Adapting to Online Label Shift with Provable Guarantees

Yong Bai\textsuperscript{1\dag}, Yu-Jie Zhang\textsuperscript{2,1\ast}, Peng Zhao\textsuperscript{1}, Masashi Sugiyama\textsuperscript{3,2}, Zhi-Hua Zhou\textsuperscript{1†}

\textsuperscript{1} National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
\textsuperscript{2} The University of Tokyo, Chiba, Japan
\textsuperscript{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 \cite{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 \cite{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 \cite{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
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, which consists of two stages, including offline initialization and online adaptation. Specifically, the learner collects labeled samples drawn independently from the initial distribution \(D_0(x, y)\) where \(x\) 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 \(D_t(x)\) and uses them to update the model. In OLaS problems, the essential environment change happens on the label distribution \(D_t(y)\) with the conditional \(D(x | y)\) 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 \(O(\sqrt{T})\) 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 (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

---

3We use OLaS instead of OLS, since OLS often refers to “ordinary least squares”. 

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...
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 $S_0 = \{(x_n, y_n)\}_{n=1}^{N_0}$ drawn from the distribution $D_0(x, y)$ 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 $S_t = \{x_n\}_{n=1}^{N_t}$ drawn from the distribution $D_t(x)$. The non-stationary nature indicates $D_t \neq D_t'$ 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 $D_t(y)$ changes over time while the class-conditional distribution $D_t(x \mid y)$ is identical throughout the process for $t \in \{0, 1, \ldots, T\}$. Moreover, it holds that $D_0(y) > 0$ 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 $\Gamma \triangleq \sup_{w, w' \in \mathcal{W}} \|w - w'\|_2$. The goal is to ensure that the $t$-round model $w_t$ generalizes well on the underlying distribution $D_t$. Thus, the model’s quality is evaluated by its risk defined as $R_t(w) = \mathbb{E}_{(x,y) \sim D_t} [\ell(f(w, x), y)]$, where $f : \mathcal{W} \times \mathcal{X} \mapsto \mathbb{R}$ is the predictive function and $\ell : \mathbb{R}^K \times \mathcal{Y} \mapsto \mathbb{R}$ is any convex surrogate loss for classification such that $\ell(f(w, x), y)$ is convex in $w$. We introduce two constants, $G \triangleq \sup_{(x,y) \in \mathcal{X} \times \mathcal{Y}, w \in \mathcal{W}} \|\nabla_w \ell(f(w, x), y)\|_2$ and $B \triangleq \sup_{(x,y) \in \mathcal{X} \times \mathcal{Y}, w \in \mathcal{W}} |\ell(f(w, x), y)|$, 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

$$\text{Reg}^d_T \triangleq \sum_{t=1}^{T} R_t(w_t) - \sum_{t=1}^{T} \min_{w \in \mathcal{W}} R_t(w) = \sum_{t=1}^{T} R_t(w_t) - \sum_{t=1}^{T} R_t(w^*_t),$$

where $w^*_t \in \arg \min_{w \in \mathcal{W}} R_t(w)$ is the model (or one of the models) with the best generalization ability on the distribution $D_t$. 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 $\text{Reg}^s_T = \sum_{t=1}^{T} R_t(w_t) - \sum_{t=1}^{T} R_t(w^*)$, where $w^* \in \arg \min_{w \in \mathcal{W}} \sum_{t=1}^{T} R_t(w)$ 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 \( \{w_t\}_{t=1}^T \) enjoying low cumulative risk \( \sum_{t=1}^T R_t(w_t) \). Thus, suppose the model’s risk \( R_t(w_t) \) is known at each round; then OGD simply updates the model by \( w_{t+1} = \Pi_{W} [w_t - \eta \nabla R_t(w_t)] \), where \( \Pi_{W} \) denotes the projection onto \( W \) and \( \eta > 0 \) is the step size. It is well-known that OGD guarantees the regret bound \( \sum_{t=1}^T R_t(w_t) - \min_{w \in W} \sum_{t=1}^T R_t(w) \leq O(\sqrt{T}) \) when risk function \( R_t(w) \) 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 \( R_t(w) \) is unknown in the current OLaS setup as it is defined over the underlying joint distribution \( D_t(x, y) \). More severely, the online environments in the OLaS problem are fully unlabelled, 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 \( \hat{R}_t(w) \) 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 \( \hat{R}_t(w) \) with unlabeled data \( S_t \) and offline data \( S_0 \) by the risk rewriting technique. To this end, let \( \mu_{y_k} \in \Delta_{K} \) denote the label distribution vector with the \( k \)-th entry \( [\mu_{y_k}]_k \triangleq D_t(y = k) \), then we have the following decomposition for the true risk:

\[
R_t(w) \triangleq \mathbb{E}_{(x, y) \sim D_t}[\ell(f(w, x), y)] = \sum_{k=1}^{K} [\mu_{y_k}]_k \cdot \hat{R}_t^k(w) = \sum_{k=1}^{K} [\mu_{y_k}]_k \cdot \hat{R}_0^k(w),
\]

where \( \hat{R}_t^k(w) \triangleq \mathbb{E}_{x \sim D_t(x | y = k)}[\ell(f(w, x), k)] \) is the risk of the model over the \( k \)-th class 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 \( D_t(x | y) = D_0(x | y) \) for any \( t \in [T] \). Since the labeled offline data \( S_0 \) is always available, one can approximate \( \hat{R}_t^k(w) \) by its empirical version \( \hat{R}_0^k(w) \) with offline data \( S_0 \), where \( \hat{R}_0^k(w) = \frac{1}{|S_0|} \sum_{x_n \in S_0} \ell(f(w, x_n), k) \) and \( S_0 \) is a subset of \( S_0 \) containing all samples with label \( k \).

Therefore, the task is now to estimate the label distribution vector \( \mu_{y_k} \). 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 \( \hat{y}_t \), and next estimate the label distribution via solving the crucial equation \( \mu_{y_k} = C_{f_0}^{-1} \mu_{\hat{y}_t} \), where \( \mu_{\hat{y}_t} \in \Delta_{K} \) is the distribution vector of the predictive labels \( \hat{y}_t \) and \( C_{f_0} \in \mathbb{R}^{K \times K} \) is the confusion matrix with \( [C_{f_0}]_{ij} = \mathbb{E}_{x \sim D_0(x | y = j)}[\mathbbm{1}\{f_0(x) = i\}] \) 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 \( \hat{R}_t(w) \):

\[
\hat{R}_t(w) = \sum_{k=1}^{K} \hat{C}_{f_0}^{-1} \mu_{\hat{y}_t} \cdot \hat{R}_0^k(w),
\]

where \( \hat{C}_{f_0} \) and \( \hat{\mu}_{\hat{y}_t} \) 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 \( \hat{R}_t(w) \) in Eq. (3) is unbiased to \( R_t(w) = \mathbb{E}_{(x, y) \sim D_t}[\ell(f(w, x), y)] \), i.e., \( \mathbb{E}_{S_t \sim D_t}[\hat{R}_t(w)] = R_t(w) \) for any \( w \in W \) independent of the dataset \( S_t \), provided \( C_{f_0} \) is invertible and the offline dataset \( S_0 \) has sufficient samples such that \( \hat{C}_{f_0} = C_{f_0} \) and \( \hat{R}_0^k(w) = R_0^k(w), \forall k \in \mathcal{Y} \).

The proof of Lemma 1 is in Appendix D.1. Note that the sufficient sample assumption is introduced on offline data \( S_0 \) to simplify the presentation. Indeed, we can further show \( |\mathbb{E}_{S_t \sim D_t}[\hat{R}_t(w)] - R_t(w)| \leq O(1/|S_0|) \) with high probability (details in Appendix D.2). Such an additional dependence on \( S_0 \) also appears in the classical offline label shift [12, 13] and is negligible when a large number of offline data is collected at the initial stage. The requirement is easy to realize and will not trivialize the online adaptation. Another caveat is that storing all the offline data can be burdensome in resource-constrained learning scenarios, then one may use data sketching techniques like corsets [22, 23] or reduced kernel mean embedding [24, 25, 26] to further improve the storage complexity.
Moreover, which are indispensable for the algorithm design and theoretical analysis. Concretely, Wu et al. [16] assumed that UOGD does not require. Under the same assumptions as Lemma 1, UOGD in Eq. (4) with step size \( \eta \) achieves a dynamic regret scaling with the label distribution drift \( V_T \). By contrast, ours does not require the restart in the algorithm.

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

\[
\mathbb{E} [\text{Reg}_T] \leq 2(KG^2/\sigma^2 + B^2)\eta T + T^2/\eta + 4(\Gamma + 1)\sqrt{BV_T T/\eta} = O\left(\eta T + 1 + \eta + \sqrt{V_T T}/\eta\right),
\]

where the constant \( \sigma > 0 \) denotes the minimum singular value of the invertible confusion matrix \( C_f_0 \). Moreover, \( V_T = \sum_{t=2}^{T} \|\mu_{y_t} - \mu_{y_{t-1}}\|_1 \) 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 2 (Non-convexity issue). Our risk estimator \( \hat{R}_t(w) \) can be non-convex as the estimated label distribution \( \bar{C}_t, \bar{\mu}_y \) 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 \( R_t(w) \) 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,

\[
w_{t+1} = \Pi_{W_t}[w_t - \eta \nabla \hat{R}_t(w_t)].
\]

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 \( O(\sqrt{T}) \) 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 only be 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 \( O(\sqrt{T}) \) 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 \( V_T \).

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, 27] on the surface; however, our result exhibits fundamental differences. The key caveat is that in our case the comparator \( w^*_t \) at each round is not the minimizer of the online function. Specifically, as the expected risk \( \hat{R}_t \) is inaccessible, one has to work on the unbiased risk estimator \( \hat{R}_t \) and requires optimizing the empirical dynamic regret \( \sum_{t=1}^{T} \hat{R}_t(w_t) - \sum_{t=1}^{T} \hat{R}_t(w^*_t) \). Importantly, \( w^*_t \in \arg\min_{w \in \mathcal{W}} \hat{R}_t(w) \) but \( w^*_t \notin \arg\min_{w \in \mathcal{W}} \hat{R}_t(w) \) in general. Although the empirical dynamic regret can be trivially bounded by \( \sum_{t=1}^{T} \hat{R}_t(w_t) - \sum_{t=1}^{T} \min_{w \in \mathcal{W}} \hat{R}_t(w) \), the bound will then be loose and related to temporal variation of risk estimators [17, 27], making it hard to establish relationship to the natural non-stationarity measure of OLaS: \( V_T \). 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

\[\text{Reg}_T \leq O(\sqrt{T})\]
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 \( \eta_k \) 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 \( \mathbf{w}_{t,i} = \Pi_{\mathcal{W}}[\mathbf{w}_{t-1,i} - \eta_t \nabla \tilde{R}_{t-1}(\mathbf{w}_{t-1,i})] \) with \( \eta_t \in \mathcal{H} \).

**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 \( p_t \in \Delta_N \) is the weight vector with \( p_{t,i} \) denoting the weight of the \( i \)-th base-learner. We use the classic Hedge algorithm [32] to update the weight vector, namely, \( p_{t,i} \propto \exp(-\varepsilon \sum_{s=1}^{t-1} \tilde{R}_s(\mathbf{w}_{s,i})) \), where \( \varepsilon > 0 \) is the learning rate of the meta-algorithm.
that can be simply set as \( O(\sqrt{\ln N}/T) \) 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 \( \{ w_t \}_{t=1}^T \) 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 \( H = \{ \eta_i = \frac{\sqrt{2}}{2G\sqrt{R_T}} \cdot 2^{i-1} \mid i \in [N] \} \), where \( N = 1 + \left\lceil \frac{1}{2} \log_2 (1 + 2T) \right\rceil \) is the number of base-learners. ATLAS ensures that \( E[\text{Reg}_p^d] \leq O(\max\{V_T^1T^{2\over 3}, \sqrt{T}\}), \)

for non-degenerated cases of \( V_T \geq \Theta(T^{-{2\over 3}}) \).

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 minmax 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 [33]. 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 [34]. We build on the framework of optimistic online learning [35, 36] by introducing a hint function \( H_t : W \rightarrow \mathbb{R} \) to encode shift patterns from historical data, which serves as an estimation of the expected risk \( R_t(w_i) \). 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

\[
\tilde{w}_{t,i} = \Pi_{W}[\tilde{w}_{t-1,i} - \eta_t \nabla R_{t-1}(w_{t-1,i})], \quad w_{t,i} = \arg \min_{w \in W} \eta_t H_t(w) + 1/2 \cdot \| w - \tilde{w}_{t,i} \|_2^2, \quad (6)
\]

where \( \tilde{w}_{t,i} \) is an intermediate output and \( w_{t,i} \) is the final returned model. When \( H_t(w) = 0 \) (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 \( w_{t,i} = \tilde{w}_{t,i} \). 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

\[
p_{t,i} \propto \exp \left( -\varepsilon \left( \sum_{s=1}^{t-1} \tilde{R}_s(w_s,i) + H_t(w_{t,i}) \right) \right)
\]

for all \( i \in [N] \), where \( \varepsilon \) 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 \( H_t(w_{t,i}) \) evaluated over the current local models \( \{ w_{t,i} \}_{i=1}^N \) 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 \( H_t : W \rightarrow \mathbb{R} \) is convex, satisfies \( \max_{w \in W} \| \nabla H_t(w) \|_2 \leq \max_{w \in W} \| \nabla \tilde{R}_t(w) \|_2 \), and is independent of current data \( S_t \). Set the step size pool as \( H = \{ \eta_i = \frac{\sqrt{2}}{2G\sqrt{R_T}} \cdot 2^{i-1} \mid i \in [N] \} \) with \( N = 2 + \left\lceil \frac{1}{2} \log_2 (3T(1 + 4G^2KT/\sigma)) \right\rceil \). ATLAS-ADA ensures \( E[\text{Reg}_p^d] \leq O(V_T^{1/3}G_T^{1/3}T^{1/3}) \), where \( G_T = \sum_{t=1}^T E[\sup_{w \in W} \| \nabla \tilde{R}_t(w) - \nabla H_t(w) \|_2] \) measures the reusability of historical information, depending on label shift patterns and hint function designs.

In the worst case, \( G_T \) is at most \( O(T) \) given a bounded gradient norm \( \| \nabla H_t(w) \|_2 \), and thus the bound presented in Theorem 3 safeguards the same \( O(V_T^{1/3}T^{2/3}) \) bound as ATLAS. More
importantly, when the hint function $H_i(\cdot)$ 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 $w_{t,i} = \Pi_{w \in \mathcal{W}}[\hat{w}_{t,i} - \eta_t \nabla H_i(w_{t-1,i})]$, which can be deemed as an approximated optimization over the linearized loss $\langle \nabla H_i(w_{t-1,i}), w \rangle$. By contrast, we directly updates over the **original function** $H_i(w)$, hence called the “implicit” update [37, 38]. 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 \( \tilde{R}_i(w) = \sum_{k=1}^K [\hat{\mu}_{y_t}]_k \cdot \tilde{R}_k(w) \). Thus, a natural construction is $H_i(w) = \sum_{k=1}^K [h_{y_t}]_k \cdot \tilde{R}_k(w)$ parametrized by hint priors $h_{y_t} \in \mathbb{R}^K$, which is used to estimate the class prior based on the past observed data $\{S_t\}_{t=1}^{T-1}$. Then, $G_T$ satisfies the bias-variance decomposition (with bias term $\mathbb{E} \| h_{y_t} - \hat{\mu}_{y_t}^2 \|$ and variance term $\mathbb{E} \| \| \hat{\mu}_{y_t} - \mu_{y_t} \|_2^2 \|)$:

$$G_T \leq KG^2 \sum_{t=1}^T \mathbb{E} \| h_{y_t} - \hat{\mu}_{y_t}^2 \| \leq 2KG^2 \sum_{t=1}^T \left( \mathbb{E} \| h_{y_t} - \hat{\mu}_{y_t}^2 \| + \mathbb{E} \| \mu_{y_t} - \hat{\mu}_{y_t}^2 \| \right).$$

Setting $h_{y_t} = \mu_{y_t}$ will make the upper bound tighter, though the underlying class prior $\mu_{y_t}$ 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 ($\text{fix}$), three OLaS algorithms proposed by the previous work [16] ($\text{ROGD}, \text{FTH}, \text{FTFWH}$), and our proposals ($\text{UOGD}, \text{ATLAS}, \text{ATLAS-ADA}$) 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** ($\text{Sin}$) and **Square Shift** ($\text{Squ}$) 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 **Linear Shift** ($\text{Lin}$) while changes fast in **Bernoulli Shift** ($\text{Ber}$). We repeat all experiments for five times and evaluate the contenders by the average error $\frac{1}{T} \sum_{t=1}^T \frac{1}{|S_t|} \sum_{n=1}^{|S_t|} 1[f(w, x_n) \neq y_n]$ 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 $N_i = 100$ samples are received at each iteration. Basically, $\text{fix}$ is inferior to the online methods, which shows the necessity of designing online algorithms for the OLaS problem. Besides, $\text{UOGD}$ outperforms $\text{ROGD}$, 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 $\text{UOGD}$ (with step size $\eta = \Theta(T^{-1/2})$) when the environments change relatively fast ($\text{Squ, Sin, and Ber}$). The results justify our theoretical finding that the small
To illustrate how the hint function works, we further vary the buffer size of the Periodic Hint (\( \text{Peri} \)). The results show that our algorithm can adaptively track the suitable step sizes according to the shift tendency as the results in the synthetic experiments. When the distribution changes rapidly (\( \text{Sin} \) and \( \text{Square} \) shift, see Appendix A.1 for details). Even in a relatively stationary environment (\( \text{Lin} \)), our algorithms are comparable with the best algorithm (UOGD), which is specifically designed for learners with small step sizes. In the fast-changing case, see Figure 1(b), larger step sizes are preferred. Figure 1 illustrates the case of slowly changing environments, where more weights are assigned to the base-learners along the learning process. Specifically, Figure 1(a) assigns larger weights to appropriate base-learners along the learning process. 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.

**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 \( p_{i,t} \) 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 different sample sizes. 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 (\( \text{Peri} \)) on the \( \text{Square} \) environment (\( \text{Sin} / 2 \)), which shifts in a periodic length \( L = 40 \). As shown in Figure 2, when the buffer size matches the multiples of length \( L, \text{Peri} \) 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 [39] 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 (\( \text{Ber} \)), ATLAS and ATLAS-ADA outperform other contenders. Similar results are also observed in the \( \text{Lin} \) and \( \text{Square} \) shift, see Appendix A.1 for details. Even in a relatively stationary environment (\( \text{Lin} \)), 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, we further validate the adaptivity of our algorithm

### Table 1: Average error (%) for different algorithms under various simulated shifts for the synthetic data.

<table>
<thead>
<tr>
<th>Type</th>
<th>Sample Size: 1</th>
<th>Sample Size: 10</th>
<th>Sample Size: 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin</td>
<td>None</td>
<td>Fwd</td>
<td>OKM</td>
</tr>
<tr>
<td></td>
<td>5.47</td>
<td>5.38</td>
<td>5.42</td>
</tr>
<tr>
<td>Squ</td>
<td>5.48</td>
<td>5.38</td>
<td>5.42</td>
</tr>
<tr>
<td>Sin</td>
<td>5.48</td>
<td>5.38</td>
<td>5.42</td>
</tr>
<tr>
<td>Ber</td>
<td>5.48</td>
<td>5.38</td>
<td>5.42</td>
</tr>
</tbody>
</table>

*Step Size (\( \square \)): 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 12.8.*

### 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).

<table>
<thead>
<tr>
<th>Shift Type</th>
<th>Sample Size: 1</th>
<th>Sample Size: 10</th>
<th>Sample Size: 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin</td>
<td>None</td>
<td>Win</td>
<td>Fwd</td>
</tr>
<tr>
<td></td>
<td>5.58</td>
<td>5.48</td>
<td>5.42</td>
</tr>
<tr>
<td>Squ</td>
<td>5.58</td>
<td>5.48</td>
<td>5.42</td>
</tr>
<tr>
<td>Sin</td>
<td>5.58</td>
<td>5.48</td>
<td>5.42</td>
</tr>
<tr>
<td>Ber</td>
<td>5.58</td>
<td>5.48</td>
<td>5.42</td>
</tr>
</tbody>
</table>

*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 (\( N_t = 1 \)).
Table 3: Average error (%) of different algorithms on various real-world datasets (Lin and Bar). 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 $N_t = 10$.

<table>
<thead>
<tr>
<th></th>
<th>FIX</th>
<th>FTH</th>
<th>FTFWH</th>
<th>ROGD</th>
<th>UOGD</th>
<th>ATLAS</th>
<th>AT-ADA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ArXiv</td>
<td>±0.07</td>
<td>±0.28</td>
<td>±0.21</td>
<td>±0.20</td>
<td>±0.11</td>
<td>±0.09</td>
<td>±0.09</td>
</tr>
<tr>
<td></td>
<td>±0.20</td>
<td>±0.13</td>
<td>±0.19</td>
<td>±0.11</td>
<td>±0.68</td>
<td>±0.70</td>
<td>±0.69</td>
</tr>
<tr>
<td>EuroSAT</td>
<td>±0.11</td>
<td>±0.12</td>
<td>±0.11</td>
<td>±0.10</td>
<td>±0.11</td>
<td>±0.13</td>
<td>±0.09</td>
</tr>
<tr>
<td></td>
<td>±0.09</td>
<td>±0.11</td>
<td>±0.05</td>
<td>±0.14</td>
<td>±0.04</td>
<td>±0.03</td>
<td>±0.05</td>
</tr>
<tr>
<td>MNIST</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.02</td>
</tr>
<tr>
<td></td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.03</td>
<td>±0.03</td>
<td>±0.03</td>
<td>±0.03</td>
<td>±0.02</td>
</tr>
<tr>
<td>Fashion</td>
<td>±0.04</td>
<td>±0.07</td>
<td>±0.06</td>
<td>±0.08</td>
<td>±0.07</td>
<td>±0.08</td>
<td>±0.09</td>
</tr>
<tr>
<td></td>
<td>±0.09</td>
<td>±0.11</td>
<td>±0.10</td>
<td>±0.08</td>
<td>±0.12</td>
<td>±0.12</td>
<td>±0.12</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>±0.12</td>
<td>±0.12</td>
<td>±0.14</td>
<td>±0.15</td>
<td>±0.11</td>
<td>±0.14</td>
<td>±0.12</td>
</tr>
<tr>
<td></td>
<td>±0.12</td>
<td>±0.12</td>
<td>±0.14</td>
<td>±0.15</td>
<td>±0.11</td>
<td>±0.14</td>
<td>±0.12</td>
</tr>
<tr>
<td>CINIC10</td>
<td>±0.22</td>
<td>±0.10</td>
<td>±0.13</td>
<td>±0.29</td>
<td>±0.15</td>
<td>±0.19</td>
<td>±0.16</td>
</tr>
<tr>
<td></td>
<td>±0.35</td>
<td>±0.17</td>
<td>±0.19</td>
<td>±0.19</td>
<td>±0.55</td>
<td>±0.58</td>
<td>±0.60</td>
</tr>
</tbody>
</table>

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.

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.

to the underlying environment regardless of the fast or slow changes, we simulate various shift 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 $O(V_T^{1/3}T^{2/3})$ guarantee, and ATLAS-ADA further introduced hint functions to exploit the shift structure and obtained an improved $O(V_T^{1/3}G_T^{1/3}T^{1/3})$ 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, 62206125), 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, Yu-Yang Qian, and Dheeraj Baby for their helpful discussions.
References


**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]

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]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (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]
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]

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.