Adapting to Generalized Online Label Shift by Invariant Representation Learning

Yu-Yang Qian
National Key Laboratory for Novel Software Technology,
School of Artificial Intelligence,
Nanjing University
Nanjing, Jiangsu, China
qianyy@lamda.nju.edu.cn

Yi-Han Wang
National Key Laboratory for Novel Software Technology,
School of Artificial Intelligence,
Nanjing University
Nanjing, Jiangsu, China
wangyh@lamda.nju.edu.cn

Zhen-Yu Zhang
Center for Advanced Intelligence Project, RIKEN
Tokyo, Japan
zhen-yu.zhang@riken.jp

Yuan Jiang
National Key Laboratory for Novel Software Technology,
School of Artificial Intelligence,
Nanjing University
Nanjing, Jiangsu, China
jiangy@lamda.nju.edu.cn

Zhi-Hua Zhou*
National Key Laboratory for Novel Software Technology,
School of Artificial Intelligence,
Nanjing University
Nanjing, Jiangsu, China
zhouzh@lamda.nju.edu.cn

Abstract

The problem of online label shift, where label distribution evolves over time while the label-conditional density remains unchanged, has attracted increasing attentions. Although existing approaches have achieved sound theoretical guarantees and encouraging performance, the assumption of an unchanged conditional distribution may limit its application in broader tasks. In this paper, we investigate an extended variant named *generalized online label shift* (GOLS) problem, in which we relax the label shift assumption on the raw feature space and instead assume the existence of an unknown *invariant representation* such that conditional distribution of this representation given the label remains constant. To handle GOLS, our main idea involves capturing the inherently stable information from non-stationary streams, in the form of learning an invariant representation. Specifically, we design a novel objective to learn the invariant representation, which exploits the unique structure in GOLS. To optimize this objective, we propose an algorithm employing online ensemble paradigm to perform *multi-resolution updates* using various historical data windows, thereby enhancing the stability of the representation. This approach is theoretically guaranteed to achieve an *optimal convergence rate*. To improve the efficiency of the ensemble framework, we further propose a mask-based implementation for ensembling with DNNs. Experiments on benchmarks and real-world tasks validate the effectiveness of our approach.

*Zhi-Hua Zhou is the corresponding author.

KDD '25, August 3–7, 2025, Toronto, ON, Canada

ACM ISBN 979-8-4007-1245-6/25/08
https://doi.org/10.1145/3690624.3709182

CCS Concepts

• **Computing methodologies** → **Machine learning algorithms**; **Online learning settings**.

Keywords

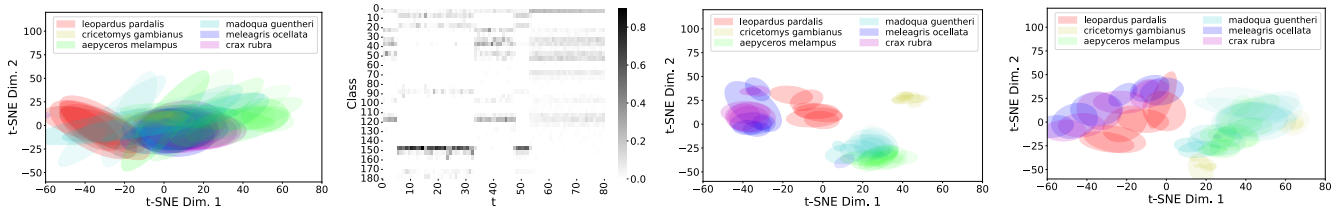
data stream; label shift; distribution shift; efficient online learning

ACM Reference Format:
Yu-Yang Qian, Yi-Han Wang, Zhen-Yu Zhang, Yuan Jiang, and Zhi-Hua Zhou. 2025. Adapting to Generalized Online Label Shift by Invariant Representation Learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3690624.3709182

1 Introduction

Machine learning methods that handle distribution shifts in online and open environments have attracted more and more interests nowadays [15, 32, 33, 46]. Recently, the *online label shift* problem, a fundamental and crucial scenario, has attracted increasing attentions [2, 3, 28, 29, 40], in which label distribution shifts as time evolves while the class-conditional distribution remains same. Research on online label shift scenarios has pioneered the exploitation of online learning approaches to deal with online distribution shift problems, achieving both rigorous theoretical guarantees and encouraging performance. Although existing algorithms for online label shift have achieved remarkable success, the assumption of unchanged conditional distribution $\mathcal{D}(\mathbf{x} \mid y)$ may limit their applications in broader tasks, especially when the raw data's complexity requires deep neural networks (DNNs) to extract representations.

Motivated by above challenge of assumption mismatches in real-world data when applying label shift algorithms, we investigate the *generalized online label shift* (GOLS) problem [12, 39], an extended scenario of label shift. The key element of GOLS lies in the existence of an unknown *invariant representation* (or feature extractor) $\phi^\star(\cdot) : \mathcal{X} \mapsto \mathcal{H}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is the feature space and $\mathcal{H} \subseteq \mathbb{R}^{d'}$ is



(a) Raw feature's distribution $\mathcal{D}_t(\mathbf{x} \mid y)$ (b) Label distribution $\mathcal{D}_t(y)$ (c) $\mathcal{D}_t(\phi(\mathbf{x}) \mid y)$ of AGOE (d) $\mathcal{D}_t(\phi(\mathbf{x}) \mid y)$ of [39]

Figure 1: We visualize iWildCam [4] dataset to test if the *generalized online label shift* assumption holds. (a) T-SNE Visualization of raw feature's conditional distribution given the label y. The conditional distribution of raw features given the label $\mathcal{D}_t(\mathbf{x} \mid y)$ is unstable, indicating previous label shift assumption may not hold in real-world tasks. (b) Visualization of label distributions in data stream, where we calculate the heatmap of label distribution at each round. (c) & (d) Visualization of the conditional distribution of representations learned by ours and by Wu et al. [39]. The transparency indicates timestamp of that cluster. The conditional distribution $\mathcal{D}_t(\phi(\mathbf{x}) \mid y)$ remains more consistent across different timestamps by our AGOE than by [39], as we design a *multi-resolution updating* strategy to better learn the representation.

the representation space. This invariant representation ensures that, even if the raw feature space does not satisfy label shift assumption, the conditional distribution of representations $\mathcal{D}(\phi^\star(\mathbf{x}) \mid y)$ remains the same in the data stream. Therefore, the GOLS problem is a flexible and general variant of label shift, and is prevalent in many real-world tasks. For instance, in the wild animal species recognition tasks [4], the appearances of each species remain consistent, but the proportion of different animals varies with season and location. Concretely, we analyze a real-world animal recognition dataset iWildCam [4] to test if the GOLS assumption holds, where we employ a MobileNetV2 [31] as the feature extractor ϕ. As illustrated in Figure 1 (a), (b), and (c), the conditional distribution of raw features $\mathbf{x}$ given the label y is unstable, while the conditional distribution $\mathcal{D}_t(\phi(\mathbf{x}) \mid y)$ is stable across all environments, and the label distribution $\mathcal{D}_t(y)$ changes over time.

Recently, Wu et al. [39] investigate the GOLS problem and employed a self-supervised learning method to update the representation based only on the current unlabeled data, which achieves promising empirical performance. In this paper, we aim to leverage historical information to learn a more stable representation in GOLS. This is challenging because the data distribution is changing over time, requiring us to adaptively determine the appropriate length of historical data to reuse. By appropriately reusing historical information, as illustrated in Figure 1 (c) and (d), the conditional distribution exhibits more consistent across different timestamps given representation learned by ours than by [39], highlighting the benefits of exploiting historical information.

We propose our *Adapting to GOLS by Online Ensemble* (AGOE) approach by exploiting *multi-resolution updates* to learn stable representations. Our principal idea is to capture the inherently stable information from non-stationary data streams, with the form of learning an *invariant representation*. Specifically, to make use of the unique structure in GOLS data stream, we first propose a novel objective to learn the invariant representation, which regularizes the representation to align with classifiers thereby enhancing the stability of the representation. Then, we optimize the objective by making use of the online ensemble paradigm [45] to leverage different lengths of historical data windows to better learn a stable representation in, which is theoretically proved to achieve the *optimal convergence rate* in the non-convex non-smooth scenarios.

To further enhance the *efficiency* of our ensemble paradigm, we propose a mask-based mechanism for online ensemble with DNNs, where we segment the network into multiple sub-parts, each treated as a base learner that is updated with a different scope of historical data. Then, the weights of the meta learner are used as learning rates, determining the importance of different historical information. Therefore, we perform *multi-resolution updates* by using various historical data windows, thereby learning a stable representation. Finally, we validate our approach empirically, demonstrating its effectiveness on benchmarks and real-world datasets, including computer vision and natural language processing tasks.

Contribution. Our contributions are mainly three-fold:

- We investigate a generalized version of online label shift problem named GOLS, and we introduce a *novel objective* to learn the *invariant representation* by exploiting unique structure of GOLS.
- We optimize the objective by the online ensemble paradigm, performing *multi-resolution updates* by various historical data windows. We achieve the *optimal convergence rate* and tackle the remaining problem of Cutkosky et al. [9] that algorithm's parameters need to be set according to the quality of initial point.
- We provide an *efficient mask-based implementation* of our approach for online ensemble with DNN architectures, which improves the computational and storage efficiency of the algorithm.

2 Related Work

In this section, we discuss the related works to our paper.

Online Label Shift. The label shift problem has been extensively studied in offline scenarios [30, 43]. Recently, the more challenging *online label shift* where label distribution evolves over time has attracted increasing attentions. Wu et al. [40] make the first attempt, they develop an unbiased risk estimator using unlabeled data for model assessment and employ online gradient descent for model updating. Bai et al. [3] introduce an algorithm based on the online ensemble structure, achieving dynamic regret guarantees by maintaining a group of base learners, each with a different step size, and employing a meta learner to combine their outputs and adapt to environmental shifts. Baby et al. [2] transform the online label shift problem into an online regression problem, and utilize

an ensemble-based approach that reweights the initial classifier to adapt to new environments. Recently, Qian et al. [29] propose a wavelet-based method to handle the online label shift problem based on the restarting mechanism, which restarts the classifier once a significant distribution shift is detected.

However, these methods typically focus on the case of traditional label shift scenarios. Such an assumption may be hard to satisfy, especially in many real-world applications where DNNs are employed to extract the representation of the data. To this end, Wu et al. [39] study the problem of generalized online label shift, and employ a self-supervised learning method to update representations. Albeit with promising empirical performance, the small sample size in each round may potentially result in unstable representation updates, which may harm the learning of the invariant representation, and they do not consider to exploit historical information to enhance the updating of representations. Besides, theoretical property of their method for learning the representation remains unclear.

Meta-Learning. Meta-learning is another popular research area of its own interests, aiming to learn a shared model prior for multiple learning episodes. A classic meta-learning problem is often formulated as a bi-level optimization problem [16], in which the inner-level takes a few gradient descent steps to adapt the model to the current task and the outer-level evaluates the model after adaptation steps are taken [13, 38]. Recent progress has been made in online meta-learning within the streaming scenario [14, 44], where task instances are sequentially revealed, and the learner learns the latest task in each round. However, meta-learning methods typically assume the existence of a global gradient direction to optimize the model, which does not hold in non-stationary GOLS streams.

Learning the Representation. There are several previous attempts to learn the representation to improve the learning performance. Many meta-learning methods focus on extracting knowledge from a variety of tasks and adapting it to new tasks that were not seen during training, by extracting a good representation across multiple training tasks. Recently, Liu et al. [20] introduce a method that initially trains an offline representation, and then "fix" the representation and utilize gradients for adaptation during the testing phase. However, these methods tend to learn a representation in an offline manner and then fix it once and for good, which is not suitable for our setting where the representation needs to be updated in an online manner in the non-stationary data stream.

Recently, several sequential methods that update representations to adapt to the new environments have been proposed in the literature. Continual learning methods [8, 18, 22] have been introduced to update model representations to avoid the catastrophic forgetting phenomenon in the data stream. More recently, test-time adaptation (TTA) methods [6, 34, 37] have been developed to adjust model outputs for online test domains in the absence of labeled data from test distributions. However, these works primarily focus on the shifting environments in a broad context, which may be too general to capture the special structure of GOLS streams.

3 Problem Formulation

We consider multi-class classifications. We denote $\mathcal{X} \subseteq \mathbb{R}^d$ as the feature space, $\mathcal{H} \subseteq \mathbb{R}^{d'}$ as the representation space, and $\mathcal{Y} = [0,1]^K$ as the label space. The model $h : \mathcal{X} \mapsto \mathcal{Y}$ consists of two components: a representation function (or feature extractor) $\phi : \mathcal{X} \mapsto \mathcal{H}$, and a classifier $\mathbf{w} : \mathcal{H} \mapsto \mathcal{Y}$ to output the prediction. The representation function ϕ is specified as $\phi(\mathbf{x}) = \varphi(\mathbf{x}; \Phi)$, where φ is the function of the representation model, such as a deep neural network; and $\Phi \in \mathbb{R}^n$ denotes the parameter of this function, e.g., parameters of a DNN. Consequently, the model h can be decomposed as $h = \mathbf{w} \circ \phi$. We formulate the GOLS problem into two stages: (i) offline initialization and (ii) online adaptation.

(i) Offline Initialization. In the offline initialization stage, the learner collects a set of labeled data $S_0 = \{\mathbf{x}_i, y_i\}_{i=1}^{m_0}$ from the offline distribution $\mathcal{D}_0(\mathbf{x}, y)$. The goal of initialization is to obtain a well-performed initial representation Φ_0 and classifier $\mathbf{w}_0$ that generalize well over the distribution $\mathcal{D}_0$.

(ii) Online Adaptation. After initialization, the learner deploys the model to a shifting data stream. At each round t, the environment reveals the current distribution $\mathcal{D}_t$ and the learner only receives a *small-size* labeled data batch S_t, where $S_t = \{\mathbf{x}_i, y_i\}_{i=1}^{m_t}$ is i.i.d. sampled from the distribution $\mathcal{D}_t$. The assumption of the GOLS is formally presented as follows.

Assumption 1 (Generalized online label shift). *The label-conditional distributions given the optimal representation $\phi^\star(\cdot) : \mathcal{X} \mapsto \mathcal{H}$ remain the same, i.e., $\mathcal{D}_t(\phi^\star(\mathbf{x}) \mid y) = \mathcal{D}_{t-1}(\phi^\star(\mathbf{x}) \mid y)$, for any $t \geq 1$ and $y \in \{1, \ldots, K\}$. Besides, the data distribution at each round $\mathcal{D}_t$ is sampled from a distribution $\mathcal{P}_{\text{all}}$, and there exists a universal distribution $\mathcal{D}_{\text{all}}$ such that $\mathbb{E}_{\mathcal{D}_t \sim \mathcal{P}_{\text{all}}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}[\ell] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{\text{all}}}[\ell]$ for any well-conditioned loss function ℓ.*

The GOLS assumption extends the traditional label shift assumption [3, 40] by considering the invariant representation $\phi^\star$, which makes it applicable to various real-world scenarios, such as wild animal recognition [4] and recommendation systems [26], where the representations given the label are stable. Unlike the adversarial case in [3], $\mathcal{D}_t$ is supposed to be stochastic and sampled from a universal distribution in our work. We remark this is reasonable for many real-world applications, where distribution is not adversarially changed but follows certain patterns such as periodic changes in location or time, e.g., the location changes of cameras or the seasonal variations in product sales. The learner aims to learn a sequence of representations $\{\Phi_t\}_{t=1}^T$ and classifiers $\{\mathbf{w}_t\}_{t=1}^T$ that minimize the *cumulative expected risk*, i.e., the goal is to minimize

$$\sum_{t=1}^{T} R_t(\mathbf{w}_t \circ \Phi_t) - \sum_{t=1}^{T} R_t(\mathbf{w}_t^\star \circ \Phi^\star),$$

where $R_t(\mathbf{w} \circ \Phi) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}[\ell(\mathbf{w} \circ \varphi(\mathbf{x}; \Phi), y)]$ is the expected risk, $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ is the loss, and $\{\mathbf{w}_t^\star\}_{t=1}^T$ and $\Phi^\star$ are the optimal classifiers and representation, respectively.

4 Our Approach

In this section, we present our *Adapting to GOLS by Online Ensemble* (Agoe) approach. We first introduce a novel objective to effectively learn the representation. Then, we propose our algorithm to optimize the objective, which employs the online ensemble paradigm to conduct *multi-resolution updates*. Finally, we develop a mask-based updating mechanism to efficiently ensemble with DNNs.

Algorithm 1: Outer-Loop: Perturbed Descent

Input: Initial representation Φ_0
Initialize: $\Delta_1 = \mathbf{0}$;
for $t = 1, \ldots, T$ **do**
  Get the Δ_t from Algorithm 2;
  Update representation $\Phi_t = \Phi_{t-1} + \Delta_t$;
  Sample a scalar s_t uniformly from $[0, 1]$;
  Get the perturbed gradient $\widehat{\mathbf{g}}_t = \nabla \widehat{L}_t(\Phi_t + (s_t - 1)\Delta_t)$;
  Send $\widehat{\mathbf{g}}_t$ to Algorithm 2.
end

Algorithm 2: Inner-Loop: Online Ensemble

Initialize: Active base learners $\mathcal{A}_t = \emptyset$
for $t = 1, \ldots, T$ **do**
  Get the gradient $\widehat{\mathbf{g}}_t$ from Algorithm 1;
  Adjust the set of active base learners $\mathcal{A}_t$.
  for each base learner $\mathcal{E}_I \in \mathcal{A}_t$ **do**
    Update the base learner as in Eq. (4);
    Update the meta learner as in Eq. (5).
  end
  Send Δ_t to Algorithm 1.
end

4.1 Learning Invariant Representation in GOLS

In this section, we propose our approach for learning the representation in the GOLS problem.

A Novel Objective to Learn $\Phi^\star$. To learn the invariant representation, we propose a novel objective to capture the unique structure in GOLS. To start with, we observe that the representation $\Phi^\star$ should not only be shared and invariant across all the environments, but also "aligned" with the corresponding optimal classifier $\mathbf{w}_t^\star$ at each round t. Therefore, inspired by [1], our objective extends beyond simply considering the cumulative risk of $\sum_{t=1}^T R_t(\mathbf{w}_t \circ \Phi_t)$, and also includes a penalty term that regularizes the representation to be aligned with the classifier. Formally, our objective is

$$\min_{\{\Phi_t\}_{t=1}^T} \sum_{t=1}^T R_t(\mathbf{w}_t \circ \Phi_t) - \sum_{t=1}^T R_t(\mathbf{w}_t^\star \circ \Phi^\star),$$
$$\text{s.t.} \sum_{t=1}^T \left\| \nabla_{\mathbf{w}|\mathbf{w}=\mathbf{w}_t^\star} \ell(\mathbf{w} \circ \varphi(\mathbf{x}; \Phi_t), y) \right\|_2^2 = 0. \quad (1)$$

Note that optimizing the aforementioned objective with constraints is challenging, especially when employing non-convex, non-smooth representation models such as DNNs. Therefore, we employ the Lagrange multiplier method to construct the following surrogate loss function $L_t : \mathbb{R}^n \mapsto \mathbb{R}$:

$$L_t(\Phi) = R_t(\mathbf{w}_t \circ \Phi) + \lambda \cdot \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t} \left[\left\| \nabla_{\mathbf{w}|\mathbf{w}=\mathbf{w}_t^\star} \ell(\mathbf{w} \circ \varphi(\mathbf{x}; \Phi), y) \right\|_2^2 \right].$$

Thus, the goal is translated to get a representation sequence to minimize $\sum_{t=1}^T L_t(\Phi_t) - \sum_{t=1}^T L_t(\Phi^\star)$. However, learner can not observe expected function L_t, but only the empirical version $\widehat{L}_t$:

$$\widehat{L}_t(\Phi) = \widehat{R}_t(\mathbf{w}_t \circ \Phi) + \lambda \cdot \sum_{(\mathbf{x},y)\in S_t} \left[\left\| \nabla_{\mathbf{w}|\mathbf{w}=\mathbf{w}_t} \ell(\mathbf{w} \circ \varphi(\mathbf{x}; \Phi), y) \right\|_2^2 \right], \quad (2)$$

where $\widehat{R}_t(\mathbf{w} \circ \Phi) = \sum_{(\mathbf{x},y)\in S_t} [\ell(\mathbf{w} \circ \varphi(\mathbf{x}; \Phi), y)]$ is the empirical risk, $\mathbf{w}_t$ is the estimated classifier at round t. The objective in Eq. (2) exhibits several benign properties, such as unbiasedness and consistency, which will be formally proved in Section 5. These properties are essential for learning invariant information in GOLS.

Remark 1. Our contribution lies in carefully exploiting the unique structure in GOLS to design a novel objective to learn the representation, as in Eq. (2). This objective is proved to be unbiased to the expected loss L_t, which is crucial for the convergence analysis of our algorithm. Finally, we remark that this objective can be efficiently optimized in an online manner, as shown in Section 4.2.

Learning the Classifier $\mathbf{w}_t$. To learn the current classifier corresponding to the distribution $\mathcal{D}_t(y)$, we tend to exploit the structure in GOLS problem, i.e., $\mathcal{D}_0(\phi^\star(\mathbf{x}) \,|\, y) = \mathcal{D}_t(\phi^\star(\mathbf{x}) \,|\, y), \forall t \in [T]$ and $y \in \{1, \ldots, K\}$. Similar to [2] and [29], we employ the class prior distribution to reweight the initial offline classifier $\mathbf{w}_0$ to get the prediction. Formally, the classifier is updated as

$$[\mathbf{w}_t \circ \phi_t(\mathbf{x})]_j = \frac{1}{Z(\mathbf{x})} \frac{[\widehat{\boldsymbol{\mu}}_t]_j}{\mathcal{D}_0(y=j)} [\mathbf{w}_0 \circ \phi_t(\mathbf{x})]_j, \ \forall j \in [K], \quad (3)$$

where $Z(\mathbf{x}) = \sum_{j=1}^K \frac{[\widehat{\boldsymbol{\mu}}_t]_j}{\mathcal{D}_0(y=j)} [\mathbf{w}_0 \circ \phi_t(\mathbf{x})]_j$ is the normalization factor, $\widehat{\boldsymbol{\mu}}_t \in [0, 1]^K$ is the estimated label distribution using S_t, and $\phi_t(\mathbf{x}) = \varphi(\mathbf{x}; \Phi_t)$ is the current round's representation function. Therefore, the current classifier $\mathbf{w}_t$ is estimated by exploring the unique structure of generalized online label shift problem, utilizing both the representation and combining offline and online data.

4.2 Optimizing Objective in an Online Manner

This section illustrates how we optimize the objective in Eq. (2) by designing *multi-resolution updates*. Specifically, our approach contains (i) an outer-loop algorithm, and (ii) an inner-loop algorithm.

Outer-loop Algorithm. We observe that our objective exhibits favorable properties, including unbiasedness $\mathbb{E}[\widehat{L}_t] = L_t$, and bounded magnitude $|\widehat{L}_t| \leq D + \lambda G^2$, where D is the upper bound of the loss function ℓ's absolute value and G is the upper bound of the loss gradient's norm. Consequently, it conforms to the form of a non-convex, non-smooth optimization problem, particularly when using modern DNN architectures with ReLU activation functions or max-pooling operators, in which the learner can only receive stochastic and unbiased feedback at each round.

Optimizing such a problem in an online manner poses a significant challenge [19]. In order to solve this problem more stably in an online manner, we draw inspiration from the insight of Cutkosky et al. [9] that the design of non-convex optimizers falls in the scope of online linear optimization, which is a well-explored setting in online learning. Our algorithm consists of two components: an *inner-loop* algorithm and an *outer-loop* algorithm. At each round t, the inner-loop algorithm obtains the descent value Δ_t, which determines the optimization direction for the representation. The outer loop then updates the representation using Δ_t, and applies a small perturbation to the gradient which is used as the feedback for the inner-loop. The motivation of the outer-loop algorithm is that the perturbation enhances *algorithmic stability* and improves optimization in non-convex, non-smooth scenarios. Such an idea

of employing randomly perturbed gradients has been seen in the optimization literature [5, 21, 25]. The outer-loop algorithm is summarized as Algorithm 1. In the following, we detail how to get the update value Δ_t by designing the inner-loop algorithm.

Inner-loop Algorithm. The inner-loop algorithm, which selects the descent value Δ_t, is cast as an online learning problem. If the problem is easy, a simple online learning algorithm such as online gradient descent (OGD) [47] with specially tunned hyperparameters is sufficient [9]. However, when the problem is challenging and complex, hyperparameters of the algorithm are hard to determine. To this end, we employ the *online ensemble* paradigm [45] to learn a more stable descent value as described in Algorithm 2. Specifically, we maintain multiple base learners, each utilizing a different length of historical data to update the descent value. A meta learner is then used to combine the predictions of these base learners, which indicates the importance of different historical information. Together, we perform *multi-resolution updates* using various historical data windows. In the following, we detail the (i) base learner and (ii) meta learner in the inner-loop algorithm.

(i) Base Learner. Our inner-loop algorithm maintains multiple base learners $\mathcal{E}_I \in \mathcal{A}_t$, each running over different intervals to exploit different lengths of historical information, where $\mathcal{A}_t$ is the set of active base learners whose interval I contains t. We employ coin-betting-based algorithm [27] for updating the descent value. For the base learner $\mathcal{E}_I$ running on the interval $I = [s_i, e_i]$, the corresponding update value is

$$\Delta_t^I = -\frac{\sum_{\tau=s_i}^{t-1} \widehat{\mathbf{g}}_\tau}{t - s_i} \left(1 - \sum_{\tau=s_i}^{t-1} \widehat{\mathbf{g}}_\tau^\top \Delta_\tau^I\right), \tag{4}$$

where Δ_t^I is the descent value learned by base learner $\mathcal{E}_I$ of the inner-loop algorithm, the perturbed gradient $\widehat{\mathbf{g}}_t = \nabla \widehat{L}_t(\Phi_t + (s_t - 1)\Delta_t)$ is generated by the outer-loop Algorithm 1. We schedule base learners based on geometric covering (GC) intervals [10]. Note that the number of active base learners is at most $\lceil \log T \rceil$ at each round.

(ii) Meta Learner. The meta learner combines the predictions of base learners to get the final output, through a weighted combination way. Specifically, at round t, meta learner assigns a weight p_t^I to each active base learner $\mathcal{E}_I \in \mathcal{A}_t$. We update weights for active base learners based on AdaNormalHedge [23]. For each learner $\mathcal{E}_I$, we maintain a "potential function" ψ, and a "reusability function" ω with respect to this potential. Subsequently, the meta learner aggregates outputs of active base learners via weighted sum:

$$\begin{aligned} p_t^I &= \frac{\omega(R_{t-1}^I, C_{t-1}^I)}{\sum_{\mathcal{E}_I \in \mathcal{A}_t} \omega(R_{t-1}^I, C_{t-1}^I)}, \text{ for all } \mathcal{E}_I \in \mathcal{A}_t, \\ \text{and the final output } \Delta_t &= \sum_{\mathcal{E}_I \in \mathcal{A}_t} p_t^I \Delta_t^I. \end{aligned} \tag{5}$$

Here, the meta learner assigns weights to each base learner based on their historical performances and combines various lengths of historical data to produce the final output. We use $\psi(R, C) = \exp([R]_+^2/3C)$ as the potential function, where $[x]_+ \triangleq \max(0, x)$ and $\psi(0, 0)$ is defined to be 1. And the weight function ω is defined w.r.t. this potential $\omega(R, C) \triangleq \frac{1}{2}(\psi(R+1, C+1) - \psi(R-1, C+1))$. Besides, in Eq. (5), $R_{t-1}^I = \sum_{\tau=i}^{t-1} \langle \widehat{\mathbf{g}}_\tau, \Delta_\tau \rangle - \langle \widehat{\mathbf{g}}_\tau, \Delta_\tau^I \rangle$, and $C_{t-1}^I = \sum_{\tau=i}^{t-1} |\langle \widehat{\mathbf{g}}_\tau, \Delta_\tau \rangle - \langle \widehat{\mathbf{g}}_\tau, \Delta_\tau^I \rangle|$, Theoretically, we remark that by employing multi-resolution updates, our algorithm improves upon previous work [9] that our algorithm does not require to know the quality of the initial point in advance, and we will demonstrate this contribution in detail in Remark 4.

4.3 Efficient Mask-based Updating for Ensemble

As it can be observed in Eq.(5), the online ensemble paradigm necessitates maintaining approximately $\log T$ base learners. This requirement may become computationally and storage expensive, especially when employing DNNs as base learners to learn the representation Φ_t where we have to maintain a total of $O(\log T)$ DNN architectures at a time, which results in significant computational and storage burdens, making the algorithm impractical.

To alleviate this computational and storage burden, inspired by the theoretical results, we propose an efficient *masked-based* DNN update mechanism [24]. Specifically, we divide the network into multiple sub-parts, each functioning as an independent base learner. We randomly generate a mask sequence $\{M_i\}_{i=1}^{\lceil \log T \rceil}$ of length $\lceil \log T \rceil$, where $M_i \in \{0, 1\}^n$ for each $i \in [\lceil \log T \rceil]$, such that

$$\sum_{i=1}^{\lceil \log T \rceil} \|M_i\|_1 = n, \text{ and } M_i^\top M_j = 0, \forall i, j \in [\lceil \log T \rceil], i \neq j, \tag{6}$$

where n is the dimension of representation Φ. Eq. (6) indicates that the combination of all masks is equal to the total number of parameters in DNN, and masks are orthogonal and not overlap to each other. The base learner i at round t is $\Phi_t \odot M_i$, where $\odot$ indicates the element-wise multiplication. In practice, we divide the neural network into a maximum of 5 parts.

Once one base leaner $\mathcal{E}_I$ is removed from $\mathcal{A}_t$ at round t, the corresponding mask is set to zero to prevent updating of this part. Conversely, if one base learner is added to $\mathcal{A}_t$, we will set the corresponding mask to one, thereby resuming to update this part of the DNN. Then, we update each part of the DNN using different lengths of historical information to better learn a stable representation, with the help of multi-resolution updates. Finally, the weight p_t^i for $i \in [\lceil \log T \rceil]$, generated by the meta learner as in Eq. (5) by the training loss of each part, is used to determine learning rates of each part of the network, which reflects the importance of different historical information. Therefore, we update the network by

$$\Phi_t = \sum_{i=1}^{\lceil \log T \rceil} \left(\Phi_{t-1} \odot M_i + p_t^i \cdot \Delta_t \odot M_i\right).$$

Efficiency Discussion. We remark that the computational and storage efficiency of our algorithm is comparable to that of the vanilla OGD. Specifically, when a mask M_i is employed, only the parameters in the corresponding sub-network are updated, while the other parameters remain unchanged and thus do not require gradient calculation. Consequently, the computational complexity of our algorithm is similar to that of vanilla OGD. Furthermore, the storage cost is also comparable to that of vanilla OGD, as we only need to store the parameters of one DNN along with the masks, which is $O(n)$, where n is the dimension of representation Φ.

To summarize, we implement the online ensemble paradigm by updating each part of the DNN with different scopes of historical

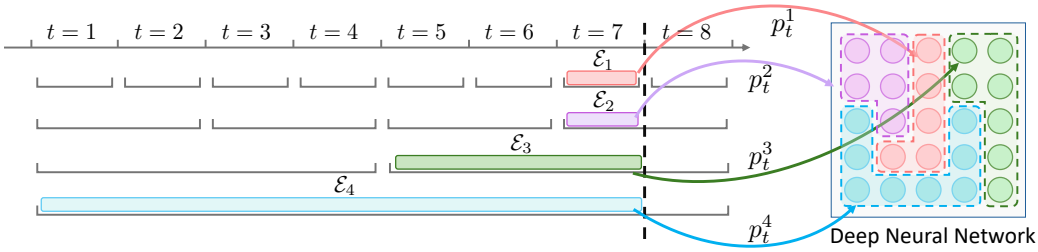


Figure 2: Our masked-based update mechanism for efficiently online ensemble with DNNs, where we segment the network into multiple sub-parts, each treated as a base learner $\mathcal{E}_I$ and updated using different lengths of historical data. Then, the weight of meta learner is employed to determine learning rate of each part. Together, we exploit multiple lengths of historical data windows to perform *multi-resolution updates*.

data. With the help of the meta learner, the weight p_t^i adaptively adjusts the learning rates according to the importance of the historical data, thereby learning a stable representation with multi-resolution updates. This mechanism substantially reduces the complexity of the online ensemble paradigm for DNNs, making it practical for real-world tasks. The whole process is illustrated in Figure 2.

Remark 2. Compared with previous mask-based or pruning-based updating methods [17, 24], we utilize the ensemble paradigm which uses different lengths of historical data to update each sub-part of the network, and the weights in meta learner p_t^i is used to determine the importance of different scopes of history. Therefore, our mechanism performs *multi-resolution updates* to exploit the temporal structure of data streams, such as periodic changes or recurring shifts, under the guidance of the online ensemble paradigm. Besides, our mechanism is *general* and can be applied to various scenarios, including computer vision and natural language processing, as demonstrated in experimental results in Section 6.2.

5 Theoretical Guarantees

In this section, we provide the theoretical guarantees. We first show that our objective $\widehat{L}_t$ is unbiased.

Proposition 1 (Unbiasedness)**.** Our designed estimator $\widehat{L}_t$ is an *unbiased estimation* of L_t, for all $\Phi \in \mathbb{R}^n$:

$$\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_t}\left[\widehat{L}_t(\Phi)\right] = L_t(\Phi).$$

Then, we prove that our Algorithm 2 can achieve the following strongly adaptive regret guarantee.

Theorem 1 (Parameter-free Strongly Adaptive Regret). *For any interval $\mathcal{I} = [s,e] \subseteq \mathbb{N}$ and any comparator $\mathbf{u} \in \mathbb{R}^n$, our Algorithm 2 satisfies*

$$\mathbb{E}\left[\mathbf{Reg}_T^{[s,e]}(\{\Delta_t\}_{t=1}^T)\right] \triangleq \mathbb{E}\left[\sum_{t=s}^{e}\langle \mathbf{g}_t, \Delta_t\rangle\right] - \sum_{t=s}^{e}\langle \mathbf{g}_t, \mathbf{u}\rangle \le \widetilde{O}\left(\|\mathbf{u}\|\sqrt{|\mathcal{I}|}\right),$$

where $\mathbf{g}_t \triangleq \nabla L_t(\Phi_t + (s_t - 1)\Delta_t)$ is the perturbed gradient, $\widetilde{O}(\cdot)$ ignores the logarithmic factors in T, $\|\mathbf{u}\|$ is the norm of the comparator, and $|\mathcal{I}| = e - s + 1$ is the length of the interval.

Then, we introduce the notion of (δ,ε)-stationary point, a concept that is commonly used in the field of optimization [9, 11, 42].

Definition 1 (Stationary Point)**.** A point $\mathbf{x}$ is a (δ,ε)-stationary point of an almost-everywhere differentiable function L if there is a finite subset $\mathcal{S}$ of the ball of radius δ centered at $\mathbf{x}$ such that for $\mathbf{y}$ selected uniformly from $\mathcal{S}$, $\mathbb{E}[\mathbf{y}] = \mathbf{x}$ and $\|\mathbb{E}[\nabla L(\mathbf{y})]\| \le \varepsilon$.

Definition 1 means that if our algorithm finds a (δ,ε)-stationary point, the gradient of that solution will be minimized, suggesting that our algorithm converges and finds the invariant representation. We then show our algorithm achieves an *optimal* convergence rate.

Corollary 1 (Convergence to the Stationary Point). *For a non-convex and non-smooth function L_t, $\forall t \in [T]$, under Assumption 1, for all $\delta > 0$, Algorithm 1 guarantees that:*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\left\|\nabla\bar{L}_t(\tilde{\Phi}_t)\right\|_\delta\right] \le \frac{2\gamma}{\delta T} + \max\left(\frac{5G^{2/3}\gamma^{1/3}}{(T\delta)^{1/3}}, \frac{6G}{\sqrt{T}}\right),$$

where $\bar{L}_t(\Phi) = \mathbb{E}_{\mathcal{D}_t\sim\mathcal{P}_{\text{all}}}[L_t(\Phi)]$, $\tilde{\Phi}_t$ is the representation randomly selected from $\{\Delta_t\}_{t=1}^T$ as formally defined in Appendix B.3, $\|\cdot\|_\delta$ is the δ-norm as defined in Definition 2, G is the upper bound of the gradient norm such that $\mathbb{E}[\|\mathbf{g}_t\|^2] \le G^2$, and $\gamma \triangleq \bar{L}_t(\Phi_0) - \bar{L}_t(\Phi^\star)$ is the quality of the initial point.

Remark 3 (Optimal Rate of Convergence)**.** Corollary 1 demonstrates that our algorithm can converge to the stationary point, thereby tracking the optimal representation at an *optimal* rate of $O(\gamma\delta^{-3}\varepsilon^{-1})$, which matches the lower bound established by Cutkosky et al. [9]. Notably, our proposed approach can handle non-convex, non-smooth functions, which broadens its applications to a wider range of scenarios, such as the ReLU activation functions or max-pooling operators in modern DNNs.

Remark 4 (Technical Contribution)**.** Compared with previous work of Cutkosky et al. [9], it needs to know the quality of the initial point and the optimal point $\bar{L}_t(\Phi_0) - \bar{L}_t(\Phi^\star)$, which is hard to estimate in practice. In contrast, our approach can handle it in a *parameter-free* way, i.e., we do not need to know the quality of the initial representation in advance. Such a property is achieved by our designed ensemble-based algorithm, which can adaptively learn the representation based on multi-resolution updates and is more flexible to any initial point cases, thereby learning a more stable representation. Besides, we extend the regret analysis of [27] and [23] from full-information to the unbiased stochastic scenario.

6 Experiment

In this section, we present empirical evaluations, including experiments on five benchmark datasets and two real-world tasks related to the GOLS problem. We aim to answer the following questions:

Table 1: Average error (%) of different algorithms on benchmark datasets. We report the mean and standard deviation over five runs. The best are emphasized in bold. The online sample size is set as $|S_t| = 128$, $\forall t \in [T]$.

	FIX	A-GEM	HAL	IWDAN	SSL	LAME	ODC	AGOE	Skyline
					Linear Shift				
CIFAR10	20.34	17.32 ± 0.15	18.25 ± 0.45	16.75 ± 0.12	16.68 ± 0.14	18.01 ± 0.09	17.92 ± 0.15	**16.52 ± 0.19**	8.32
CINIC10	33.15	28.55 ± 0.12	32.42 ± 2.55	26.44 ± 0.21	28.21 ± 0.11	31.23 ± 0.17	29.44 ± 0.23	**26.11 ± 0.23**	15.23
EuroSAT	16.32	11.35 ± 0.12	10.01 ± 3.17	7.21 ± 0.13	7.25 ± 0.12	13.72 ± 0.11	9.13 ± 0.15	**7.18 ± 0.34**	9.98
Fashion	12.98	**7.84 ± 0.08**	8.15 ± 0.05	8.39 ± 0.09	8.37 ± 0.08	11.32 ± 0.09	7.99 ± 0.06	8.42 ± 0.76	3.43
MNIST	1.75	1.25 ± 0.03	1.32 ± 0.04	1.07 ± 0.05	1.13 ± 0.03	**1.02 ± 0.03**	1.13 ± 0.03	1.09 ± 0.03	0.52
					Square Shift				
	FIX	**A-GEM**	**HAL**	**IWDAN**	**SSL**	**LAME**	**ODC**	**AGOE**	**Skyline**
CIFAR10	21.98	17.24 ± 0.15	17.35 ± 0.24	16.92 ± 0.16	17.82 ± 0.19	19.72 ± 0.13	**16.89 ± 0.13**	17.02 ± 0.33	8.74
CINIC10	34.14	27.15 ± 0.13	30.42 ± 2.25	26.34 ± 0.23	28.92 ± 0.13	30.52 ± 0.15	29.82 ± 0.15	**26.32 ± 0.29**	15.01
EuroSAT	16.03	11.01 ± 0.96	9.34 ± 3.15	**7.22 ± 0.14**	7.94 ± 0.15	13.77 ± 0.21	10.01 ± 0.16	7.49 ± 0.38	10.23
Fashion	12.65	8.92 ± 0.08	9.32 ± 0.13	8.15 ± 0.13	8.92 ± 0.13	12.41 ± 0.17	8.72 ± 0.10	**8.03 ± 0.76**	3.87
MNIST	1.65	1.13 ± 0.02	1.15 ± 0.05	1.05 ± 0.03	1.37 ± 0.05	1.31 ± 0.02	1.19 ± 0.02	**1.03 ± 0.03**	0.64
					Bernoulli Shift				
	FIX	**A-GEM**	**HAL**	**IWDAN**	**SSL**	**LAME**	**ODC**	**AGOE**	**Skyline**
CIFAR10	20.23	18.02 ± 0.52	18.01 ± 0.12	18.33 ± 0.17	17.91 ± 0.32	18.75 ± 0.23	19.34 ± 0.14	**17.13 ± 0.31**	9.85
CINIC10	33.59	27.32 ± 0.15	30.98 ± 1.98	26.77 ± 0.28	29.13 ± 0.21	29.76 ± 0.10	29.31 ± 0.72	**26.73 ± 0.25**	15.96
EuroSAT	15.67	9.03 ± 0.87	9.72 ± 2.01	8.41 ± 0.10	7.99 ± 0.19	14.42 ± 0.11	10.29 ± 0.31	**7.32 ± 0.37**	10.75
Fashion	12.62	9.43 ± 0.11	**8.05 ± 1.11**	8.65 ± 0.21	9.02 ± 0.18	12.06 ± 0.13	9.12 ± 0.07	8.77 ± 0.69	4.02
MNIST	1.83	1.24 ± 0.05	1.14 ± 0.13	1.07 ± 0.06	1.39 ± 0.06	1.42 ± 0.03	1.13 ± 0.03	**1.06 ± 0.05**	0.55
					Sine Shift				
	FIX	**A-GEM**	**HAL**	**IWDAN**	**SSL**	**LAME**	**ODC**	**AGOE**	**Skyline**
CIFAR10	22.03	17.82 ± 0.35	18.92 ± 0.15	18.42 ± 0.18	18.79 ± 0.45	21.64 ± 0.19	18.13 ± 0.14	**17.44 ± 0.43**	9.88
CINIC10	35.62	**27.59 ± 0.29**	31.12 ± 1.76	27.69 ± 0.98	30.22 ± 0.32	30.28 ± 0.23	29.02 ± 0.23	28.02 ± 0.31	16.00
EuroSAT	15.76	11.21 ± 0.77	9.98 ± 1.80	8.78 ± 0.32	8.85 ± 0.31	11.23 ± 0.19	9.13 ± 0.09	**8.15 ± 0.30**	10.69
Fashion	13.79	9.15 ± 0.12	8.51 ± 1.15	9.01 ± 0.52	9.70 ± 0.16	11.86 ± 0.08	9.58 ± 0.12	**8.48 ± 0.25**	4.23
MNIST	1.92	1.34 ± 0.03	1.30 ± 0.09	1.19 ± 0.03	1.24 ± 0.04	1.18 ± 0.05	1.31 ± 0.03	**1.15 ± 0.06**	0.59

- **Q1.** Does AGOE outperform other contenders in GOLS with various types of shifts?
- **Q2.** Does AGOE show effectiveness in real-world tasks with the generalized online label shift?
- **Q3.** Does AGOE correctly learn the underlying invariant representation $\Phi^\star$ as time evolves?

6.1 Benchmark Datasets

In this section, we seek to answer **Q1**. We compare our AGOE with eight contenders using five benchmark datasets. The competing methods contain a baseline method (*FIX*), two continual learning methods (*A-GEM* [8] and *HAL* [7]), a domain adaptation method (*IWDAN* [12]), a self-supervised learning method for updating the representation (*SSL* [39]), a sequential representation learning method (*ODC* [41]), a test-time adaptation method (*LAME* [6]), and a method that trains the representation on all data (*Skyline*). We defer the details of the contenders in Appendix A.

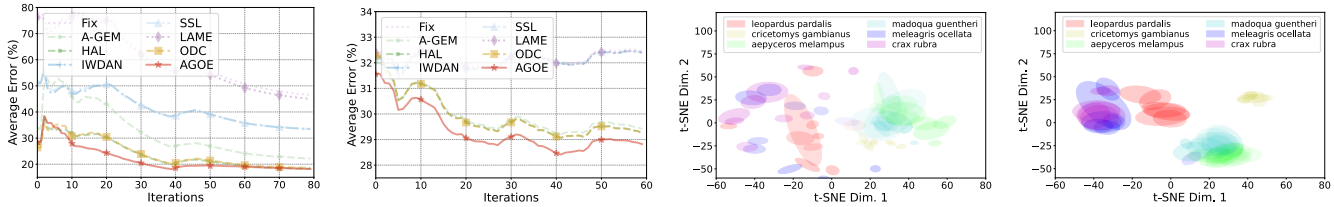
Implementation Details. In this part, we provide implementation details of the experiments. For the five benchmark datasets, we utilize a finetuned MobileNetV2 [31] to extract image features. The images used to train the MobileNetV2 do not overlap with either the offline or online datasets. The benchmark datasets' images are cropped and resized to 32×32.

For all the benchmark datasets in the generalized online label shift scenario, we simulate four types of environmental change patterns to encompass various non-stationary environments. For each case, the current distribution at round t is a mixture of two different stable distributions, i.e., $\boldsymbol{\mu}, \boldsymbol{\mu}' \in [0,1]^K$ with a time-varying coefficient α_t, i.e., $\boldsymbol{\mu}_t = (1-\alpha_t)\boldsymbol{\mu} + \alpha_t \boldsymbol{\mu}'$, where $\boldsymbol{\mu}_t$ denotes the current distribution at round t and α_t controls the non-stationarity and patterns. Specifically,

- `Linear Shift`: $\alpha_t = t/T$, simulating a linear change pattern.
- `Square Shift`: α_t switches between 0 and 1 following a quadratic pattern $\alpha_t = \sqrt{t/T}$.
- `Bernoulli Shift`: α_t randomly switches between 1 and 0 following the pattern $\alpha_t \sim b(\alpha)$, where b is a binomial distribution which returns 0 and 1 with the same probability $1/2$.
- `Sine Shift`: $\alpha_t = |\sin(t\pi/C)|$, simulating a sinusoidal change with a period of C.

We repeat all experiments with the same five random seeds and then evaluate the average error and the standard deviation.

Results on Benchmark Datasets. The comparison results on benchmark datasets are reported in Table 1. These results demonstrate that our proposed algorithm, AGOE, effectively adapts to the GOLS problem, outperforming other contenders. The baseline *FIX* is inferior to the other algorithms, highlighting the necessity of updating the representation in the changing environments. Our approach surpasses both continual learning methods (*A-GEM* and



(a) err curve of iWildCam (b) err curve of Amazon (c) $\mathcal{D}(\phi_{t=40}(\mathbf{x}) \mid y)$ (d) $\mathcal{D}(\phi_{t=80}(\mathbf{x}) \mid y)$

Figure 3: Results on real-world datasets. (a) & (b): The timely performances of average error rate on the iWildCam and Amazon datasets. (c) & (d) The t-SNE visualization of the conditional distribution of the iWildCam dataset based on the representation function Φ_t learned by AGOE at round $t = 40$ and $t = 80$. The transparency indicates the timestamp of that cluster.

Table 2: Average error (%) of different algorithms on the real-world applications of iWildCam [4] and Amazon [26] datasets. The performance metrics reported include both the mean accuracy and the standard deviation of different algorithms over a total of five separate runs. The best are emphasized in bold.

		Fix	A-GEM	HAL	IWDAN	SSL	LAME	ODC	AGOE	Skyline
iWildCam	Error (%)	46.51 ± 0.000	22.20 ± 0.754	18.36 ± 0.893	33.50 ± 0.180	33.54 ± 0.213	45.02 ± 0.032	18.67 ± 0.718	**18.16** ± 0.543	8.81 ± 0.000
	Efficinecy (s/batch)	0.540 ± 0.032	21.57 ± 0.286	24.24 ± 0.690	41.67 ± 0.870	36.04 ± 0.382	0.836 ± 0.045	22.87 ± 0.397	32.21 ± 2.060	—
	Energy (kJ/batch)	0.038 ± 0.003	2.746 ± 1.031	3.413 ± 1.219	6.061 ± 1.848	5.764 ± 1.602	0.061 ± 0.004	3.978 ± 1.208	4.776 ± 1.296	—
Amazon	Error (%)	32.47 ± 0.000	29.41 ± 0.065	29.26 ± 0.051	32.39 ± 0.057	32.41 ± 0.054	32.40 ± 0.003	29.26 ± 0.050	**28.82** ± 0.050	6.53 ± 0.000
	Efficinecy (s/batch)	22.43 ± 0.376	89.23 ± 0.879	113.1 ± 0.360	174.6 ± 0.541	172.0 ± 0.820	24.84 ± 0.530	123.5 ± 0.432	166.5 ± 16.04	—
	Energy (kJ/batch)	4.713 ± 1.908	22.91 ± 7.709	25.57 ± 5.860	48.37 ± 11.30	48.40 ± 10.03	5.235 ± 2.407	28.13 ± 6.391	46.65 ± 10.04	—

HAL) and the sequential representation learning method (*ODC*), indicating that, compared with methods that handle the general case of distribution shifts, our approach successfully mines the unique structure of GOLS, thereby achieving better performance. Besides, compared with *IWDAN* and *SSL*, which only utilize current unlabeled data for representation learning, our approach incorporates a novel learning objective and explores multi-resolution updates for learning representation. AGOE also outperforms the test-time-adaptation method *LAME*, demonstrating the necessity of updating the representation in GOLS. Therefore, we validate that AGOE is effective in handling the GOLS with various types of shifts. Notably, in the `Bernoulli` and `Sine` shift scenarios with periodic changes, our approach exhibits a more superior performance by leveraging multi-resolution updates to capture the unique structure in the generalized online label shift streams.

6.2 Real-world Applications

In this part, we aim to answer **Q2** and **Q3**. We compare the proposed approach with other contenders on two real-world applications: (i) a computer vision task of wild animal recognition on the iWildCam [4] dataset; and (ii) a natural language processing task of sentiment analysis on the Amazon [26] dataset.

Implementation Details. For the iWildCam dataset, we utilize a finetuned MobileNetV2 [31] to extract image features. The images used to train the MobileNetV2 do not overlap with either the offline or online datasets. The iWildCam dataset's images 96×96.

Table 3: Ablation study of our AGOE. `Penalty` is the penalty term in our optimization objective in 2. `Ensemble` represents the online ensemble to exploit multi-resolution updates.

ID	`Penalty`	`Ensemble`	iWildCam	Amazon
(i)	-	-	20.13 ± 0.60	29.26 ± 0.05
(ii)	√	-	19.35 ± 0.51	29.11 ± 0.06
(iii)	-	√	19.03 ± 0.45	28.97 ± 0.04
AGOE	√	√	**18.16 ± 0.54**	**28.82 ± 0.05**

For the Amazon dataset, we utilize a finetuned BERT-Mini [35] to extract text features. The corpus used to train the BERT-Mini does not overlap with either the offline or online datasets. The Amazon dataset's texts are trimmed and padded to a token length of 128.

Results on Real-world Datasets. To answer **Q2**, we report the average error of various algorithms on the iWildCam and Amazon datasets in Table 2, along with their timely performance illustrated in Figures 3 (a) and (b). Our proposed approach exhibits better performance compared to the continual learning methods (*A-GEM* and *HAL*), demonstrating that our penalty and ensemble paradigm can effectively handle the GOLS problem. The proposed method also surpasses the sequential representation learning method (*ODC*) that is designed for general distribution shift, indicating that our approach successfully mines the special GOLS structure by the multi-resolution updating strategy. Additionally, compared with *IWDAN* and *SSL*, our approach incorporates a specially designed

Table 4: Sensitivity analysis of the hyperparameter in AGOE. We report the average error (%) of AGOE with different λ values on iWildCam and Amazon datasets.

Dataset	$\lambda = 0.05$	$\lambda = 0.10$	$\lambda = 0.20$	$\lambda = 1.00$
iWildCam	18.53 ± 1.44	18.16 ± 0.54	18.50 ± 0.70	18.62 ± 1.39
Amazon	28.80 ± 0.03	28.82 ± 0.05	28.80 ± 0.04	28.81 ± 0.04

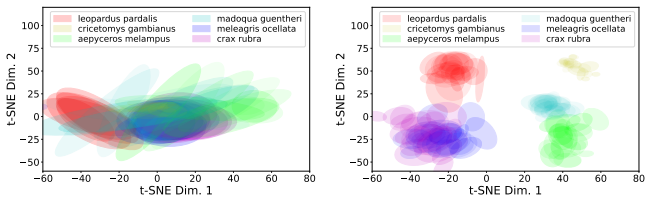
learning objective and explores different lengths of historical information for better representation learning. Our AGOE also outperforms the test-time adaptation method *LAME*, which adjusts the outputs of the model for adaptation, validating the importance of updating the representation in the GOLS data streams. The *Skyline*, which utilizes all the data to learn the representation, performs well, indicating that the generalized online label shift assumption of an invariant representation does hold in many real-world applications.

Effectiveness of Learning the Representation. To answer **Q3**, we visualize the learned representation Φ_t produced by our AGOE at different rounds t in Figure 3 (c) and (d). Using t-SNE [36], we visualize $\Phi_t(\mathbf{x})$ given certain labels at rounds $t = 40$ and $t = 80$. The results demonstrate that by our designed objective with the mask-based updating mechanism for leveraging multiple lengths of historical data, AGOE accurately traces the invariant representation $\Phi^\star$ over time, therefore capturing the inherently stable information from the non-stationary GOLS data stream, and can achieve a better representation with the help of the multi-resolution updates.

Efficiency Comparison. We also compare the computational efficiency of our approach with those of the other contenders. As shown in Table 2, the test-time-adaptation method *LAME* is the most efficient as it does not update the representation, but it does not yield a good performance in our experiments. Though the methods of *IWDAN* and *SSL* exhibit slower speed, they accomplish superior performance. Our approach, albeit with a slight compromise on computational cost, attains the best performance among all. Additionally, we compare the energy consumption in Table 2, calculating the energy needed for each method to process one batch of data. Similarly, our approach achieves the best performance with a slight compromise on the energy consumption.

Ablation Study. To validate each component's contribution in AGOE, we evaluate three variants: (i) removing both the penalty term in Eq. (2) and online ensemble structure, (ii) removing only the online ensemble structure for multi-resolution updating, and (iii) keeping the mask-based online ensemble but removing the penalty term. All variants use identical hyperparameters for fair comparison. As shown in Table 3, the online ensemble structure substantially improves performance through effective exploitation of historical information via our mask-based updating mechanism. The penalty term in Eq. (2) also proves essential, as removing it degrades performance. These results validate that both components are crucial for AGOE to achieve superior performance.

Sensitivity Analysis. We also conduct experiments to investigate the sensitivity of the hyperparameter in our approach. In all the experiments, we choose $\lambda = 0.10$ by default without any tuning. We provide the sensitivity analysis to show that the value of λ will not significantly affect the performance, as demonstrated in Table 4.



(a) $\mathcal{D}(\mathbf{x} \mid y)$ of iWildCam (b) $\mathcal{D}(\phi^\star(\mathbf{x}) \mid y)$ of iWildCam

Figure 4: (a) The conditional distribution of the raw features $\mathbf{x}$ given the label y. (b) The conditional distribution of the optimal representation $\phi^\star(\mathbf{x})$ given the label y.

GOLS assumption vs. label shift assumption. Online label shift (OLS) is a very good starting point to study online distribution shift adaptation with provable guarantees. However, as mentioned in Section 1, OLS assumes that $\mathcal{D}(\mathbf{x} \mid y)$ is fixed across the time horizon, which may not hold in many tasks. To this end, GOLS problem is motivated by real-world applications. For example, for vision and natural language processing tasks, if one directly uses the raw feature $\mathbf{x}$ without extracting the representation, deploying OLS algorithms with such features will have an unfavorable performance. In contrast, we use DNN models such as ResNet and MobileNet to extract representations before classification. We have included a figure comparing the $\mathcal{D}(\mathbf{x} \mid y)$ and $\mathcal{D}(\phi(\mathbf{x}) \mid y)$ for the real-world iWildCam data, as illustrated in Figure 4. The results indicate that the conditional distribution of raw features $\mathbf{x}$ given the label y is unstable across different timestamps, whereas the conditional distribution of the representation $\phi(\mathbf{x})$ given the label y remains stable. Therefore, Assumption 1 is essential and applicable in many real-world applications.

7 Conclusion

In this paper, we investigate the problem of adapting to *generalized online label shift* (GOLS), a generalized variant of traditional online label shift, where the key is to learn an unknown *invariant representation* such that the conditional distribution remains the same across all the environments. To tackle the problem, we design a novel objective for learning the underlying invariant representation, and propose a new algorithm to optimize the objective, which leverages online ensemble paradigm to perform *multi-resolution updates* using various historical data windows, thereby enhancing the stability of representation learning, which is theoretically guaranteed to achieve the *optimal convergence rate*. We also introduce an *efficient* mask-based implementation for ensembling with DNNs in practice. Extensive experiments on benchmark and two real-world datasets, including computer vision and natural language processing tasks, further demonstrate the effectiveness of our proposal.

Acknowledgment

We acknowledge the funding provided Jiangsu Science Foundation Leading-edge Technology Program (BK20232003). Z.-H. Zhou is the corresponding author.

References

[1] Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant Risk Minimization. *ArXiv preprint* arXiv:1907.02893 (2019).

[2] Dheeraj Baby, Saurabh Garg, Tzu-Ching Yen, Sivaraman Balakrishnan, Zachary Chase Lipton, and Yu-Xiang Wang. 2023. Online Label Shift: Optimal Dynamic Regret meets Practical Algorithms. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*. 65703–65742.

[3] Yong Bai, Yu-Jie Zhang, Peng Zhao, Masashi Sugiyama, and Zhi-Hua Zhou. 2022. Adapting to Online Label Shift with Provable Guarantees. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*. 29960–29974.

[4] Sara Beery, Elijah Cole, and Arvi Gjoka. 2020. The iWildCam 2020 Competition Dataset. *ArXiv preprint* arXiv:2004.10340 (2020).

[5] Devansh Bisla, Jing Wang, and Anna Choromanska. 2022. Low-Pass Filtering SGD for Recovering Flat Optima in the Deep Learning Optimization Landscape. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 8299–8339.

[6] Malik Boudiaf, Romain Müller, Ismail Ben Ayed, and Luca Bertinetto. 2022. Parameter-free Online Test-time Adaptation. In *Proceedings of the 35th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 8334–8343.

[7] Arslan Chaudhry, Albert Gordo, Puneet K. Dokania, Philip H. S. Torr, and David Lopez-Paz. 2021. Using Hindsight to Anchor Past Knowledge in Continual Learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*. 6993–7001.

[8] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient Lifelong Learning with A-GEM. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*.

[9] Ashok Cutkosky, Harsh Mehta, and Francesco Orabona. 2023. Optimal Stochastic Non-smooth Non-convex Optimization through Online-to-Non-convex Conversion. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*. 6643–6670.

[10] Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. 2015. Strongly Adaptive Online Learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. 1405–1411.

[11] Damek Davis, Dmitriy Drusvyatskiy, Yin Tat Lee, Swati Padmanabhan, and Guanghao Ye. 2022. A gradient sampling method with complexity guarantees for Lipschitz functions in high and low dimensions. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*. 6692–6703.

[12] Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoffrey J. Gordon. 2020. Domain adaptation with conditional distribution matching and generalized label shift. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*. 19276–19289.

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. 1126–1135.

[14] Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. 2019. Online Meta-Learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*. 1920–1930.

[15] Lan-Zhe Guo, Zhi Zhou, and Yufeng Li. 2020. RECORD: Resource Constrained Semi-Supervised Learning under Distribution Shift. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 1636–1644.

[16] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2022. Meta-Learning in Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 9 (2022), 5149–5169.

[17] Haeyong Kang, Rusty John Lloyd Mina, Sultan Rizky Hikmawan Madjid, Jaehong Yoon, Mark Hasegawa-Johnson, Sung Ju Hwang, and Chang D. Yoo. 2022. Forget-free Continual Learning with Winning Subnetworks. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*. 10734–10750.

[18] Sudipta Kar, Giuseppe Castellucci, Simone Filice, Shervin Malmasi, and Oleg Rokhlenko. 2022. Preventing Catastrophic Forgetting in Continual Learning of New Natural Language Tasks. In *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 3137–3145.

[19] Guy Kornowski and Ohad Shamir. 2021. Oracle Complexity in Nonsmooth Nonconvex Optimization. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*. 324–334.

[20] Yejia Liu, Shijin Duan, Xiaolin Xu, and Shaolei Ren. 2023. MetaLDC: Meta Learning of Low-Dimensional Computing Classifiers for Fast On-Device Adaption. In *Proceedings of the 3rd TinyML Research Symposium*. 1–8.

[21] Yong Liu, Siqi Mai, Minhao Cheng, Xiangning Chen, Cho-Jui Hsieh, and Yang You. 2022. Random Sharpness-Aware Minimization. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*. 24543–24556.

[22] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient Episodic Memory for Continual Learning. In *Advances in Neural Information Processing Systems 30 (NIPS)*. 6467–6476.

[23] Haipeng Luo and Robert E. Schapire. 2015. Achieving All with No Parameters: AdaNormalHedge. In *Proceedings of The 28th Conference on Learning Theory (COLT)*. 1286–1304.

[24] Arun Mallya and Svetlana Lazebnik. 2018. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. In *Proceedings of the 31st IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7765–7773.

[25] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. 2017. Exploring Generalization in Deep Learning. In *Advances in Neural Information Processing Systems 30 (NIPS)*. 5947–5956.

[26] Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 188–197.

[27] Francesco Orabona and Dávid Pál. 2016. Coin Betting and Parameter-Free Online Learning. In *Advances in Neural Information Processing Systems 29 (NIPS)*. 577–585.

[28] Yu-Yang Qian, Yong Bai, Zhen-Yu Zhang, Peng Zhao, and Zhi-Hua Zhou. 2023. Handling New Class in Online Label Shift. In *Proceedings of the 23rd IEEE International Conference on Data Mining (ICDM)*. 1283–1288.

[29] Yu-Yang Qian, Peng Zhao, Yu-Jie Zhang, Masashi Sugiyama, and Zhi-Hua Zhou. 2024. Efficient Non-stationary Online Learning by Wavelets with Applications to Online Distribution Shift Adaptation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. to appear.

[30] Marco Saerens, Patrice Latinne, and Christine Decaestecker. 2002. Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure. *Neural Computation* 14, 1 (2002), 21–41.

[31] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the 33rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4510–4520.

[32] Jie-Jing Shao, Yunlu Xu, Zhanzhan Cheng, and Yufeng Li. 2022. Active Model Adaptation Under Unknown Shift. In *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 1558–1566.

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

[34] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*. 9229–9248.

[35] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *ArXiv preprint* arXiv:1908.08962 (2019).

[36] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 11 (2008), 2579–2605.

[37] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. 2022. Continual test-time domain adaptation. In *Proceedings of 35th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7191–7201.

[38] Song Wang, Zhen Tan, Huan Liu, and Jundong Li. 2023. Contrastive Meta-Learning for Few-shot Node Classification. In *Proceedings of the 29th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 2386–2397.

[39] Ruihan Wu, Siddhartha Datta, Yi Su, Dheeraj Baby, Yu-Xiang Wang, and Kilian Q Weinberger. 2023. Online Feature Updates Improve Online (Generalized) Label Shift Adaptation. *NeurIPS Workshop on Self-Supervised Learning: Theory and Practice* (2023).

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

[41] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. 2020. Online Deep Clustering for Unsupervised Representation Learning. In *Proceedings of the 33rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6687–6696.

[42] Jingzhao Zhang, Hongzhou Lin, Stefanie Jegelka, Suvrit Sra, and Ali Jadbabaie. 2020. Complexity of Finding Stationary Points of Nonconvex Nonsmooth Functions. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*. 11173–11182.

[43] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. 2013. Domain Adaptation under Target and Conditional Shift. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*. 819–827.

[44] Chen Zhao, Feng Chen, and Bhavani Thuraisingham. 2021. Fairness-Aware Online Meta-learning. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 2294–2304.

[45] Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. 2024. Adaptivity and Non-stationarity: Problem-dependent Dynamic Regret for Online Convex Optimization. *Journal of Machine Learning Research* 25, 98 (2024), 1–52.

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

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

A Contenders

In this section, we introduce the details of the contenders.

- *FIX*: The offline model is never updated in the online stage, serving as a baseline for other methods.
- *HAL* [7]: A continual learning method which maintains prototypes in both the input space and the representation space to regularize and stabilize online updates.
- *A-GEM* [8]: A continual learning method which projects the gradient on the latest online batch to a direction that does not conflict with the gradient on a batch sampled from observed data to contain catastrophic forgetting.
- *IWDAN* [12]: An unsupervised domain adaptation (UDA) method focusing on the generalized label shift problem. The representation model is forced to map the target domain data into the source domain data's feature space, so that the domain discriminator cannot distinguish between the feature of the target domain data and the feature of the source domain data.
- *SSL* [39]: A self-supervised learning (SSL) method designed for online representation update under a GOLS setting similar to the problem setting in our paper.
- *LAME* [6]: A test-time adaptation (TTA) method. The representation is kept to be fixed, and only adjust the prediction based on online data. Specifically, *LAME* forces the posterior probabilities of two online test points with similar representations to be closer.
- *ODC* [41]: A SSL method designed for sequential representation update. The cluster identity of an online data point is regarded as its pseudo label to update the representation and the classifier. For a fair comparison, we additionally update *ODC*'s model with a supervised cross-entropy loss.
- *Skyline*: Trained on all data including the offline data and the online data, *Skyline* is assumed to have the optimal underlying representation and classifier in hindsight.

B Proofs

This section provides the omitted proofs of Section 5.

B.1 Proof of Proposition 1

Proof. For any representation $\Phi \in \mathbb{R}^n$, by taking the expectation, we have

$$\begin{aligned}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}[\widehat{L}_t(\Phi)] &= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}\Bigg[\widehat{R}_t(\mathbf{w}_t\circ\Phi) \\ &\quad + \frac{\lambda}{2}\sum\nolimits_{(\mathbf{x},y)\in S_t}\left[\left\|\nabla_{\mathbf{w}|\mathbf{w}=\mathbf{w}_t}\ell(\mathbf{w}\circ\varphi(\mathbf{x};\Phi),y)\right\|_2^2\right]\Bigg] \\ &= R_t(\mathbf{w}_t\circ\Phi) + \frac{\lambda}{2}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}\left[\left\|\nabla_{\mathbf{w}|\mathbf{w}=\mathbf{w}_t^\star}\ell(\mathbf{w}\circ\varphi(\mathbf{x};\Phi),y)\right\|_2^2\right] = L_t(\Phi),\end{aligned}$$

where in the second equality, we assume that the current classifier can be accurately estimated by the representation function, the offline data, and the current online data by Eq. (3). This holds if the GOLS assumption in Assumption 1 is valid and if there is sufficient data to estimate the representation and the label distribution. Therefore, we finish the proof. □

B.2 Proof of Theorem 1

Proof. We first illustrate how to get the regret bound for any interval $[i,j] \in \mathcal{I}_{\mathrm{GC}}$, and then extend to any interval $[s,e]$.

For any interval $[i,j] \in \mathcal{I}_{\mathrm{GC}}$, for the base learner $\mathcal{E}_{\mathcal{I}} \in \mathcal{A}_t$ that runs on the interval $\mathcal{I} = [i,j]$, we decompose the regret into two parts: the meta regret and the base regret as following.

$$\begin{aligned}&\mathbb{E}\left[\mathbf{Reg}_T^{[i,j]}(\{\Delta_t\}_{t=1}^T)\right] \triangleq \mathbb{E}\left[\sum_{t=i}^{j}\langle\mathbf{g}_t,\Delta_t\rangle\right] - \sum_{t=i}^{j}\langle\mathbf{g}_t,\mathbf{u}\rangle \\ &= \underbrace{\mathbb{E}\left[\sum_{t=i}^{j}\langle\mathbf{g}_t,\Delta_t\rangle - \sum_{t=i}^{j}\langle\mathbf{g}_t,\Delta_t^{\mathcal{I}}\rangle\right]}_{\text{meta regret}} + \underbrace{\mathbb{E}\left[\sum_{t=i}^{j}\langle\mathbf{g}_t,\Delta_t^{\mathcal{I}}\rangle\right] - \sum_{t=i}^{j}\langle\mathbf{g}_t,\mathbf{u}\rangle}_{\text{base regret}}.\end{aligned}$$

(i) Base Regret. The base regret measures the gap between the base model and the comparator. To further examine the base regret, we decompose the base regret as

$$\underbrace{\mathbb{E}_{i:j}\left[\sum_{t=i}^{j}\langle\mathbf{g}_t-\widehat{\mathbf{g}}_t,\Delta_t-\mathbf{u}\rangle\right]}_{\texttt{term (a)}} + \underbrace{\mathbb{E}_{i:j}\left[\sum_{t=i}^{j}\langle\widehat{\mathbf{g}}_t,\Delta_t-\mathbf{u}\rangle\right]}_{\texttt{term (b)}},$$

where $\mathbb{E}_{i:j}[\cdot]$ denotes the expectation taken over the random draw of dataset $\{S_\tau\}_{\tau=i}^j$, and $\widehat{\mathbf{g}}_t = \nabla\widehat{L}_t(\Delta_t)$. For `term (a)`, we have

$$\texttt{term (a)} = \mathbb{E}_{i:t-1}\left[\langle\mathbf{g}_t - \mathbb{E}_t[\widehat{\mathbf{g}}_t \mid i:t-1], \Delta_t-\mathbf{u}\rangle\right] = 0,$$

where the last equality is due to the unbiasedness of the risk estimator $\widehat{L}_t$ as stated in Proposition 1, such that $\mathbf{g}_t = \mathbb{E}_t[\widehat{\mathbf{g}}_t \mid i:t-1]$. Thus, it is sufficient to analyze `term (b)` to provide an upper bound for `term (a)`. Without the loss of generality, we assume $\|\widehat{\mathbf{g}}_t\| \le 1$ and $\|\mathbf{g}_t\| \le 1$ for all $t \in [T]$, which can be achieved by scaling the objective by a constant factor. We introduce the following lemma:

Lemma 1 (Corollary 5 in [27]). *Let $\{\widehat{\mathbf{g}}_t\}_{t=1}^\infty$ be any sequence of gradient vectors such that $\|\widehat{\mathbf{g}}_t\| \le 1$, and the interval $\mathcal{I} = [i,j]$. Then, the regret of the base learner $\mathcal{E}_{\mathcal{I}}$ update as Eq. (4) satisfies*

$$\begin{aligned}\sum_{t=i}^{j}\langle\widehat{\mathbf{g}}_t,\Delta_t^{\mathcal{I}}-\mathbf{u}\rangle &\le \|\mathbf{u}\|\sqrt{|\mathcal{I}|\ln(1+24|\mathcal{I}|^2\|\mathbf{u}\|^2)} + \left(1-\frac{1}{\sqrt{\pi|\mathcal{I}|}}\right) \\ &= \widetilde{O}\left(\|\mathbf{u}\|\sqrt{|\mathcal{I}|}\right),\end{aligned}$$

where $|\mathcal{I}| = j-i+1$ is the length of the interval.

Combining upper bounds of `term (a)` and `term (b)` yields

$$\mathbb{E}\left[\sum_{t=i}^{j}\langle\mathbf{g}_t,\Delta_t^{\mathcal{I}}\rangle\right] - \sum_{t=i}^{j}\langle\mathbf{g}_t,\mathbf{u}\rangle \le \widetilde{O}\left(\|\mathbf{u}\|\sqrt{|\mathcal{I}|}\right),$$

where $\widetilde{O}(\cdot)$ ignores the logarithmic factors in T.

(ii) Meta Regret. For the meta regret, similarly, we decompose meta regret into two parts.

$$\underbrace{\mathbb{E}_{i:j}\left[\sum_{t=i}^{j}\langle\mathbf{g}_t-\widehat{\mathbf{g}}_t,\Delta_t-\Delta_t^{\mathcal{I}}\rangle\right]}_{\texttt{term (c)}} + \underbrace{\mathbb{E}_{i:j}\left[\sum_{t=i}^{j}\langle\widehat{\mathbf{g}}_t,\Delta_t-\Delta_t^{\mathcal{I}}\rangle\right]}_{\texttt{term (d)}},$$

For `term (c)`, we have

$$\begin{aligned}\texttt{term (c)} &= \mathbb{E}_{i:j}\left[\langle\mathbf{g}_t-\widehat{\mathbf{g}}_t,\Delta_t-\Delta_t^{\mathcal{I}}\rangle\right] \\ &= \mathbb{E}_{i:t-1}\left[\langle\mathbf{g}_t-\mathbb{E}_t[\widehat{\mathbf{g}}_t \mid i:t-1],\Delta_t-\Delta_t^{\mathcal{I}}\rangle\right] = 0,\end{aligned}$$

where the last equality is due to the unbiasedness of the risk estimator $\widehat{L}_t$ as stated in Proposition 1, such that $\mathbf{g}_t = \mathbb{E}_t[\widehat{\mathbf{g}}_t \mid i : t-1]$. To upper bound the term (d), we introduce the following lemma.

Lemma 2 (Theorem 1 in [23]). *Let $\{\widehat{\mathbf{g}}_t\}_{t=1}^{\infty}$ be any sequence of gradient vectors such that $\|\widehat{\mathbf{g}}_t\| \le 1$, then for any interval $\mathcal{I} = [i, j] \in \mathcal{I}_{\mathrm{GC}}$, the meta algorithm that runs as AdaNormalHedge satisfies*

$$\sum_{t=i}^{j} \langle \widehat{\mathbf{g}}_t, \Delta_t \rangle - \sum_{t=i}^{j} \langle \widehat{\mathbf{g}}_t, \Delta_t^{\mathcal{I}} \rangle \le \sqrt{3|\mathcal{I}|c(j)} = \widetilde{O}\left(\sqrt{|\mathcal{I}|}\right),$$

where $c(j) \le 1 + \ln j + \ln(1+\log_2 j) + \ln \frac{5+3\ln(1+j)}{2}$ is a logarithmic factor, in which $m(t)$ is the total number of base learners created up to round t, and we show that $m(t) \le t(1+\log_2 t)$ in Lemma 3.

Thus, by combing the base regret and the meta regret, we can prove the regret in the GC interval, i.e., for any interval $\mathcal{I} = [i, j] \in \mathcal{I}_{\mathrm{GC}}$,

$$\mathbb{E}\left[\sum_{t=i}^{j} \langle \mathbf{g}_t, \Delta_t \rangle\right] - \sum_{t=i}^{j} \langle \mathbf{g}_t, \mathbf{u} \rangle = \widetilde{O}\left(\|\mathbf{u}\|\sqrt{|\mathcal{I}|}\right). \tag{7}$$

(iii) Extend to Any Interval $[s, e] \subseteq \mathbb{N}$. In the above, we have proved the regret in GC intervals $[i, j] \in \mathcal{I}_{\mathrm{GC}}$ in Eq. (7). In this part, we extend the strongly adaptive regret to any interval $[s, e] \subseteq \mathbb{N}$. We first introduce the following lemma to describe the benign property of the GC intervals.

Lemma 3 (Lemma 1.2 of [10]). *For any interval $[s, e] \subseteq \mathbb{N}$, it can be partitioned into two sequences of disjoint and consecutive intervals, i.e., $\mathcal{I}_{-p}, \ldots, \mathcal{I}_0 \in \mathcal{I}_{\mathrm{GC}}$ and $\mathcal{I}_1, \ldots, \mathcal{I}_q \in \mathcal{I}_{\mathrm{GC}}$, such that $|\mathcal{I}_{-i}|/|\mathcal{I}_{-i+1}| \le 1/2, \forall i \ge 1$ and $|\mathcal{I}_i|/|\mathcal{I}_{i-1}| \le 1/2, \forall i \ge 2$. Let $m(t)$ be the total number of base learners created up to round t. Then,*

$$m(t) \le t\left(1 + \log_2 t\right),$$

because the active base learners in t-round is smaller than $1 + \log_2 t$.

Next, we extend the above strongly adaptive regret bound in Eq. (7) to any interval $\mathcal{I} = [s, e] \subseteq \mathbb{N}$ by utilizing Lemma 3. We first decompose the strongly adaptive regret over $\mathcal{I} = [s, e]$ as

$$\begin{aligned}
&\sum_{t=s}^{e} \langle \mathbf{g}_t, \Delta_t \rangle - \sum_{t=s}^{e} \langle \mathbf{g}_t, \mathbf{u} \rangle \\
&= \underbrace{\sum_{i=-p}^{0} \left(\sum_{t \in \mathcal{I}_i} \langle \mathbf{g}_t, \Delta_t \rangle - \sum_{t \in \mathcal{I}_i} \langle \mathbf{g}_t, \mathbf{u} \rangle \right)}_{\texttt{term (e)}} + \underbrace{\sum_{i=1}^{q} \left(\sum_{t \in \mathcal{I}_i} \langle \mathbf{g}_t, \Delta_t \rangle - \sum_{t \in \mathcal{I}_i} \langle \mathbf{g}_t, \mathbf{u} \rangle \right)}_{\texttt{term (f)}}.
\end{aligned}$$

Then, we bound term (e) based on the adaptive regret in Eq. (7),

$$\begin{aligned}
\texttt{term (e)} \le \sum_{i=-p}^{0} \Bigg(&\|\mathbf{u}\|\sqrt{|\mathcal{I}| \ln(1 + 24|\mathcal{I}|^2\|\mathbf{u}\|^2)} \\
&+ \left(1 - \frac{1}{\sqrt{\pi|\mathcal{I}|}}\right) + \sqrt{3|\mathcal{I}|c(e)} \Bigg) = \widetilde{O}\left(\|\mathbf{u}\|\sqrt{|\mathcal{I}|}\right),
\end{aligned}$$

Furthermore, the term (f) can be bounded in the same way. Therefore, we have

$$\sum_{t=s}^{e} \langle \mathbf{g}_t, \Delta_t \rangle - \sum_{t=s}^{e} \langle \mathbf{g}_t, \mathbf{u} \rangle = \widetilde{O}\left(\|\mathbf{u}\|\sqrt{|\mathcal{I}|}\right).$$

Thus, we complete the proof of Theorem 1. □

B.3 Proof of Corollary 1

Proof. We first provide the definition of δ-norm in Corollary 1.

Definition 2 (δ-norm)**.** Given a point $\mathbf{x}$, a number $\delta > 0$ and a almost-everywhere differentiable function F, define

$$\|\nabla F(\mathbf{x})\|_{\delta} \triangleq \inf_{S \subset B(\mathbf{x}, \delta), \frac{1}{|S|}\sum_{\mathbf{y} \in S} \mathbf{y} = \mathbf{x}} \left\| \frac{1}{|S|} \sum_{\mathbf{y} \in S} \nabla F(\mathbf{y}) \right\|.$$

Then, we introduce some preliminaries of non-convex, non-smooth optimization problems. To start with, Cutkosky et al. [9] address non-convex, non-smooth optimization using a "restart" mechanism. They divide the entire budget of gradient evaluations into multiple intervals, with the inner-loop algorithm restarting at each interval. Given the total number of evaluations as T, we define the restart cycle (i.e., length of each interval) as C, therefore, the number of restarts is $K = \lfloor T/C \rfloor$. We introduce following lemma.

Lemma 4 (Corollary 9 of Cutkosky et al. [9]). *Suppose we have a budget of T gradient evaluations, the function L is well-behaved, $\mathbb{E}[\|\mathbf{g}_t\|^2] \le G^2$, $\|\Delta_t\| \le D$ for some user-specified D, and that the inner-loop algorithm ensures the regret bound in every interval $\mathcal{I} = [kC+1, kC+C], \forall k \in [K]$ to satisfy $\mathbb{E}[\mathbf{Reg}_T^{[s,e]}(\{\Delta_t\}_{t=1}^T)] \le DGK\sqrt{|\mathcal{I}|}$ for all $\|\mathbf{u}^k\| \le D$. Let $\delta > 0$ be an arbitrary number. Set $D = \delta/C, C = \min(\lceil (GT\delta/\gamma)^{2/3} \rceil, T/2)$, and $K = \lfloor T/C \rfloor$. Then,*

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left\|\frac{1}{C}\sum_{t=kC+1}^{kC+C}\nabla \bar{L}_t(\Phi_t)\right\|\right] \le \frac{2\gamma}{\delta T} + \max\left(\frac{5G^{2/3}\gamma^{1/3}}{(T\delta)^{1/3}}, \frac{6G}{\sqrt{T}}\right),$$

where $\bar{L}_t(\Phi) = \mathbb{E}_{\mathcal{D}_t \sim \mathcal{P}_{\mathrm{all}}}[L_t(\Phi)]$, and $\gamma \triangleq \bar{L}_t(\Phi_0) - \bar{L}_t(\Phi^{\star})$ is the quality of the initial point.

To prove Corollary 1, we denote $\bar{\Phi}^k \triangleq \frac{1}{C}\sum_{t=kC+1}^{kC+C}\Phi_t$, thus we have $\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left\|\nabla \bar{L}_t(\bar{\Phi}^k)\right\|_{\delta}\right] \le \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left\|\frac{1}{C}\sum_{t=kC+1}^{kC+C}\nabla \bar{L}_t(\Phi_t)\right\|\right]$. If we choose $\bar{\Phi}_t$ randomly and uniformly from the sequence $\{\bar{\Phi}^k\}_{k=1}^K$,

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\left\|\nabla \bar{L}_t(\bar{\Phi}_t)\right\|_{\delta}\right] \le \frac{2\gamma}{\delta T} + \max\left(\frac{5G^{2/3}\gamma^{1/3}}{(T\delta)^{1/3}}, \frac{6G}{\sqrt{T}}\right).$$

Therefore, if one can access the quality of the initial point $\gamma \triangleq \bar{L}_t(\Phi_0) - \bar{L}_t(\Phi^{\star})$, using Lemma 4, taking a simple online gradient descent [47] as the inner-loop algorithm can achieve the optimal rate to find the stationary points in non-convex, non-smooth optimization problems. However, in practice, it is challenging to estimate the quality of the initial point in advance. Therefore, we need to find a way to handle this challenge in a *parameter-free* manner, i.e., even do not know the quality of the initial point.

To tackle this problem, as shown in Theorem 1, our approach enjoys the advantageous properties of being parameter-free and strongly adaptive. This means that regardless of the values of restart cycle C and the diameter D, our approach can handle these parameters by setting C and D to their optimal values only in the analysis without actually knowing them, achieving the same regret order as if these parameters were known beforehand. This is possible because our algorithm using the online ensemble paradigm is parameter-free which can adapt to more complex problems where parameters are unknown in advance, as detailed in Theorem 1. Therefore, we complete the proof of Corollary 1. □