0}\|_2 \cdot \|\widehat{\boldsymbol{R}}_0(\mathbf{w}) - \boldsymbol{R}_0(\mathbf{w})\|_2 \\ &\leq B\|\widehat{W}_{f_0} - W_{f_0}\|_2 + \|\widehat{\boldsymbol{R}}_0(\mathbf{w}) - \boldsymbol{R}_0(\mathbf{w})\|_2,\end{aligned}$$

where the first inequality is due to the property of the operator norm. The second inequality is a consequence of the fact that $\|\widehat{\boldsymbol{R}}(\mathbf{w})\|_2 \leq B$ for any $\mathbf{w} \in \mathcal{W}$ and $\|W_{f_0}\|_2 \leq 1$.

Then, by the matrix Bernstein inequality in [71, Theorem 1.4], we have

$$\|W_{f_0} - \widehat{W}_{f_0}\|_2 \leq \frac{2\log(2K/\delta)}{3|S_0|} + \sqrt{\frac{2\log(2K/\delta)}{|S_0|}} \tag{16}$$

with probability at least $1-\delta$.

Now, we turn to bound the term $\|\widehat{\boldsymbol{R}}_0(\mathbf{w}) - \boldsymbol{R}_0(\mathbf{w})\|_2$. Let $N_{\min} = \min\{|S_0^1|, \ldots, |S_0^K|\}$. According to Lemma 5, we can first show that $N_{\min} \geq (\kappa|S_0|)/2$ with probability at least $1-\delta$ for any $\delta \in (0, 1/4]$. Then, we can bound the last term by

$$\|\widehat{\boldsymbol{R}}_0(\mathbf{w}) - \boldsymbol{R}_0(\mathbf{w})\|_2 \leq \sqrt{K}\max_{k\in[K]}\{|R_0^k(\mathbf{w}) - \widehat{R}_0^k(\mathbf{w})|\}.$$

According to the Hoeffding's lemma, for each $k \in [K]$, we can bound

$$|R_0^k(\mathbf{w}) - \widehat{R}_0^k(\mathbf{w})| \leq \sqrt{\frac{B^2\log(1/\delta)}{2|S_0^k|}} \leq \sqrt{\frac{B^2\log(1/\delta)}{2N_{\min}}}$$

with probability at least $1-\delta$. Combining the K events, which holds with probability $1-\delta$, and the event that $N_{\min} \geq (\kappa|S_0|)/2$, we can obtain that

$$\|\widehat{\boldsymbol{R}}_0(\mathbf{w}) - \boldsymbol{R}_0(\mathbf{w})\|_2 \leq \sqrt{\frac{KB^2\log(1/\delta)}{\kappa|S_0|}} \tag{17}$$

with probability at least $1-(K+1)\delta$.

Putting (16) and (17) together, we have

$$\texttt{term (b)} \leq \frac{B}{\kappa\sigma}\left(\frac{2\log(2K/\delta)}{3|S_0|} + \sqrt{\frac{2\log(2K/\delta)}{|S_0|}} + \sqrt{\frac{K\log(1/\delta)}{\kappa|S_0|}}\right)$$

with probability at least $1-(K+2)\delta$. Combining the upper bounds for term (a) and term (b), we can complete the proof by

$$|\mathbb{E}_{S_t\sim\mathcal{D}_t}[\widehat{R}_t(\mathbf{w})] - R_t(\mathbf{w})| \leq \frac{4\sqrt{K}B}{\kappa^2\sigma^2}\left(\frac{\log(2K/\delta)}{|S_0|} + \sqrt{\frac{2\log(2K/\delta)}{|S_0|}}\right)$$

with probability at least $1-(K+3)\delta$.

□

D.3 Static Regret Bound of UOGD Algorithm

The following lemma presents the static regret guarantees of the UOGD algorithm when the step size is set as $\eta = \Theta(T^{-1/2})$.

Lemma 3. *Under the same assumptions of Lemma 1, UOGD in Eq. (4) with a step size $\eta = \Theta(T^{-1/2})$ satisfies*

$$\mathbb{E}_{1:T}\left[\mathbf{Reg}_T^{\mathbf{s}}\right] \leq \frac{2G\Gamma}{\sigma}\sqrt{KT} = \mathcal{O}(\sqrt{T}),$$

where $\mathbb{E}_{1:T}[\cdot]$ denotes the expectation taken over the random draw of dataset $\{S_t\}_{t=1}^T$. The constant $\sigma > 0$ denotes the minimum singular value of confusion matrix C_{f_0}.

D.4 Useful Lemmas

This section presents several useful lemmas for the proofs in Appendix D.

Lemma 4. *Let J_{f_0} and $\widehat{J}_{f_0}$ be the matrix defined as (15). Then, for any $\delta \in [0.1)$, with probability at least $1-\delta$ we have*

$$\|\widehat{J}_{f_0}^{-1} - J_{f_0}^{-1}\|_2 \leq \frac{2\log(2K/\delta)}{\kappa^2\sigma^2|S_0|} + \frac{2}{\kappa^2\sigma^2}\sqrt{\frac{2\log(2K/\delta)}{|S_0|}},$$

given that $|S_0| \geq (25\log(2K/\delta))/(\kappa\sigma)^2$, where $\kappa = \min_{k\in[K]} \mathcal{D}_0(y=k)$ and σ is the minimum singular value of C_{f_0}.

Proof of Lemma 4. Denoting by $\Delta_J = \widehat{J}_{f_0} - J_{f_0}$, according to [13, Lemma 2], we have

$$\|\Delta_J\|_2 \leq \frac{2\log(2K/\delta)}{3|S_0|} + \sqrt{\frac{2\log(2K/\delta)}{|S_0|}} \leq \frac{\kappa\sigma}{2}$$

with probability at least $1-\delta$, where the last inequality is due to the assumption that $|S_0| \geq (\log(2K/\delta))/(\sigma\kappa)^2$. We can check the condition for Lemma 10

$$\|J_{f_0}^{-1}\Delta_J\|_2 \leq \|C_{f_0}^{-1}\|_2 \cdot \|W_{f_0}^{-1}\|_2 \cdot \|\Delta_J\|_2 \leq \frac{1}{\kappa\sigma}\cdot\frac{\kappa\sigma}{2},$$

where the last inequality holds since σ is the minimum singular value of C_{f_0} and κ is that of W_{f_0}.

Then, according to Lemma 10, we can complete the proof by

$$\|\widehat{J}_{f_0}^{-1} - J_{f_0}^{-1}\|_2 \leq 2\|J_{f_0}^{-1}\|_2^2\|\Delta\|_2 \leq \frac{2\log(2K/\delta)}{\kappa^2\sigma^2|S_0|} + \frac{2}{\kappa^2\sigma^2}\sqrt{\frac{2\log(2K/\delta)}{|S_0|}}.$$

□

Lemma 5. *Let $|S_0| \geq (16\log(1/\delta))/\kappa^2$. Then, for any $\delta \in (0, 1/4]$, with probability at least $1-\delta$ the following holds:*

$$N_{\min} \geq \frac{\kappa|S_0|}{2},$$

where $N_{\min} = \min\{|S_0^1|, \ldots, |S_0^K|\}$ and $\kappa = \min_{k\in[K]} \mathcal{D}_0(y=k)$.

Proof of Lemma 5. For notation simplicity, we denote by $\widetilde{\boldsymbol{\mu}}_0$ the empirical class prior at $t=0$, i.e. $[\widetilde{\boldsymbol{\mu}}_0]_k = |S_0^k|/|S_0|$ and $\boldsymbol{\mu}_0$ the expected class prior with $[\boldsymbol{\mu}_0]_k = \mathcal{D}_0(y=k)$. Then, according to [72, Proposition 19], we have

$$\|\widetilde{\boldsymbol{\mu}}_0 - \boldsymbol{\mu}_0\|_2 \leq \frac{1}{\sqrt{|S_0|}} + \sqrt{\frac{\log(1/\delta)}{|S_0|}}$$

with probability at least $1-\delta$, which implies that

$$[\widetilde{\boldsymbol{\mu}}_0]_k \geq [\boldsymbol{\mu}_0]_k - \frac{1}{\sqrt{|S_0|}} - \sqrt{\frac{\log(1/\delta)}{|S_0|}}, \quad \forall k \in [K]$$

holds with probability at least $1-\delta$. We complete the proof by noticing that $[\boldsymbol{\mu}_0]_k \geq \kappa$ holds for any $k \in [K]$ and further noting the assumption that $|S_0| \geq \frac{16\log(1/\delta)}{\kappa^2}$. □

E Omitted Proofs for Section 3.2

This section provides the proof omitted in section 3.2. We will first present the regret bound for the UOGD algorithm without step size tuning (Theorem 1), and then show the overall dynamic regret bound of the ATLAS algorithm (Theorem 2). At the end of this section, we list several useful lemmas.

Before presenting the proofs, we highlight the main challenge of the regret analysis. As noted by Section 3.2 and Appendix C.2, our performance measure (1) is different from the conventional notion of (worst-case) dynamic regret [22], where the performance of the learned model is measured by the observed loss function. However, in the OLaS problem, the model is desired to perform well over the underlying distribution, and its quality is evaluated over the expected risk function $R_t(\cdot)$. The main challenge here is that the expected risk function $R_t(\cdot)$ is unavailable in the learning process, which makes it hard to apply the analysis of (worst-case) dynamic regret for our case.

E.1 Proof of Theorem 1

This part presents the proof of Theorem 1. The key idea is to decompose the overall dynamic regret into two parts by introducing a reference sequence, which changes in a piece-wise stationary manner. In our analysis, the first part measures the regret of our algorithm when compared with the reference sequence, and the second part reflects the quality of the reference sequence. Since the reference sequence appears only in the analysis, we can make a tight regret bound by choosing a proper one to balance the first and second parts.

Proof of Theorem 1. We can decompose the regret bound into two parts by introducing a reference sequence $\{\mathbf{u}_t\}_{t=1}^T$ that only changes every Δ iteration. More specifically, denoting by $\mathcal{I}_m = [(m-1)\Delta+1, m\Delta]$ the m-th time interval, any comparator $\mathbf{u}_t$ that falls into $\mathcal{I}_m$ is taken as the single best decision over the interval, i.e. $\mathbf{u}_t = \mathbf{w}^*_{\mathcal{I}_m} = \arg\min_{\mathbf{w}\in\mathcal{W}} \sum_{t\in\mathcal{I}_m} R_t(\mathbf{w})$ for any $t \in \mathcal{I}_m$. We have

$$
\begin{aligned}
&\mathbb{E}_{1:T}\left[\sum_{t=1}^T R_t(\mathbf{w}_t)\right] - \sum_{t=1}^T R_t(\mathbf{w}_t^*) \\
&= \mathbb{E}_{1:T}\left[\sum_{t=1}^T R_t(\mathbf{w}_t)\right] - \sum_{t=1}^T R_t(\mathbf{u}_t) + \sum_{t=1}^T R_t(\mathbf{u}_t) - \sum_{t=1}^T R_t(\mathbf{w}_t^*) \\
&= \underbrace{\mathbb{E}_{1:T}\left[\sum_{t=1}^T R_t(\mathbf{w}_t)\right] - \sum_{m=1}^M \sum_{t\in\mathcal{I}_m} R_t(\mathbf{w}^*_{\mathcal{I}_m})}_{\texttt{term (a)}} + \underbrace{\sum_{m=1}^M \sum_{t\in\mathcal{I}_m} R_t(\mathbf{w}^*_{\mathcal{I}_m}) - \sum_{t=1}^T R_t(\mathbf{w}_t^*)}_{\texttt{term (b)}},
\end{aligned}
$$

where $M = \lceil \frac{T}{\Delta} \rceil \leq T/\Delta + 1$ is the number of the intervals. Then, we turn to analyze term (a) and term (b), respectively.

Analysis for term (a). Term (a) measures the regret of our algorithm when compared with a piece-wise stationary sequence. We first show that the regret defined over the expected risk $R_t(\cdot)$ can be related to the empirical risk estimator $\widehat{R}_t(\cdot)$ due to its unbiased property.

$$
\begin{aligned}
&\texttt{term (a)} \qquad (18)\\
&= \mathbb{E}_{1:T}\left[\sum_{t=1}^T R_t(\mathbf{w}_t)\right] - \sum_{t=1}^T R_t(\mathbf{u}_t) \\
&\leq \mathbb{E}_{1:T}\left[\sum_{t=1}^T \langle \nabla R_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{u}_t\rangle\right] \\
&= \underbrace{\mathbb{E}_{1:T}\left[\sum_{t=1}^T \langle \nabla R_t(\mathbf{w}_t) - \nabla \widehat{R}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{u}_t\rangle\right]}_{\texttt{term (a-1)}} + \underbrace{\mathbb{E}_{1:T}\left[\sum_{t=1}^T [\langle \nabla \widehat{R}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{u}_t\rangle\right]}_{\texttt{term (a-2)}}, \qquad (19)
\end{aligned}
$$

where the first inequality is due to the convexity of the risk function $R_t(\cdot)$. For term (a-1), we have

$$
\begin{aligned}
\texttt{term (a-1)} &= \mathbb{E}_{1:T}\left[\langle \nabla R_t(\mathbf{w}_t) - \nabla \widehat{R}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{u}_t\rangle\right] \\
&= \mathbb{E}_{1:t-1}\left[\langle \nabla R_t(\mathbf{w}_t) - \mathbb{E}_t\left[\nabla \widehat{R}_t(\mathbf{w}_t) \,\middle|\, 1:t-1\right], \mathbf{w}_t - \mathbf{u}_t\rangle\right] = 0, \qquad (20)
\end{aligned}
$$

where the last equality is due to the unbiasedness of the risk estimator $\widehat{R}_t$ such that $\nabla R_t(\mathbf{w}_t) = \mathbb{E}_t\left[\nabla \widehat{R}_t(\mathbf{w}_t) \,\middle|\, 1:t-1\right]$. Thus, it is sufficient to analyze term (a-2) to provide an upper bound for term (a). For the model sequence $\{\mathbf{w}_t\}_{t=1}^T$ generated by the UOGD algorithm in Eq. (4), we have the following lemma, whose proof is deferred to Appendix E.3.

Lemma 6. *Under same assumptions of Theorem 1, UOGD in Eq. (4) with a step size $\eta > 0$ satisfies*

$$\sum_{t=1}^{T} \langle \nabla \widehat{R}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{u}_t \rangle \leq \frac{2\eta K G^2 T}{\sigma^2} + \frac{2\Gamma P_T + \Gamma^2}{2\eta}$$

for any comparator sequence $\{\mathbf{u}_t\}_{t=1}^T$ with $\mathbf{u}_t \in \mathcal{W}$. Moreover, $P_T = \sum_{t=2}^T \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$ measures the variation of the comparator sequence.

Since the comparator sequence in term (a) only changes $M-1$ times, its variation is bounded by $P_T \leq \Gamma(M-1) \leq (\Gamma T)/\Delta$. By Lemma 6 and taking an expectation over both sides, we have

$$\texttt{term (a-2)} \leq \frac{2\eta K G^2 T}{\sigma^2} + \frac{2\Gamma^2 T/\Delta + \Gamma^2}{2\eta}.$$

Combining the upper bounds of term(a-1) and term (a-2) yields

$$\texttt{term (a)} \leq \texttt{term (a-1)} + \texttt{term (a-2)} \leq \frac{2\eta K G^2 T}{\sigma^2} + \frac{2\Gamma^2 T/\Delta + \Gamma^2}{2\eta}. \tag{21}$$

Analysis for term (b). For term (b), we can use a similar argument in [17] to show that

$$\begin{aligned}
\texttt{term (b)} &= \sum_{m=1}^{M} \sum_{t \in \mathcal{I}_m} \left(R_t(\mathbf{w}^*_{\mathcal{I}_m}) - R_t(\mathbf{w}^*_t)\right) \\
&\leq \sum_{m=1}^{M} \sum_{t \in \mathcal{I}_m} \left(R_t(\mathbf{w}^*_{s_m}) - R_t(\mathbf{w}^*_t)\right) \\
&= \sum_{m=1}^{M} \sum_{t \in \mathcal{I}_m} \left(R_t(\mathbf{w}^*_{s_m}) - R_{s_m}(\mathbf{w}^*_{s_m}) + R_{s_m}(\mathbf{w}^*_{s_m}) - R_t(\mathbf{w}^*_t)\right) \\
&\leq \sum_{m=1}^{M} \sum_{t \in \mathcal{I}_m} \left(R_t(\mathbf{w}^*_{s_m}) - R_{s_m}(\mathbf{w}^*_{s_m}) + R_{s_m}(\mathbf{w}^*_t) - R_t(\mathbf{w}^*_t)\right) \\
&\leq 2\Delta \sum_{m=1}^{M} \sum_{t \in \mathcal{I}_m} \sup_{w \in \mathcal{W}} |R_t(\mathbf{w}) - R_{t-1}(\mathbf{w})| \\
&= 2\Delta \sum_{t=1}^{T} \sup_{w \in \mathcal{W}} |R_t(\mathbf{w}) - R_{t-1}(\mathbf{w})|,
\end{aligned}$$

where $s_m = (m-1)\Delta + 1$ is the first time step at interval $\mathcal{I}_m$. In above, the first inequality is due to the optimality of $\mathbf{w}^*_{\mathcal{I}_m}$ over the interval $\mathcal{I}_m$. The second inequality holds since $\mathbf{w}^*_{s_m} = \arg\min_{\mathbf{w} \in \mathcal{W}} R_{s_m}(\mathbf{w})$. According to the label shift condition, we can further bound the variation of the loss function by the variation of the class prior $V_T = \sum_{t=2}^T \|\boldsymbol{\mu}_{y_t} - \boldsymbol{\mu}_{y_{t-1}}\|_1$:

$$\begin{aligned}
\sup_{\mathbf{w} \in \mathcal{W}} |R_t(\mathbf{w}) - R_{t-1}(\mathbf{w})| &= \sup_{\mathbf{w} \in \mathcal{W}} \left| \sum_{k=1}^{K} ([\boldsymbol{\mu}_{y_t}]_k - [\boldsymbol{\mu}_{y_{t-1}}]_k) R_0^k(\mathbf{w}) \right| \\
&\leq B \sum_{k=1}^{K} \left| [\boldsymbol{\mu}_{y_t}]_k - [\boldsymbol{\mu}_{y_{t-1}}]_k \right| \\
&= B \|\boldsymbol{\mu}_{y_t} - \boldsymbol{\mu}_{y_{t-1}}\|_1 = B V_T.
\end{aligned}$$

Thus, we have

$$\texttt{term (b)} \leq 2B\Delta V_T. \tag{22}$$

Combining term (a) and term (b), we have

$$\begin{aligned}
\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_t)\right] - \sum_{t=1}^{T} R_t(\mathbf{w}_t^*) &\leq \frac{2\eta K G^2 T}{\sigma^2} + \frac{2\Gamma^2 T/\Delta + \Gamma^2}{2\eta} + 2B\Delta V_T \\
&\leq \frac{2\eta K G^2 T}{\sigma^2} + \frac{\Gamma^2}{2\eta} + 4\Gamma\sqrt{\frac{BTV_T}{\eta}} + 2BV_T \\
&\leq \frac{2\eta K G^2 T}{\sigma^2} + \frac{\Gamma^2}{2\eta} + 4\Gamma\sqrt{\frac{BTV_T}{\eta}} + 4B\sqrt[3]{T^2 V_T} \\
&\leq \big(\frac{2KG^2}{\sigma^2} + 2B^2\big)\eta T + \frac{\Gamma^2}{\eta} + 4(\Gamma+1)\sqrt{\frac{BTV_T}{\eta}},
\end{aligned}$$

where the second inequality is to the parameter setting $\Delta = \lceil \sqrt{\Gamma^2 T/(\eta B V_T)} \rceil$. The third inequality is due to the fact that $V_T \leq 2T$. The last inequality holds by the AM-GM inequality $3B\sqrt[3]{T^2 V_T} \leq 2\sqrt{\frac{BTV_T}{\eta}} + B^2\eta T$. Hence, we complete the proof. □

E.2 Proof of Theorem 2

Proof of Theorem 2. We can decompose the dynamic regret into two parts with the piece-wise stationary sequence $\{\mathbf{u}_t\}_{t=1}^T$ introduced in the proof of Theorem 1.

$$\begin{aligned}
&\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_t)\right] - \sum_{t=1}^{T} R_t(\mathbf{w}_t^*) \\
=& \mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_t)\right] - \sum_{t=1}^{T} R_t(\mathbf{u}_t) + \sum_{t=1}^{T} R_t(\mathbf{u}_t) - \sum_{t=1}^{T} R_t(\mathbf{w}_t^*) \\
=& \underbrace{\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_t)\right] - \sum_{m=1}^{M}\sum_{t\in\mathcal{I}_m} R_t(\mathbf{w}_{\mathcal{I}_m}^*)}_{\texttt{term (a)}} + \underbrace{\sum_{m=1}^{M}\sum_{t\in\mathcal{I}_m} R_t(\mathbf{w}_{\mathcal{I}_m}^*) - \sum_{t=1}^{T} R_t(\mathbf{w}_t^*)}_{\texttt{term (b)}},
\end{aligned}$$

where term (b) can be bounded by

$$\texttt{term (b)} \leq 2B\Delta V_T$$

following the same arguments in deriving (22). So, we only need to focus on the analysis of term (a), the regret of the model sequence returned by ATLAS algorithm to the piece-wise stationary compactors.

For any i-th base UOGD algorithm, we can further decompose term (a) into the meta-regret and the base-regret

$$\begin{aligned}
&\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_t)\right] - \sum_{t=1}^{T} R_t(\mathbf{u}_t) \\
=& \underbrace{\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_t) - \sum_{t=1}^{T} R_t(\mathbf{w}_{t,i})\right]}_{\texttt{meta-regret}} + \underbrace{\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_{t,i})\right] - \sum_{t=1}^{T} R_t(\mathbf{u}_t)}_{\texttt{base-regret}},
\end{aligned} \tag{23}$$

where the meta-regret measures the performance gap between the model returned by ATLAS algorithm and that by the i-th base UOGD algorithm, and the base-regret is the regret of the i-th base UOGD algorithm running with step size $\eta_i > 0$.

Analysis for meta-regret. For the meta-regret, we have

$$
\begin{aligned}
&\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_t) - \sum_{t=1}^{T} R_t(\mathbf{w}_{t,i})\right] \\
&\leq \mathbb{E}_{1:T}\left[\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j} R_t(\mathbf{w}_{t,j}) - \sum_{t=1}^{T} R_t(\mathbf{w}_{t,i})\right] \\
&= \mathbb{E}_{1:T}\left[\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j} \widehat{R}_t(\mathbf{w}_{t,j}) - \sum_{t=1}^{T} \widehat{R}_t(\mathbf{w}_{t,i})\right] \\
&\quad + \mathbb{E}_{1:T}\left[\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j}(R_t(\mathbf{w}_{t,j}) - \widehat{R}_t(\mathbf{w}_{t,j})) + \sum_{t=1}^{T}(R_t(\mathbf{w}_{t,i}) - \widehat{R}_t(\mathbf{w}_{t,i}))\right] \\
&= \mathbb{E}_{1:T}\left[\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j} \widehat{R}_t(\mathbf{w}_{t,j}) - \sum_{t=1}^{T} \widehat{R}_t(\mathbf{w}_{t,i})\right], \qquad (24)
\end{aligned}
$$

where the first inequality is due to the Jensen's inequality and the last equality is due to the unbiasedness of the risk estimator such that $\mathbb{E}_t[\widehat{R}_t(\mathbf{w}_{t,i})] = R_t(\mathbf{w}_{t,i})$ for any $i \in [N]$. Then, we can upper bound the meta-regret by the following lemma, whose proof is deferred to Appendix E.3.

Lemma 7. *By setting the learning rate $\varepsilon = \frac{\sigma}{B}\sqrt{\frac{\ln N+2}{KT}}$, the meta-algorithm of* ATLAS *(Algorithm 2) satisfies*

$$
\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j} \widehat{R}_t(\mathbf{w}_{t,j}) - \sum_{t=1}^{T} \widehat{R}_t(\mathbf{w}_{t,i}) \leq \frac{2B}{\sigma}\sqrt{(\ln N + 2)KT}
$$

for any $i \in [N]$, where $B \triangleq \sup_{(\mathbf{x},y)\in\mathcal{X}\times\mathcal{Y},\mathbf{w}\in\mathcal{W}} |\ell(f(\mathbf{w},\mathbf{x}),y)|$ is defined as the upper bound of the loss function.

As a consequence of Lemma 7, we can upper bound the meta-regret as

$$
\texttt{meta-regret} \leq \frac{2B}{\sigma}\sqrt{(\ln N + 2)KT}.
$$

Analysis for base-regret. Since the base-algorithm of ATLAS algorithm (Algorithm 1) is taken as the UOGD algorithm, following a similar argument for obtaining Lemma 6, we have

$$
\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_{t,i})\right] - \sum_{t=1}^{T} R_t(\mathbf{u}_t) \leq \frac{2\eta_i K G^2 T}{\sigma^2} + \frac{2\Gamma P_T + \Gamma^2}{2\eta_i}
$$

for any base-algorithm with the index $i \in [N]$, where P_T is the variation of the comparator sequence $P_T = \sum_{t=2}^{T}\|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$. Then, by combining the meta-regret and the base-regret, we can obtain that

$$
\begin{aligned}
\texttt{term (a)} &= \mathbb{E}_{1:T}\left[\sum_{t=1}^{T}(\mathbf{w}_t)\right] - \sum_{t=1}^{T} R_t(\mathbf{u}_t) \\
&\leq \frac{2B}{\sigma}\sqrt{(\ln N + 2)KT} + \frac{2\eta_i K G^2 T}{\sigma^2} + \frac{2\Gamma P_T + \Gamma^2}{2\eta_i}
\end{aligned}
$$

for any base-algorithm $i \in [N]$.

It remains to identify the optimal step size η_{i_*} to make the bound tight. Due to the construction of the step size pool

$$
\mathcal{H} = \left\{\frac{\Gamma\sigma}{2G\sqrt{KT}} \cdot 2^{i-1} \mid i \in [N]\right\}
$$

with $N = 1 + \lceil \frac{1}{2}\log_2(1+2T) \rceil$, we can make sure that the optimal step size $\eta^* = \frac{\sigma}{2G}\sqrt{\frac{\Gamma^2+2\Gamma P_T}{KT}}$ is covered by the step size pool. Even better, due to the logarithmic construction of the pool, we claim that there must exist an $i_* \in [N]$ satisfies that $\eta_{i_*}/2 \leq \eta^* \leq \eta_{i_*}$. So, we further bound term (a) as

$$\texttt{term (a)} \leq \frac{2B}{\sigma}\sqrt{(\ln N + 2)KT} + \frac{3G}{\sigma}\sqrt{KT(2\Gamma P_T + \Gamma^2)}$$

by comparing with the i_*-th base-algorithm in the decomposition (23).

Overall dynamic regret bound. Combining term (a) and term (b) and noticing that $\{\mathbf{u}_t\}_{t=1}^T$ is the piecewise stationary comparator sequences with $P_T = \sum_{t=2}^T \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_{t=1}^T \leq \Gamma T/\Delta$, we can bound the overall dynamic regret as

$$\begin{aligned}
\mathbf{Reg}_T^{\mathbf{d}} &= \mathbb{E}_{1:T}\left[\sum_{t=1}^T R_t(\mathbf{w}_t)\right] - \sum_{t=1}^T R_t(\mathbf{w}_t^*) \\
&\leq 2B\Delta V_T + \frac{3G\Gamma}{\sigma}\sqrt{KT\left(1+\frac{2T}{\Delta}\right)} + \frac{2B}{\sigma}\sqrt{(\ln N+2)KT} \\
&\leq 2B\Delta V_T + \frac{3G\Gamma T}{\sigma}\sqrt{\frac{2K}{\Delta}} + \frac{3G\Gamma}{\sigma}\sqrt{KT} + \frac{2B}{\sigma}\sqrt{(\ln N+2)KT},
\end{aligned}$$

where the last inequality is due to the fact that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for any $a, b \geq 0$. We can complete the proof by choosing $\Delta = \lceil (\frac{3GT\Gamma\sqrt{2K}}{4\sigma B V_T})^{\frac{2}{3}} \rceil$ and obtain the bound as

$$\begin{aligned}
\mathbf{Reg}_T^{\mathbf{d}} &\leq 3\left(\frac{9KBG^2\Gamma^2}{\sigma^2}\right)^{\frac{1}{3}} \cdot V_T^{1/3}T^{2/3} + \frac{3\sqrt{K}(G\sqrt{\Gamma} + B\sqrt{\ln N+2})}{\sigma}\sqrt{T} + 2BV_T \\
&\leq 3\left(\frac{9KBG^2\Gamma^2}{\sigma^2}\right)^{\frac{1}{3}} \cdot V_T^{1/3}T^{2/3} + \frac{3\sqrt{K}(G\sqrt{\Gamma} + B\sqrt{\ln N+2})}{\sigma}\sqrt{T} + 4BV_T^{1/3}T^{2/3},
\end{aligned}$$

where the last inequality is due to the fact that $V_T \leq 2T$. □

E.3 Useful Lemmas in the Analysis of Theorem 1 and Theorem 2

This section presents several useful lemmas that are omitted in the proof of Theorem 1 and Theorem 2.

Lemma 6. *Under same assumptions of Theorem 1, UOGD in Eq. (4) with a step size $\eta > 0$ satisfies*

$$\sum_{t=1}^T \langle \nabla \widehat{R}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{u}_t \rangle \leq \frac{2\eta K G^2 T}{\sigma^2} + \frac{2\Gamma P_T + \Gamma^2}{2\eta}$$

for any comparator sequence $\{\mathbf{u}_t\}_{t=1}^T$ with $\mathbf{u}_t \in \mathcal{W}$. Moreover, $P_T = \sum_{t=2}^T \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$ measures the variation of the comparator sequence.

Proof of Lemma 6. We can decompose the L.H.S. of the inequality in Lemma 6 as

$$\langle \nabla \widehat{R}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{u}_t \rangle = \underbrace{\langle \nabla \widehat{R}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_{t+1} \rangle}_{\texttt{term (a)}} + \underbrace{\langle \nabla \widehat{R}_t(\mathbf{w}_t), \mathbf{w}_{t+1} - \mathbf{u}_t \rangle}_{\texttt{term (b)}}.$$

For term (a), we have

$$\begin{aligned}
\texttt{term (a)} &= \langle \nabla \widehat{R}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_{t+1} \rangle \\
&\leq \|\nabla \widehat{R}_t(\mathbf{w}_t)\|_2 \|\mathbf{w}_t - \mathbf{w}_{t+1}\|_2 \\
&\leq 2\eta \|\nabla \widehat{R}_t(\mathbf{w}_t)\|_2^2 + \frac{1}{2\eta}\|\mathbf{w}_t - \mathbf{w}_{t+1}\|_2^2,
\end{aligned}$$

where the first inequality is due to the Cauchy-Schwarz inequality and the second inequality comes from the AM-GM inequality. Then, we bound term (b) by [19, Lemma 7],

$$\texttt{term (b)} \leq \frac{1}{2\eta}\left(\|\mathbf{w}_t - \mathbf{u}_t\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{u}_t\|_2^2 - \|\mathbf{w}_t - \mathbf{w}_{t+1}\|_2^2\right).$$

Combining the two terms and taking the summation over T iterations, we have

$$
\begin{aligned}
&\sum_{t=1}^{T}\langle \nabla \widehat{R}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{u}_t\rangle \\
&\leq 2\eta \sum_{t=1}^{T}\|\nabla \widehat{R}_t(\mathbf{w}_t)\|_2^2 + \frac{1}{2\eta}\sum_{t=1}^{T}\left(\|\mathbf{w}_t - \mathbf{u}_t\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{u}_t\|_2^2\right) \\
&\leq 2\eta \sum_{t=1}^{T}\|\nabla \widehat{R}_t(\mathbf{w}_t)\|_2^2 + \frac{1}{2\eta}\sum_{t=2}^{T}\left(\|\mathbf{w}_t - \mathbf{u}_t\|_2^2 - \|\mathbf{w}_t - \mathbf{u}_{t-1}\|_2^2\right) + \frac{\Gamma^2}{2\eta} \\
&\leq 2\eta \sum_{t=1}^{T}\|\nabla \widehat{R}_t(\mathbf{w}_t)\|_2^2 + \frac{1}{2\eta}\sum_{t=2}^{T}(\|\mathbf{w}_t - \mathbf{u}_t\|_2 + \|\mathbf{w}_t - \mathbf{u}_{t-1}\|_2)\cdot\|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2 + \frac{\Gamma^2}{2\eta} \\
&\leq 2\eta \sum_{t=1}^{T}\|\nabla \widehat{R}_t(\mathbf{w}_t)\|_2^2 + \frac{\Gamma}{\eta}\sum_{t=2}^{T}\|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2 + \frac{\Gamma^2}{2\eta} \\
&= 2\eta \sum_{t=1}^{T}\|\nabla \widehat{R}_t(\mathbf{w}_t)\|_2^2 + \frac{2\Gamma P_T + \Gamma^2}{2\eta}, \qquad (25)
\end{aligned}
$$

where we denote by $P_t = \sum_{t=2}^{T}\|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$ the variation of the comparator sequence $\{\mathbf{u}_t\}_{t=1}^{T}$. The third inequality is due to the fact $\|a\|_2^2 - \|b\|_2^2 \leq (\|a\|_2 + \|b\|_2)\cdot\|a-b\|_2$ for any $a, b \in \mathbb{R}^d$.

We can further bound the first term of (25) by

$$
\begin{aligned}
2\eta\sum_{t=1}^{T}\|\nabla \widehat{R}_t(\mathbf{w}_t)\|_2^2 &\overset{(3)}{=} 2\eta\sum_{t=1}^{T}\left\|\sum_{k=1}^{K}[C_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}]_k\cdot\nabla R_0^k(\mathbf{w}_t)\right\|_2^2 \\
&\leq 2\eta\sum_{t=1}^{T}\left(\sum_{k=1}^{K}\left|[C_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}]_k\right|\|\nabla R_0^k(\mathbf{w}_t)\|_2\right)^2 \\
&\leq 2\eta G^2\sum_{t=1}^{T}\left\|C_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\right\|_1^2 \\
&\leq 2\eta K G^2\sum_{t=1}^{T}\left\|C_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\right\|_2^2 \\
&\leq 2\eta K G^2\sum_{t=1}^{T}\left\|C_{f_0}^{-1}\right\|_2^2\cdot\|\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\|_2^2 \\
&\leq 2\eta K G^2\sum_{t=1}^{T}\|C_{f_0}^{-1}\|_2^2 \leq \frac{2\eta K G^2 T}{\sigma^2}, \qquad (26)
\end{aligned}
$$

where σ is the minimum singular value of the matrix C_{f_0}. In above, the second inequality is due to the definition $R_0^k(\mathbf{w}) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_0(\mathbf{x}\mid y=k)}[\ell(f(\mathbf{w},\mathbf{x}),k)]$ and the norm of gradient of the loss function $\|\nabla_{\mathbf{w}}\ell(f(\mathbf{w},\mathbf{x}),y)\|_2$ is bounded by G for any $\mathbf{x}\in\mathcal{X}$ and $y\in[K]$. The third inequality is a consequence of the fact that $\|\mathbf{x}\|_1 \leq \sqrt{\mathbf{x}_2}$ for any $\mathbf{x}\in\mathbb{R}^K$. The last second inequality hold since $\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\in\Delta_{K-1}$ and then we have $\|\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\|_2^2 \leq 1$. We complete the proof by plugging (26) into (25). □

Then, we provide the proof of Lemma 7, which provide an upper bound for the meta-algorithm of the ATLAS algorithm.

Lemma 7. *By setting the learning rate $\varepsilon = \frac{\sigma}{B}\sqrt{\frac{\ln N+2}{KT}}$, the meta-algorithm of* ATLAS *(Algorithm 2) satisfies*

$$
\sum_{t=1}^{T}\sum_{j=1}^{N}p_{t,j}\widehat{R}_t(\mathbf{w}_{t,j}) - \sum_{t=1}^{T}\widehat{R}_t(\mathbf{w}_{t,i}) \leq \frac{2B}{\sigma}\sqrt{(\ln N+2)KT}
$$

for any $i \in [N]$, where $B \triangleq \sup_{(\mathbf{x},y)\in\mathcal{X}\times\mathcal{Y},\mathbf{w}\in\mathcal{W}} |\ell(f(\mathbf{w},\mathbf{x}),y)|$ is defined as the upper bound of the loss function.

Proof. Since ATLAS takes the Hedge algorithm as the meta-algorithm, we can exploit the standard analysis to upper bound the meta-regret. According to the regret guarantee of the Hedge algorithm (Lemma 11 in Appendix G) and setting $\mathbf{m}_t = 0$, we can bound the meta-regret bound by

$$\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j}\widehat{R}_t(\mathbf{w}_{t,j}) - \sum_{t=1}^{T}\widehat{R}_t(\mathbf{w}_{t,i}) \leq \frac{\ln N + 2}{\varepsilon} + \varepsilon\sum_{t=1}^{T}|\max_{i\in[N]}\{\widehat{R}(\mathbf{w}_{t,i})\}|^2, \tag{27}$$

where the last term can be further bounded by

$$\begin{aligned}
|\widehat{R}_t(\mathbf{w}_{t,i})| &= \left|\sum_{k=1}^{K}[C_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}]_k \cdot R_0^k(\mathbf{w}_{t,i})\right| \\
&\leq B\|C_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\|_1 \leq B\sqrt{K}\|C_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\|_2 \\
&\leq B\sqrt{K}\|C_{f_0}^{-1}\|_2 \cdot \|\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\|_2 \leq \frac{B\sqrt{K}}{\sigma},
\end{aligned} \tag{28}$$

where the third inequality is due to the Cauchy–Schwarz inequality. The last inequality comes from the fact that $\|C_{f_0}^{-1}\|_2 \leq \sigma^{-1}$ and $\|\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\|_2 \leq 1$. Plugging (28) into (27) yields

$$\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j}\widehat{R}_t(\mathbf{w}_{t,j}) - \sum_{t=1}^{T}\widehat{R}_t(\mathbf{w}_{t,i}) \leq \frac{\ln N + 2}{\varepsilon} + \varepsilon\sum_{t=1}^{T}\frac{B^2KT}{\sigma^2}.$$

We complete the proof by setting $\varepsilon = \frac{\sigma}{B}\sqrt{\frac{(\ln N+2)}{KT}}$. □

F Omitted Details for Section 3.3

In this section, we first introduce the omitted algorithm details for the ATLAS-ADA algorithm in Appendix F.1. Then, we present the proof of the Theorem 3 (Appendix F.2), followed by the useful lemmas in the regret analysis.

F.1 Algorithm Details for ATLAS-ADA

This part provides algorithm details of the ATLAS-ADA. The main procedures are presented in Algorithm 4 (base-algorithm) and Algorithm 5 (meta-algorithm).

Hint Functions. As shown in Section 3.3, a guidance for designing the hint function is to approximate the true priors $\boldsymbol{\mu}_{y_t}$ with hint priors $\boldsymbol{h}_{y_t}$. A direct way is to use current data S_t to estimate the hint priors, which we call *Forward Hint*. However, when the sample size $|S_t|$ is small at each round, the estimated priors can be far from the true priors, which motivates us to reuse the historical information. The intuition is that we can benefit from the historical data if there are some regular shift patterns. To see this, we design three additional hint functions. Details are listed in the following.

- **Forward Hint**. We perform transductive learning with the current data S_t by $\boldsymbol{h}_{y_t} = \widehat{\boldsymbol{\mu}}_{y_t}$.
- **Window Hint**. With the belief that recent data are useful, we set a sliding window size of L_m and use the average of the estimated priors in the window as the hint priors:

$$\boldsymbol{h}_{y_t} = \frac{1}{L_m}\sum_{s=t-L_m}^{t-1}\widehat{\boldsymbol{\mu}}_{y_s}.$$

- **Periodic Hint**. For a periodic situation where a similar pattern recurs at a specific time interval, we consider reusing the historical data in a periodic manner by maintaining a buffer $\{\bar{\boldsymbol{\mu}}_i \mid i \in [L_p]\}$ with the size of L_p. At round t, we utilize the estimated prior $\widehat{\boldsymbol{\mu}}_{y_t}$ to update the priors in the buffer:

$$\bar{\boldsymbol{\mu}}_i^{(t)} = \begin{cases} \bar{\boldsymbol{\mu}}_i^{(t-1)} + \lambda_t\left(\widehat{\boldsymbol{\mu}}_{y_t} - \bar{\boldsymbol{\mu}}_i^{(t-1)}\right) & i = t \bmod L_p \\ \bar{\boldsymbol{\mu}}_i^{(t-1)} & \text{otherwise} \end{cases},$$

Algorithm 4 ATLAS-ADA: base-algorithm

Input: step size $\eta_i \in \mathcal{H}$

Let $\mathbf{w}_{1,i}$ be any point in $\mathcal{W}$

for $t = 2$ **to** T **do**

  construct the risk estimator $\widehat{R}_{t-1}$ as (3)

  update the model of base-learner by

$$\widehat{\mathbf{w}}_{t,i} = \Pi_{\mathcal{W}}[\widehat{\mathbf{w}}_{t-1,i} - \eta_i \nabla \widehat{R}_{t-1}(\mathbf{w}_{t-1,i})]$$

$$\mathbf{w}_{t,i} = \arg\min_{\mathbf{w}\in\mathcal{W}} H_t(\mathbf{w}) + \frac{1}{2\eta_i}\|\mathbf{w} - \widehat{\mathbf{w}}_{t,i}\|_2^2.$$

  send $\mathbf{w}_{t,i}$ to the meta-algorithm

end for

Algorithm 5 ATLAS-ADA: meta-algorithm

Input: step size pool $\mathcal{H}$; learning rate ε

initialization: $\forall i \in [N], p_{1,i} = 1/N$

for $t = 2$ **to** T **do**

  receive $\{\mathbf{w}_{t,i}\}_{i=1}^N$ from base-learners

  update weight $\mathbf{p}_t \in \Delta_N$ according to

$$p_{t,i} \propto \exp\left(-\varepsilon\left(\sum_{s=1}^{t-1} \widehat{R}_s(\mathbf{w}_{s,i}) + H_t(\mathbf{w}_{t,i})\right)\right)$$

  predict final output $\mathbf{w}_t = \sum_{i=1}^N p_{t,i}\mathbf{w}_{t,i}$

end for

where $i \in [L_p]$ and $\bar{\boldsymbol{\mu}}_i^{(t)}$ denotes the value of $\bar{\boldsymbol{\mu}}_i$ at round t. $\lambda_t \in [0,1]$ is a coefficient to balance the historical and present information, which can be set to be $1/t$ or a constant value. Then we set $\boldsymbol{h}_{y_t} = \bar{\boldsymbol{\mu}}_{i_t}^{(t)}$, where $i_t = t \bmod L_p$.

- **Online KMeans Hint**. To further use the feature similarity of the historical data, we design a hint function with the online k-means clustering algorithms [73]. We maintain L_k prototypes $\{(\bar{\mathbf{x}}_i, \bar{\boldsymbol{\mu}}_i) \mid i \in [L_k]\}$. At round t, we find the prototype i_t, which is the most similar to the current unlabeled data S_t:

$$i_t = \arg\min_{i\in[L_k]} \frac{1}{|S_t|}\sum_{\mathbf{x}\in S_t} \|\bar{\mathbf{x}}_i - \mathbf{x}\|^2,$$

and then we update the corresponding prototype by

$$\bar{\mathbf{x}}_{i_t}^{(t)} = \bar{\mathbf{x}}_{i_t}^{(t-1)} + \kappa_t\left[\left(\frac{1}{|S_t|}\sum_{\mathbf{x}\in S_t}\mathbf{x}\right) - \bar{\mathbf{x}}_{i_t}^{(t-1)}\right],$$

$$\bar{\boldsymbol{\mu}}_{i_t}^{(t)} = \bar{\boldsymbol{\mu}}_{i_t}^{(t-1)} + \kappa_t\left(\widehat{\boldsymbol{\mu}}_{y_t} - \bar{\boldsymbol{\mu}}_{i_t}^{(t-1)}\right),$$

where $\left(\bar{\mathbf{x}}_{i_t}^{(t)}, \bar{\boldsymbol{\mu}}_{i_t}^{(t)}\right)$ represents the value of i_t-th prototype at round t. $\kappa_t \in [0,1]$ is a coefficient to balance the historical and present information, which can be set to be $1/t$ or a constant value. Then we set $\boldsymbol{h}_{y_t} = \bar{\boldsymbol{\mu}}_{i_t}^{(t)}$.

F.2 Proof of Theorem 3

The proof of Theorem 3 shares a similar spirit with that of Theorem 2, where we decompose the overall dynamic regret into meta-regret and base-regret. The new challenge is that the ATLAS-ADA algorithm involves the hint function $H_t(\cdot)$ to exploit the structure of distribution change, which requires a more delicate analysis for both the meta- and base-regret.

Proof of Theorem 3. Similar to the proof of Theorem 1 and Theorem 2, we can decompose the dynamic regret into two parts with the piece-wise stationary sequence $\{\mathbf{u}_t\}_{t=1}^T$,

$$\begin{aligned}
&\mathbb{E}_{1:T}\left[\sum_{t=1}^T R_t(\mathbf{w}_t)\right] - \sum_{t=1}^T R_t(\mathbf{w}_t^*) \\
=\ &\underbrace{\mathbb{E}_{1:T}\left[\sum_{t=1}^T R_t(\mathbf{w}_t)\right] - \sum_{m=1}^M\sum_{t\in\mathcal{I}_m} R_t(\mathbf{w}_{\mathcal{I}_m}^*)}_{\texttt{term (a)}} + \underbrace{\sum_{m=1}^M\sum_{t\in\mathcal{I}_m} R_t(\mathbf{w}_{\mathcal{I}_m}^*) - \sum_{t=1}^T R_t(\mathbf{w}_t^*)}_{\texttt{term (b)}}.
\end{aligned}$$

According to (22), we have `term (b)` $\leq 2B\Delta V_T$. Besides, when the parameter is set as $\Delta = 1$, we have `term (b)` $= 0$. Thus, the term (b) can be bounded as

$$\texttt{term (b)} \leq \mathbb{1}\{\Delta > 1\}\cdot 2B\Delta V_T. \tag{29}$$

As a consequence, we can focus on the analysis of term (a), which can be further decomposed into the meta-regret and the based-regret.

$$\begin{aligned}\texttt{term (a)} &= \mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_t)\right] - \sum_{t=1}^{T} R_t(\mathbf{u}_t) \\ &= \underbrace{\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_t) - \sum_{t=1}^{T} R_t(\mathbf{w}_{t,i})\right]}_{\texttt{meta-regret}} + \underbrace{\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_{t,i})\right] - \sum_{t=1}^{T} R_t(\mathbf{u}_t)}_{\texttt{base-regret}},\end{aligned}$$

where meta-regret measures the performance gap between the model returned by ATLAS-ADA algorithm and that by the i-th base algorithm. The base-regret is the regret of the i-th base algorithm running with step size $\eta_i > 0$.

Analysis for the meta-regret. Since $\mathbf{w}_t$ is independent of the sample observed at iteration t, following the same arguments in obtaining (24), we can bound the meta-regret as

$$\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_t) - \sum_{t=1}^{T} R_t(\mathbf{w}_{t,i})\right] \leq \mathbb{E}_{1:T}\left[\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j}\widehat{R}_t(\mathbf{w}_{t,j}) - \sum_{t=1}^{T}\widehat{R}_t(\mathbf{w}_{t,i})\right].$$

Remark 5. The main difference between the analysis of the meta-algorithm for ATLAS-ADA and that of ATLAS is that the former method involves an update step with the hint function. To obtain a meta-regret bound that adapts the quality of the hint function, the previous work [19] crucially relies on the *smoothness* of the loss function, which is generally hard to be satisfied in the OLaS problem (e.g., the commonly used hinge loss is not smooth). A more delicate analysis shows that the smoothness assumption is no longer necessary to achieve an adaptive meta-regret bound. The result is stated as the following lemma, whose proof is deferred to Appendix F.3.

Lemma 8. *By setting the learning rate* $\varepsilon = \sqrt{\frac{\ln N+2}{1+\sum_{t=1}^{T}(\max_{i\in[N]}\{|\widetilde{R}_t(\mathbf{w}_{t,i})-\widetilde{H}_t(\mathbf{w}_{t,i})|\})^2}}$, *the meta-algorithm of* ATLAS-ADA *(Algorithm 5) satisfies*

$$\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j}\widehat{R}_t(\mathbf{w}_{t,j}) - \sum_{t=1}^{T}\widehat{R}_t(\mathbf{w}_{t,i}) \leq \Gamma\sqrt{(\ln N+2)\left(1+\sum_{t=1}^{T}\sup_{\mathbf{w}\in\mathcal{W}}\|\nabla\widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2^2\right)}$$

for any $i \in [N]$, *where* $\widetilde{R}_t(\mathbf{w}_{t,i}) = \widehat{R}_t(\mathbf{w}_{t,i}) - \widehat{R}_t(\mathbf{w}_{\texttt{ref}})$ *and* $\widetilde{H}_t(\mathbf{w}_{t,i}) = H_t(\mathbf{w}_{t,i}) - H_t(\mathbf{w}_{\texttt{ref}})$ *are the reference loss functions built upon an arbitrary reference point* $\mathbf{w}_{\texttt{ref}} \in \mathcal{W}$.

According to Lemma 8, we can bound the meta-regret by

$$\begin{aligned}&\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_t) - \sum_{t=1}^{T} R_t(\mathbf{w}_{t,i})\right] \\ &\leq \Gamma\mathbb{E}_{1:T}\left[\sqrt{(\ln N+2)\sum_{t=1}^{T}\sup_{\mathbf{w}\in\mathcal{W}}\|\nabla\widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2^2}\right] \\ &\leq \Gamma\sqrt{(\ln N+2)\left(1+\sum_{t=1}^{T}\mathbb{E}_t\left[\sup_{\mathbf{w}\in\mathcal{W}}\|\nabla\widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2^2\right)\right]},\end{aligned}$$

where the last inequality is due to the Jensen's inequality and the concavity of the square root function.

Analysis for the base-regret. Since the model $\mathbf{w}_t$ is independent of the draw of dataset S_t received at iteration t, following a similar argument in obtain (21), we can bound the base-regret as

$$\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_{t,i})\right] - \sum_{t=1}^{T} R_t(\mathbf{u}_t) \leq \mathbb{E}_{1:T}\left[\sum_{t=1}^{T}[\langle\nabla\widehat{R}_t(\mathbf{w}_{t,i}), \mathbf{w}_{t,i} - \mathbf{u}_t\rangle\right].$$

Different from the analysis for the base-algorithm of ATLAS, the base-algorithm of ATLAS-ADA exploit a hint function $H_t(\cdot)$ to better exploit the pattern of the distribution challenge. We have the following regret guarantees on the output $\mathbf{w}_{t,i}$ of the base-algorithm.

Lemma 9. *Suppose the hint function $H_t : \mathcal{W} \mapsto \mathbb{R}$ is convex. The base-algorithm of* ATLAS-ADA *(Algorithm 4) with a step size $\eta_{t,i} > 0$ satisfies*

$$\sum_{t=1}^{T} \langle \nabla \widehat{R}_t(\mathbf{w}_{t,i}), \mathbf{w}_{t,i} - \mathbf{u}_t \rangle \leq 2\eta_i \sum_{t=1}^{T} \sup_{\mathbf{w}\in\mathcal{W}} \|\nabla \widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2^2 + \frac{\Gamma^2 + 2\Gamma P_T}{2\eta_i}$$

for any comparator sequence $\{\mathbf{u}_t\}_{t=1}^T$ with $\mathbf{u}_t \in \mathcal{W}$. Moreover, $P_T = \sum_{t=2}^T \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$ measures the variation of the comparator sequence.

By Lemma 9 and taking the expectation over both sides, we can bound the base-regret as

$$\mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_{t,i})\right] - \sum_{t=1}^{T} R_t(\mathbf{u}_t) \leq 2\eta_i \sum_{t=1}^{T} \mathbb{E}_t\left[\sup_{\mathbf{w}\in\mathcal{W}} \|\nabla \widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2^2\right] + \frac{\Gamma^2 + 2\Gamma P_T}{2\eta_i}.$$

Then, combining the meta- and base-regret, we can obtain that

$$\texttt{term(a)} \leq 2\eta_i G_T + \frac{\Gamma^2 + 2\Gamma P_T}{2\eta_i} + \Gamma\sqrt{(\ln N + 2)(1 + G_T)}$$

for any base-algorithm $i \in [N]$, where $G_T = \sum_{t=1}^T \mathbb{E}_t\left[\sup_{\mathbf{w}\in\mathcal{W}} \|\nabla \widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2^2\right]$ measures the reusability of the history information.

It remains to tune the step size to make the bound tight. We can bound the term G_t by

$$\begin{aligned} G_T &= \sum_{t=1}^{T} \mathbb{E}_t\left[\sup_{\mathbf{w}\in\mathcal{W}} \|\nabla \widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2^2\right] \\ &\leq \sum_{t=1}^{T} \mathbb{E}_t\left[2\sup_{\mathbf{w}\in\mathcal{W}} \|\nabla \widehat{R}_t(\mathbf{w})\|_2^2 + 2\sup_{\mathbf{w}\in\mathcal{W}} \|\nabla H_t(\mathbf{w})\|_2^2\right] \leq \frac{4TG^2K}{\sigma^2}, \end{aligned}$$

where the last inequality is due to the fact that $\sup_{\mathbf{w}\in\mathcal{W}} \|\nabla \widehat{R}_t(\mathbf{w}_t)\|_2 \leq (G\sqrt{K})/\sigma$ (shown as (26)) and the assumption that $\sup_{\mathbf{w}\in\mathcal{W}} \|\nabla H_t(\mathbf{w})\|_2 \leq (G\sqrt{K})/\sigma$. In such a case, we can show that the optimal step size $\eta_* = \frac{1}{2}\sqrt{\frac{\Gamma^2+2\Gamma P_T}{1+G_T}}$ lies in the range $[\frac{1}{2}\sqrt{\frac{\Gamma^2\sigma^2}{\sigma^2+4TG^2K}}, \frac{1}{2}\sqrt{\Gamma^2 + 2\Gamma^2 T}]$. So, we can construct the following step size pool to cover the optimal step size

$$\mathcal{H} = \left\{\frac{\Gamma\sigma}{\sqrt{\sigma^2 + 4TG^2K}} \cdot 2^{i-1} \mid i \in [N]\right\}$$

with $N = 1 + \left\lceil \frac{1}{2}\log_2\left((1+2T)(1+4TG^2K/\sigma^2)\right)\right\rceil$. Even better, due to the logarithmic construction of the pool, we claim that there must exist an $i_* \in [N]$ satisfies that $\eta_{i_*}/2 \leq \eta_* \leq \eta_{i_*}$. So, we further bound term (a) as

$$\texttt{term(a)} \leq 3\sqrt{(1+G_T)(\Gamma^2 + 2\Gamma P_T)} + \Gamma\sqrt{(\ln N + 2)(1+G_T)}.$$

Overall dynamic regret bound. We can then use similar arguments in the proof of Theorem 2 to obtain the final dynamic regret bound. Combining term (a) and term (b) and noticing that $\{\mathbf{u}_t\}_{t=1}^T$ is the piecewise stationary comparator sequences with $P_T = \sum_{t=2}^T \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_{t=1}^T \leq \Gamma T/\Delta$, we can bound the overall dynamic regret as

$$\begin{aligned} &\mathbb{E}_{1:T}\left[\mathbf{Reg}_T^{\mathbf{d}}\right] = \mathbb{E}_{1:T}\left[\sum_{t=1}^{T} R_t(\mathbf{w}_t)\right] - \sum_{t=1}^{T} R_t(\mathbf{w}_t^*) \\ &\leq \mathbb{1}\{\Delta > 1\} \cdot 2B\Delta V_T + 3\sqrt{(1+G_T)(\Gamma^2 + 2\Gamma P_T)} + \Gamma\sqrt{(\ln N + 2)(1+G_T)} \\ &\leq \mathbb{1}\{\Delta > 1\} \cdot 2B\Delta V_T + 3\Gamma\sqrt{\frac{2(1+G_T)T}{\Delta}} + 3\Gamma\sqrt{1+G_T} + \Gamma\sqrt{(\ln N + 2)(1+G_T)}, \end{aligned}$$

where the last inequality is due to the fact that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for any $a, b \geq 0$. To obtain an upper bound of the above inequality, we consider the two following cases,

Case 1: $3\Gamma\sqrt{2(1+G_T)T} > 4BV_T$, in such a case, we can set $\Delta = \lceil \frac{3\Gamma\sqrt{2(1+G_T)T}}{4BV_T} \rceil$. Then, we can bound the dynamic regret by

$$\begin{aligned}\mathbb{E}_{1:T}\left[\mathbf{Reg}_T^{\mathbf{d}}\right] &\leq 3\left(9B\Gamma^2\right)^{\frac{1}{3}} \cdot V_T^{1/3} T^{1/3} (1+G_T)^{1/3} + (3+\sqrt{2+\ln N})\Gamma\sqrt{1+G_T} \\ &= \mathcal{O}(V_T^{1/3} G_T^{1/3} T^{1/3} + \sqrt{1+G_T}).\end{aligned}$$

Case 2: $3\Gamma\sqrt{2(1+G_T)T} \leq 4BV_T$, in such a case, we can set $\Delta = 1$. Then, we have

$$\begin{aligned}\mathbb{E}_{1:T}\left[\mathbf{Reg}_T^{\mathbf{d}}\right] &\leq 3\Gamma\sqrt{2(1+G_T)T} + 3\Gamma\sqrt{1+G_T} + \Gamma\sqrt{(\ln N+2)(1+G_T)} \\ &\leq \left(6\Gamma(1+G_T)T\right)^{1/3}\left(12\Gamma BV_T\right)^{1/3} + 3\Gamma\sqrt{1+G_T} + \Gamma\sqrt{(\ln N+2)(1+G_T)} \\ &= \mathcal{O}(V_T^{1/3} G_T^{1/3} T^{1/3} + \sqrt{1+G_T}),\end{aligned}$$

where the second inequality is due to the inequality that $3\Gamma\sqrt{2(1+G_T)T} \leq 4BV_T$. We complete the proof by combining the two cases. □

F.3 Useful Lemmas in Analysis of Theorem 3

Lemma 8. *By setting the learning rate* $\varepsilon = \sqrt{\frac{\ln N+2}{1+\sum_{t=1}^T (\max_{i\in[N]}\{|\widetilde{R}_t(\mathbf{w}_{t,i}) - \widetilde{H}_t(\mathbf{w}_{t,i})|\})^2}}$, *the meta-algorithm of* ATLAS-ADA *(Algorithm 5) satisfies*

$$\sum_{t=1}^T \sum_{j=1}^N p_{t,j}\widehat{R}_t(\mathbf{w}_{t,j}) - \sum_{t=1}^T \widehat{R}_t(\mathbf{w}_{t,i}) \leq 2\Gamma\sqrt{(\ln N+2)\left(1+\sum_{t=1}^T \sup_{\mathbf{w}\in\mathcal{W}} \|\nabla\widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2^2\right)}$$

for any $i \in [N]$, *where* $\widetilde{R}_t(\mathbf{w}_{t,i}) = \widehat{R}_t(\mathbf{w}_{t,i}) - \widehat{R}_t(\mathbf{w}_{\mathtt{ref}})$ *and* $\widetilde{H}_t(\mathbf{w}_{t,i}) = H_t(\mathbf{w}_{t,i}) - H_t(\mathbf{w}_{\mathtt{ref}})$ *are the reference loss functions built upon an arbitrary reference point* $\mathbf{w}_{\mathtt{ref}} \in \mathcal{W}$.

Proof of Lemma 8. The key challenge in the proof is that the meta-algorithm is updated with the loss function $\widehat{R}_t(\mathbf{w}_{t,i})$ and the hint function $H_t(\mathbf{w}_{t,i})$, while we desire a regret that can scale with the gradient to make it compatible with the base-regret bound. To this end, we introduce the reference loss $\widetilde{R}_t(\mathbf{w}_{t,i}) = \widehat{R}_t(\mathbf{w}_{t,i}) - \widehat{R}_t(\mathbf{w}_{\mathtt{ref}})$ and $\widetilde{H}_t(\mathbf{w}_{t,i}) = H_t(\mathbf{w}_{t,i}) - H_t(\mathbf{w}_{\mathtt{ref}})$. It is easy to verify that the update procedure (4) with the original loss function is identical to that with the reference loss:

$$p_{t,i} = \frac{\exp\left(-\varepsilon\left(\sum_{s=1}^{t-1}\widehat{R}_s(\mathbf{w}_{s,i}) + H_t(\mathbf{w}_{t,i})\right)\right)}{\sum_{i=1}^K \exp\left(-\varepsilon\left(\sum_{s=1}^{t-1}\widehat{R}_s(\mathbf{w}_{s,i}) + H_t(\mathbf{w}_{t,i})\right)\right)} = \frac{\exp\left(-\varepsilon\left(\sum_{s=1}^{t-1}\widetilde{R}_s(\mathbf{w}_{s,i}) + \widetilde{H}_t(\mathbf{w}_{t,i})\right)\right)}{\sum_{i=1}^K \exp\left(-\varepsilon\left(\sum_{s=1}^{t-1}\widetilde{R}_s(\mathbf{w}_{s,i}) + \widetilde{H}_t(\mathbf{w}_{t,i})\right)\right)}.$$

As a consequence, we can bound the meta-regret by

$$\begin{aligned}&\sum_{t=1}^T \sum_{j=1}^N p_{t,j}\widehat{R}_t(\mathbf{w}_{t,j}) - \sum_{t=1}^T \widehat{R}_t(\mathbf{w}_{t,i}) \\ &= \sum_{t=1}^T \sum_{j=1}^N p_{t,j}\widetilde{R}_t(\mathbf{w}_{t,j}) - \sum_{t=1}^T \widetilde{R}_t(\mathbf{w}_{t,i}) \\ &\leq \varepsilon\sum_{t=1}^T \left(\max_{i\in[N]}\{|\widetilde{R}_t(\mathbf{w}_{t,i}) - \widetilde{H}_t(\mathbf{w}_{t,i})|\}\right)^2 + \frac{\ln N+2}{\varepsilon} \\ &\leq 2\sqrt{\sum_{t=1}^T \left(\max_{i\in[N]}\{|\widetilde{R}_t(\mathbf{w}_{t,i}) - \widetilde{H}_t(\mathbf{w}_{t,i})|\}\right)^2 (\ln N+2)},\end{aligned} \tag{30}$$

where the first inequality is due to Lemma 11 and the second inequality is by the step size setting of $\varepsilon = \sqrt{(\ln N+2)/(1+\sum_{t=1}^T \max_{i\in[N]}\{|\widetilde{R}_t(\mathbf{w}_{t,i}) - \widetilde{H}_t(\mathbf{w}_{t,i})|\})}$.

Then, we show that the deviation between the loss function $\max_{i\in[N]}\{|\widetilde{R}_t(\mathbf{w}_{t,i}) - \widetilde{H}_t(\mathbf{w}_{t,i})|\}$ can be converted to the deviation between the gradient. Let $G_t(\mathbf{w}_{t,i}) = \widehat{R}_t(\mathbf{w}_{t,i}) - H_t(\mathbf{w}_{t,i})$, for any $i \in [N]$, we have

$$\begin{aligned}
&|\widetilde{R}_t(\mathbf{w}_{t,i}) - \widetilde{H}_t(\mathbf{w}_{t,i})| = |G_t(\mathbf{w}_{t,i}) - G_t(\mathbf{w}_{\texttt{ref}})| \\
&= |\langle \nabla G_t(\boldsymbol{\xi}_{t,i}), \mathbf{w}_{t,i}\rangle| \leq \Gamma \|\nabla \widehat{R}_t(\boldsymbol{\xi}_{t,i}) - \nabla H_t(\boldsymbol{\xi}_{t,i})\|_2 \leq \Gamma \sup_{\mathbf{w}\in\mathcal{W}} \|\nabla \widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2, \quad (31)
\end{aligned}$$

where the first equality is due to the mean value theorem and $\boldsymbol{\xi}_{t,i} = c_{t,i}\mathbf{w}_{t,i} + (1 - c_{t,i})\mathbf{w}_{\texttt{ref}}$ with a certain constant $c_{t,i} \in [0, 1]$. Plugging (31) into (30), we obtain that

$$\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\widehat{R}_t(\mathbf{w}_{t-1,i}) - \sum_{t=1}^{T}\widehat{R}_t(\mathbf{w}_{t-1,i}) \leq 2\Gamma\sqrt{(\ln N + 2)\sum_{t=1}^{T}\sup_{\mathbf{w}\in\mathcal{W}}\|\nabla \widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2^2},$$

which completes the proof. □

Then, we provide the proof of the regret of the base-algorithm for ATLAS-ADA.

Lemma 9. *Suppose the hint function $H_t : \mathcal{W} \mapsto \mathbb{R}$ is convex. The base-algorithm of* ATLAS-ADA *(Algorithm 4) with a step size $\eta_{t,i} > 0$ satisfies*

$$\sum_{t=1}^{T}\langle \nabla \widehat{R}_t(\mathbf{w}_{t,i}), \mathbf{w}_{t,i} - \mathbf{u}_t\rangle \leq 2\eta_i \sum_{t=1}^{T}\sup_{\mathbf{w}\in\mathcal{W}}\|\nabla \widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2^2 + \frac{\Gamma^2 + 2\Gamma P_T}{2\eta_i},$$

for any comparator sequence $\{\mathbf{u}_t\}_{t=1}^T$ with $\mathbf{u}_t \in \mathcal{W}$. Moreover, $P_T = \sum_{t=2}^T \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$ measures the variation of the comparator sequence.

Proof of Lemma 9. We decompose the instantaneous loss of the base algorithm as,

$$\begin{aligned}
&\langle \nabla \widehat{R}_t(\mathbf{w}_{t,i}), \mathbf{w}_{t,i} - \mathbf{u}_t\rangle \\
&\leq \underbrace{\langle \nabla \widehat{R}_t(\mathbf{w}_{t,i}) - \nabla H_t(\mathbf{w}_{t,i}), \mathbf{w}_{t,i} - \widehat{\mathbf{w}}_{t+1,i}\rangle}_{\texttt{term (a)}} + \underbrace{\langle \nabla H_t(\mathbf{w}_{t,i}), \mathbf{w}_{t,i} - \widehat{\mathbf{w}}_{t+1,i}\rangle}_{\texttt{term (b)}} + \underbrace{\langle \nabla \widehat{R}_t(\mathbf{w}_{t,i}), \widehat{\mathbf{w}}_{t+1,i} - \mathbf{u}_t\rangle}_{\texttt{term (c)}}.
\end{aligned}$$

For term (a), by Cauchy-Schwarz inequality and AM-GM inequality, we have

$$\begin{aligned}
\texttt{term (a)} &\leq \|\nabla \widehat{R}_t(\mathbf{w}_{t,i}) - \nabla H_t(\mathbf{w}_{t,i})\|_2 \|\mathbf{w}_{t,i} - \widehat{\mathbf{w}}_{t+1,i}\|_2 \\
&\leq 2\eta_i \|\nabla \widehat{R}_t(\mathbf{w}_{t,i}) - \nabla H_t(\mathbf{w}_{t,i})\|_2^2 + \frac{1}{2\eta_i}\|\mathbf{w}_{t,i} - \widehat{\mathbf{w}}_{t+1,i}\|_2^2.
\end{aligned}$$

Then, we bound term (b) by Proposition 4.1 of Campolongo and Orabona [34],

$$\texttt{term (b)} \leq \frac{1}{2\eta_i}\left(\|\widehat{\mathbf{w}}_{t+1,i} - \widehat{\mathbf{w}}_{t,i}\|_2^2 - \|\widehat{\mathbf{w}}_{t+1,i} - \mathbf{w}_{t,i}\|_2^2 - \|\widehat{\mathbf{w}}_{t,i} - \mathbf{w}_{t,i}\|_2^2\right).$$

The term (c) is bounded by Lemma 7 of Zhao et al. [19],

$$\texttt{term (c)} \leq \frac{1}{2\eta_i}\left(\|\mathbf{u}_t - \widehat{\mathbf{w}}_{t,i}\|_2^2 - \|\mathbf{u}_t - \widehat{\mathbf{w}}_{t+1,i}\|_2^2 - \|\widehat{\mathbf{w}}_{t,i} - \widehat{\mathbf{w}}_{t+1,i}\|_2^2\right).$$

Combining the three upper bounds yields

$$\begin{aligned}
&\sum_{t=1}^{T}\langle \widehat{R}_t(\mathbf{w}_{t,i}), \mathbf{w}_{t,i} - \mathbf{u}_t\rangle \\
&\leq 2\eta_i \sum_{t=1}^{T}\|\nabla \widehat{R}_t(\mathbf{w}_{t,i}) - \nabla H_t(\mathbf{w}_{t,i})\|_2^2 + \frac{1}{2\eta_i}\sum_{t=1}^{T}\left(\|\mathbf{u}_t - \widehat{\mathbf{w}}_{t,i}\|_2^2 - \|\mathbf{u}_t - \widehat{\mathbf{w}}_{t+1,i}\|_2^2\right) + \frac{\Gamma^2}{2\eta_i} \\
&\leq 2\eta_i \sum_{t=1}^{T}\|\nabla \widehat{R}_t(\mathbf{w}_{t,i}) - \nabla H_t(\mathbf{w}_{t,i})\|_2^2 + \frac{\Gamma^2 + 2\Gamma P_T}{2\eta_i} \\
&\leq 2\eta_i \sum_{t=1}^{T}\sup_{\mathbf{w}\in\mathcal{W}}\|\nabla \widehat{R}_t(\mathbf{w}) - \nabla H_t(\mathbf{w})\|_2^2 + \frac{\Gamma^2 + 2\Gamma P_T}{2\eta_i},
\end{aligned}$$

which completes the proof. □

G Technical Lemmas

Lemma 10. *Let $A \in \mathbb{R}^{K\times K}$ be an invertible matrix and $\Delta \in \mathbb{R}^{K\times K}$ be a squared matrix such that $\|A^{-1}\Delta\|_2 \leq 1/2$. For $B = A + \Delta$, we have the following two claims:*

- *B is an invertible matrix.*
- $\|B^{-1} - A^{-1}\|_2 \leq 2\|A^{-1}\|_2^2 \cdot \|\Delta\|_2$.

Proof. The proof follows the same arguments in [74, Page 381]. We first show that $B = A(I+A^{-1}\Delta)$ is an invertible matrix. Since A is an invertible matrix by the assumption, it is sufficient to show that $I + A^{-1}\Delta$ is invertible. The statement is true since the spectral radius of $A^{-1}\Delta$ satisfies $\rho(A^{-1}\Delta) \leq \|A^{-1}\Delta\|_2 \leq 1/2$. The bounded spectral radius implies that -1 is not an eigenvalue of the matrix $A^{-1}\Delta$, and thus it is invertible.

Then, we can upper bound the norm

$$\|B^{-1} - A^{-1}\|_2 = \|A^{-1}\Delta B^{-1}\|_2 \leq \|A^{-1}\Delta\|_2\|B^{-1}\|_2, \tag{32}$$

where the term $\|B^{-1}\|_2$ can be further bounded by

$$\|B^{-1}\|_2 \leq \|A^{-1}\|_2 + \|B^{-1} - A^{-1}\|_2 \leq \|A^{-1}\| + \|A^{-1}\Delta\|_2\|B^{-1}\|_2.$$

Rearranging the above inequality, we have

$$\|B^{-1}\|_2 \leq \frac{\|A^{-1}\|_2}{1 - \|A^{-1}\Delta\|_2}. \tag{33}$$

Then, plugging (33) into (32) and rearranging the terms, we have

$$\|B^{-1} - A^{-1}\|_2 \leq \frac{\|A^{-1}\Delta\|_2\|A^{-1}\|_2}{1 - \|A^{-1}\Delta\|_2} \leq \frac{\|A^{-1}\|_2^2\|\Delta\|_2}{1 - \|A^{-1}\Delta\|_2} \leq 2\|A^{-1}\|_2^2\|\Delta\|_2,$$

which completes the proof. □

Lemma 11 (Theorem 19 of Syrgkanis et al. [75])**.** *Let $\boldsymbol{\ell}_t \in \mathbb{R}^N$ be the loss vector taking $\ell_{t,i} \in \mathbb{R}$ as its i-th entry and $\mathbf{m}_t \in \mathbb{R}^N$ the hint vector taking $m_{t,i}$ as its i-th entry. The Hedge algorithm updating with $p_{t,i} \propto \exp\Big(-\varepsilon\big(\sum_{s=1}^{t-1}\ell_{s,i} + m_{t,i}\big)\Big)$ satisfies*

$$\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j}\ell_{t,j} - \sum_{t=1}^{T}\ell_{t,i} \leq \frac{\ln N + 2}{\varepsilon} + \varepsilon\sum_{t=1}^{T}\|\boldsymbol{\ell}_t - \mathbf{m}_t\|_\infty^2$$

for any $i \in [N]$, where $\varepsilon > 0$ is the step size.