Online Feature Updates Improve Online (Generalized) Label Shift Adaptation

Ruihan Wu[*][†]
UCSD
ruw076@ucsd.edu

Siddhartha Datta[*][†]
University of Oxford
siddhartha.datta@cs.ox.ac.uk

Yi Su
Google Deepmind
yisumtv@google.com

Dheeraj Baby
UCSB
dheeraj@cs.ucsb.edu

Yu-Xiang Wang
UCSB
yuxiangw@cs.ucsb.edu

Kilian Q. Weinberger
Cornell University
kilian@cornell.edu

Abstract

This paper addresses the prevalent issue of label shift in an online setting with missing labels, where data distributions change over time and obtaining timely labels is challenging. While existing methods primarily focus on adjusting or updating the final layer of a pre-trained classifier, we delve into the untapped potential of enhancing feature representations using unlabeled data at test-time. Our novel Online Label Shift adaptation with Online Feature Updates (OLS-OFU) method harnesses self-supervised learning to refine the feature extraction process, thus improving the prediction model. Theoretical analyses confirm that OLS-OFU reduces algorithmic regret by capitalizing on self-supervised learning for feature refinement. Empirical tests on CIFAR-10 and CIFAR-10C datasets, under both online label shift and generalized label shift conditions, underscore OLS-OFU's effectiveness and robustness, especially in cases of domain shifts.

1 Introduction

The effectiveness of most supervised learning models relies on a key assumption that the train data and test data share the same distribution. However, this assumption rarely holds in real-world scenarios, giving rise to *distribution shift*. Prior work focused on understanding distribution shifts in offline or batch settings, where a single shift occurs between the train and test distributions. In contrast, real-world applications often involve test data arriving in an *online* fashion, and the distribution shift can continuously evolve over time. There is another challenging issue of *missing and delayed* feedback labels, where gathering labels for the streaming data in a timely manner becomes a challenging task.

To tackle distribution shift, prior work makes further assumptions on the nature of the shift, such as label shift or covariate shift. Our work focuses on the common *(generalized) label shift* problem in an online fashion with missing labels. The learner is given a fixed set of labeled training data $D_0 \sim \mathcal{P}^{\text{train}}$ in advance and trains a model f_0. At test-time, only a small batch of unlabelled test data $S_t \sim \mathcal{P}_t^{\text{test}}$ arrives in an online fashion ($t = 1, 2, \cdots$). For online label shift, we assume the label distribution $\mathcal{P}_t^{\text{test}}(y)$ may change over time t while the conditional distribution stays the same, i.e. $\mathcal{P}_t^{\text{test}}(x|y) = \mathcal{P}^{\text{train}}(x|y)$. The *generalized* label shift relaxes this assumption by assuming there exists a transformation h of the covariate, such that the conditional distribution $\mathcal{P}_t^{\text{test}}(h(x)|y) = \mathcal{P}^{\text{train}}(h(x)|y)$ stays the same. A feature extractor h exists mapping these variations

[*] Equal contribution

[†] Work done while at Cornell University

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

to the same point in the latent space. The goal of the learner is to adapt to the (generalized) label shift within the non-stationary environment, continually adjusting the model's predictions in real time.

Existing online label shift adaptation algorithms (OLS) primarily adopt one of two strategies: either directly reweighting of the pretrained classifier f_0, or re-training only the final linear layer of f_0 — typically keeping the feature extractor frozen. Recent work has demonstrated that feature extractors can still be improved, even during test-time and in the absence of labels. We hypothesize that a similar effect can be leveraged in the (generalized) label shift setting and propose to improve the feature representation during testing. In online label shift, updating the feature extractor offers two possible advantages. First, it enables the utilization of additional unlabeled samples to enhance the sample efficiency of the feature extractor. Second, it allows the adaptation of the feature extractor to label shift. Latter is important because the optimal feature extractor is not necessarily independent of the label distribution. In particular, in generalized label shift, the feature transformation h is typically unknown, and additional unlabeled test samples can ease the learning of h.

This paper introduces the *Online Label Shift adaptation with Online Feature Updates* (OLS-OFU) framework, aimed at enhancing feature representation learning in the context of online label shift adaptation. Each instantiation of OLS-OFU incorporates a self-supervised learning method associated with a loss function denoted as l_{ssl} and an existing online label shift adaptation (OLS) algorithm. In each time step, it executes a modified version of OLS, updates the feature extractor through self-supervised learning, and subsequently refines the last linear prediction layer. Theoretically, we demonstrate that OLS-OFU reduces the loss of the overall algorithm by leveraging self-supervised learning techniques to enhance the feature extractor, thereby improving predictions for test samples at each time step t. Empirical evaluations on CIFAR-10 and CIFAR-10C datasets, considering both online label shift and online generalized label shift scenarios, affirm the effectiveness of OLS-OFU.

2 Problem Setting

We start with some basic notations. Let Δ^{K-1} be the probability simplex. Let $f : \mathcal{X} \to \Delta^{K-1}$ denote a classifier. Given an input x from domain $\mathcal{X}$, $f(x)$ outputs a probabilistic prediction on K classes. For example, f can be the output from the softmax operation after any neural network. If we reweight a model f by a vector $p \in \mathbb{R}^K$, we refer to this model as $g(\cdot; f, p)$ with g denotes the method of reweighting. For any two vectors p and q, p/q denotes the element-wise division.

Online distribution shift adaptation. When a well-trained model f_0 is deployed in the real world, it enters the test stage, which can be a sequence of time periods or time steps. The test distribution at time step t, $\mathcal{P}_t^{\text{test}}$, from which test data x_t is sampled, may vary over time. The challenge lies in how to adjust the model continuously from f_{t-1} to f_t in an online fashion to adapt to the current distribution $\mathcal{P}_t^{\text{test}}$. We call this problem *online distribution shift adaptation* and illustrate it in Figure **??**. Given a total T steps in the online test stage, we define the average loss for any online algorithm $\mathcal{A}$ through the loss of the sequence of models f_t, $t \in [T]$ that are produced from $\mathcal{A}$, i.e.,

$$L(\mathcal{A}; \mathcal{P}_1^{\text{test}}, \cdots, \mathcal{P}_T^{\text{test}}) = \frac{1}{T} \sum_{t=1}^{T} \ell(f_t; \mathcal{P}_t^{\text{test}}), \tag{1}$$

where $\ell(f; \mathcal{P}) = \mathbb{E}_{(x,y)\sim\mathcal{P}} \ell_{\text{sup}}(f(x), y)$ and ℓ_{sup} is the loss function, for example, 0-1 loss or cross-entropy loss for classification tasks.

In this paper, we consider a realistic but highly non-trivial case where at each time step t, only *a small batch of unlabeled samples* $S_t = \{x_t^1, \cdots, x_t^B\}$ is received. An effective algorithm under this scenario has to utilize information from all historical data (both train data D_0, validation data D_0' and test data up to time t), as well as previously deployed models $f_{0,1,\cdots,t-1}$. We formalize the algorithm $\mathcal{A}$ as:

$$f_t := \mathcal{A}\left(\{S_1, \cdots, S_{t-1}\}, \{f_0, f_1, \cdots, f_{t-1}\}, D_0, D_0'\right) \quad \forall t \in [T]. \tag{2}$$

We particularly focus on the online (generalized) label shift adaptation.

- In online label shift, the conditional distribution $\mathcal{P}_t^{\text{test}}(x|y)$ is invariant and equivalent to $\mathcal{P}^{\text{train}}(x|y)$ for all $t \in [T]$, while the marginal distribution of the label $\mathcal{P}_t^{\text{test}}(y)$ changes over time. This assumption is most typical when the label y is the causal variable and and the feature x is the observation Schölkopf et al. [2012]. Most previous methods tackle this problem by a non-trivial reduction to the classical online learning problem. Given this, most of the online label shift algorithms Wu et al. [2021], Bai et al. [2022], Baby et al. [2023] study the theoretical guarantee of the algorithm via the convergence of the regret function, either static regret or dynamic regret.

- The concept of generalized label shift, as introduced in Tachet des Combes et al. [2020], formalizes this situation by postulating the existence of an unknown function h, such that the conditional probability distribution $\mathcal{P}(h(x)|y)$ remains invariant. The primary challenge is to find this underlying transformation h. Building upon this, online generalized label shift assumes that, for every time step t, the test distribution $\mathcal{P}_t^{\text{test}}$ exhibits a form of generalized label shift from the training distribution $\mathcal{P}^{\text{train}}$, and this shift is governed by the same underlying transformation h.

3 Method

Inspired by the body of work in semi-supervised learning Grandvalet and Bengio [2004b], Lee et al. [2013], Laine and Aila [2016], Gidaris et al. [2018], Miyato et al. [2018] and unsupervised representation learning Chen et al. [2020a], He et al. [2019], Grill et al. [2020], He et al. [2022], self-supervised learning (SSL) techniques emerge as promising tools for enhancing feature extraction from unlabeled data, e.g. for image classification. In this section, we discuss how to utilize various SSL techniques to improve the feature representation learning of existing online label shift algorithms. To illustrate the concept, we narrow our focus to three particular SSL techniques for classification tasks: rotation degree prediction Gidaris et al. [2018], Sun et al. [2020], entropy minimization Grandvalet and Bengio [2004b], Wang et al. [2020] and MoCo He et al. [2019], Chen et al. [2020b, 2021]. It is important to note that this concept extends beyond these three SSL techniques, and the incorporation of more advanced SSL techniques has the potential to further elevate the performance. The details of these SSL are introduced in Appendix D.10.

Method. We formally introduce *Online Label Shift adaptation with Online Feature Updates* (OLS-OFU; Algorithm 1), which requires a self-supervised learning loss ℓ_{ssl} and an online label shift adaptation algorithm (OLS) ($\in${ROGD Wu et al. [2021], FTH Wu et al. [2021], UOGD Bai et al. [2022], ATLAS Bai et al. [2022], FLHFTL Baby and Wang [2022]}) that either reweights the offline pretrained model f_0 or updates the last linear layer. In the train stage, we train f_0 by minimizing the supervised and self-supervised loss together. In the test stage, OLS-OFU comprises three steps at each time step t: (1) running the refined version of OLS, which we refer to as OLS-R, (2) updating the feature extractor by ℓ_{ssl}, (3) re-training the last linear layer. Algorithm 1 formally described the procedure; check the details of these three steps in Appendix B.1.

Analysis for online label shift. The original OLS methods in the literature exhibit theoretical guarantees in terms of regret convergence for online label shift setting. With the incorporation of the additional online feature update step, OLS-OFU demonstrates analogous theoretical results; check the details of theorems in Appendix C. By comparing the theoretical results between OLS and OLS-OFU, we can gain insights into potential enhancements from OLS to OLS-OFU.

Analysis for online generalized label shift. Existing research in test-time training (TTT) Sun et al. [2020], Wang et al. [2020], Liu et al. [2021], Niu et al. [2022] demonstrates that feature updates driven by SSL align the source and target domains in feature space. When the source and target domains achieve perfect alignment, such feature extractor effectively serves as the feature map h as assumed in generalized label shift. Therefore, the sequence of feature extractors in $f_1, \cdots, f_T$ generated by Algorithm 1 progressively approximates the underlying h. This suggests that, compared to the original OLS, OLS with online feature updates (Algorithm 1) experiences a milder violation of the label shift assumption within the feature space and is actually expected to have better performance in the setting of online generalized label shift.

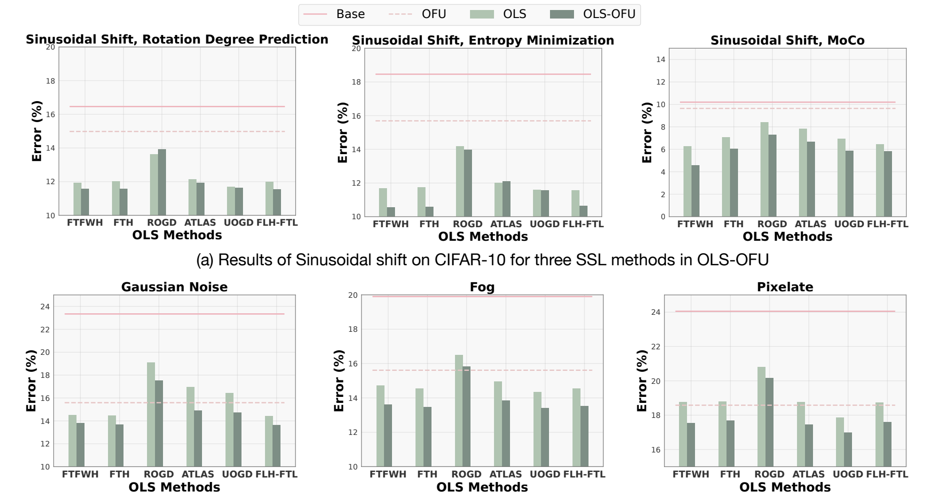
4 Experiment

In this section, we empirically evaluate how OLS-OFU improves the original OLS methods on both online label shift and online generalized label shift[3]. The train data in the experiment is CIFAR-10 train set and the online test data is sampled from CIFAR-10 Krizhevsky et al. [2009] and CIFAR-10C Hendrycks and Dietterich [2019] to evaluate online label shift or online generalized label shift. We vary the shift patterns (Sinusoidal and Bernoulli shifts) and SSL techniques (rotation degree

[3] Code released at `https://anonymous.4open.science/r/online_label_shift_with_online_feature_updates-3A1F/`.

Algorithm 1 Online label shift adaptation with online feature updates (OLS-OFU).

```
Require: An online label shift adaptation algorithm OLS ($\in$\{ROGD, FTH, UOGD, ATLAS, FLHFTL\}), a self-supervised learning loss $\ell_{\text{ssl}}$. A pretrained model $f_0$ and initialize $f_1 = f_0$.
for $t = 1, \cdots, T$ do
    Input at time $t$: Samples $S_1 \cup \cdots \cup S_t$, models $\{f_1, \cdots, f_t\}$, train set $D_0$, validation set $D'_0$.
    1. Run the revised version of OLS, that is, OLS-R, and get $f'_{t+1}$. (See Appendix B.2 for revisions of ROGD, FTH, UOGD, ATLAS, FLHFTL)
    2. Update the feature extractor $\theta_t^{\text{feat}}$ in $f'_{t+1}$ by
           $$\theta_{t+1}^{\text{feat}} := \theta_t^{\text{feat}} - \eta \cdot \nabla_{\theta^{\text{feat}}} \ell_{\text{ssl}}(S_t; f'_{t+1}).$$
       Replace the feature extractor $\theta_t^{\text{feat}}$ in $f'_{t+1}$ by $\theta_{t+1}^{\text{feat}}$.
    3. Within the feature extractor $\theta_{t+1}^{\text{feat}}$, re-train the last linear layer from random initialization by minimizing the average loss among train data $D_0$:
           $$\theta_{t+1}^{\text{linear}} := \arg\min_{\theta_{\text{linear}}} \sum_{(x,y)\in D_0} \ell_{\text{ce}}\bigg(f(x|\theta_{t+1}^{\text{feat}}, \theta^{\text{linear}}), y\bigg).$$
       Calibrate the model $f(\cdot|\theta_{t+1}^{\text{feat}}, \theta_{t+1}^{\text{linear}})$ by temperature calibration using the validation set $D'_0$ and denote the model after calibration as $f''_{t+1}$.
    4. If the parameter space of OLS is a reweighting version of the prediction model (ROGD, FTH, FLHFTL), suppose the reweighting vector in $f'_{t+1}$ is $p_{t+1}$ and we define $f_{t+1} := g(\cdot; f''_{t+1}, p_{t+1})$; else (UOGD, ATLAS), we define $f_{t+1} := f'_{t+1}$.
end for
```



(a) Results of Sinusoidal shift on CIFAR-10 for three SSL methods in OLS-OFU

(b) Results of Sinunoidal shift on three types of corruptions in CIFAR-10C. SSL method in OLS-OFU is rotation degree prediction.

Figure 1: Comparison of OLS-OFU and OLS in CIFAR-10 and CIFAR-10C.

prediction, entropy minimization, and MoCo) to evaluate the efficacy of the method. More details and additional results are presented in Appendix D.

Main results. Figure 1(a) compares the performance of various OLS-OFU algorithms with their respective OLS counterparts, under classical online label shift. Figure 1(b) presents the performance of OLS-OFU in the context of online generalized label shift, where the test images exhibit three types of domain shifts in CIFAR-10C — Gaussian noise, Fog, and Pixelation—with mild severity. Firstly, the advantage of OLS over BASE and OFU in Figure 1(a) demonstrates the inherent advantages of OLS methods in effectively addressing the online label shift problem. Some OLS methods perform worse than OFU on CIFAR-10C as expected because the assumption of label shift no longer holds in this generalized label shift setting. It's worth highlighting that OLS-OFU, when integrated with three distinct SSL methods and tested with three types of domain shifts, consistently outperforms OLS across all six OLS methods. This underscores the importance of improving feature representation learning within (generalized) label shift adaptation in OLS-OFU.

Acknowledgments and Disclosure of Funding

RW and KQW are supported by grants from LinkedIn and DARPA Geometries of Learning. RW is also supported by grants from from the National Science Foundation NSF (2241100), Army Research Office ARO MURI (W911NF2110317), and Office of Naval Research ONR (N00014-20-1-2334).

References

A. Alexandari, A. Kundaje, and A. Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International Conference on Machine Learning*, pages 222–232. PMLR, 2020.

E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020.

K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations*, 2019.

D. Baby and Y.-X. Wang. Optimal dynamic regret in proper online learning with strongly convex losses and beyond. In *International Conference on Artificial Intelligence and Statistics*, pages 1805–1845. PMLR, 2022.

D. Baby, S. Garg, T.-C. Yen, S. Balakrishnan, Z. C. Lipton, and Y.-X. Wang. Online label shift: Optimal dynamic regret meets practical algorithms. *To appear at Advances in Neural Information Processing Systems*, 2023.

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

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

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020a.

X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

X. Chen, S. Xie, and K. He. An empirical study of training self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9620–9629, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00950. URL `https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00950`.

A. Coates, A. Ng, and H. Lee. An Analysis of Single Layer Networks in Unsupervised Feature Learning. In *AISTATS*, 2011. `https://cs.stanford.edu/~acoates/papers/coatesleeng_aistats_2011.pdf`.

L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey. Cinic-10 is not imagenet or cifar-10, 2018.

S. Garg, Y. Wu, S. Balakrishnan, and Z. C. Lipton. A unified view of label shift estimation. *arXiv preprint arXiv:2003.07554*, 2020.

S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=S1v4N2l0-`.

Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004a. URL `https://proceedings.neurips.cc/paper_files/paper/2004/file/96f2b50b5d3613adf9c27049b2a888c7-Paper.pdf`.

Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004b.

A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

P. Helber, B. Bischke, A. Dengel, and D. Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. doi: 10.1109/JSTARS.2019.2918242.

D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

J. Hoffman, T. Darrell, and K. Saenko. Continuous manifold based adaptation for evolving visual domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 867–874, 2014.

J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, volume 19, pages 601–608. Citeseer, 2006.

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop: Challenges in Representation Learning*, 2013.

D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013.

Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in nonstandard situations. *Machine learning*, 46(1):191–202, 2002.

Z. Lipton, Y.-X. Wang, and A. Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pages 3122–3130. PMLR, 2018.

Y. Liu, P. Kothari, B. Van Delft, B. Bellot-Gurlet, T. Mordan, and A. Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820, 2021.

T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

R. T. Mullapudi, S. Chen, K. Zhang, D. Ramanan, and K. Fatahalian. Online model distillation for efficient video inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3573–3582, 2019.

S. Niu, J. Wu, Y. Zhang, Y. Chen, S. Zheng, P. Zhao, and M. Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022.

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

B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

S. Shalev-Shwartz. 2012.

H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020.

R. Tachet des Combes, H. Zhao, Y.-X. Wang, and G. J. Gordon. Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, 33:19276–19289, 2020.

D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

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

Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2020. doi: 10.1109/CVPR42600.2020.01070.

B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 114, 2004.

Y.-J. Zhang, Z.-Y. Zhang, P. Zhao, and M. Sugiyama. Adapting to continuous covariate shift via online density ratio estimation. *arXiv preprint arXiv:2302.02552*, 2023.

A Further Related Work

Offline distribution shift and domain shift. Offline label shift and covariate shift have been studied for many years. Some early work Saerens et al. [2002], Lin et al. [2002] assumes the knowledge of how the distribution is shifted. Later work Shimodaira [2000], Zadrozny [2004], Huang et al. [2006], Gretton et al. [2009], Lipton et al. [2018], Alexandari et al. [2020], Azizzadenesheli et al. [2019], Garg et al. [2020] relaxes this assumption and estimates this knowledge from unlabeled test data.

Online distribution shift with provable guarantees. There has been several work modeling online distribution shift as the classic online learning problem Wu et al. [2021], Bai et al. [2022], Baby et al. [2023], Zhang et al. [2023], which leverage the classical online learning algorithms Shalev-Shwartz [2012], Besbes et al. [2015], Baby and Wang [2022] to bound the static or dynamic regret. However, none of them updates the feature extractor in a deep learning model but only the last linear layer or the post-hoc linear reweighting vectors. Our proposed method OLS-OFU utilizes the deep learning SSL to improve the feature extractor, which brings better performance.

Domain shift adaptation within online streaming data. When we consider the most authentic online learning setup where the learner only receives the unlabeled samples, the most representative idea is test-time training Sun et al. [2020], Wang et al. [2020], Liu et al. [2021], Niu et al. [2022], which utilizes a (deep learning) self-supervised loss to online update the model. However, it focuses on how to adapt to a *fixed* domain shifted distribution from online streaming data and is not designed for how to adapt to continuous distribution changes during the test stage, while our algorithm concentrates the later problem. Besides test-time training, Hoffman et al. [2014] and Mullapudi et al. [2019] study the online domain shift for specific visual applications.

Algorithm 2 Revised ROGD for online feature updates ROGD-R. See the original version in Equation 7 and Equation 8 in Wu et al. [2021].

```
Require: Learning rate $\eta$.
for $t = 1, \cdots, T$ do
    Input at time $t$: Samples $S_1 \cup \cdots \cup S_t$, models $\{f_1, \cdots, f_t\}$, and intermediate model $\{f_1'', \cdots, f_t''\}$ from step 3 in Algorithm 1, the validation set $D_0'$, the training label marginal $q_0 := \mathcal{P}^{\text{train}}(y)$.
    1. Compute the unbiased estimator for label marginal distribution:
        $s_t = \frac{1}{|S_t|} \sum_{x_t \in S_t} C^{-1}_{f_t'', D_0'} f_t''(x_t)$.        ▷ In the original ROGD, it is $f_0$ rather than $f_t''$.
    2. Grab the weight $p_t$ from $f_t$.
    3. Update $p_{t+1} := \text{Proj}_{\Delta^{K-1}} \left[ p_t - \eta \cdot J_p(p_t)^\top s_t \right]$,
        where $J_{p, f_t''}(p_t) = \frac{\partial}{\partial p}(1 - \text{diag}(C_{f_t'', D_0, p}))|_{p=p_t}$, and let $f_{t+1}$ be a reweighting version of
        $f_t''$ by the weight $\left( \frac{p_{t+1}[k]}{q_0[k]} : k = 1, \cdots K \right)$        ▷ In the original ROGD, it is $f_0$ rather than $f_t''$.
    Output at time $t$: $f_{t+1}'$.
end for
```

Algorithm 3 Revised FTH for online feature updates (FTH-R). See the original version in Equation 9 in Wu et al. [2021].

```
for $t = 1, \cdots, T$ do
    Input at time $t$: Samples $S_1 \cup \cdots \cup S_t$, models $\{f_1, \cdots, f_t\}$, and intermediate model $\{f_1'', \cdots, f_t''\}$ from step 3 in Algorithm 1, the validation set $D_0'$, the train label marginal $q_0 := \mathcal{P}^{\text{train}}(y)$.
    1. Compute the unbiased estimator for label marginal distribution:
        $s_t = \frac{1}{|S_t|} \sum_{x_t \in S_t} C^{-1}_{f_t, D_0'} f_t''(x_t)$.        ▷ In the original FTL, it is $f_0$ rather than $f_t''$.
    2. Compute $p_{t+1} = \frac{1}{t} \sum_{\tau=1}^{t} s_\tau$
    3. Let $f_{t+1}$ be a reweighting version of $f_t''$ by
        the weight $\left( \frac{p_{t+1}[k]}{q_0[k]} : k = 1, \cdots K \right)$        ▷ In the original FTL, it is $f_0$ rather than $f_t''$.
    Output at time $t$: $f_{t+1}'$.
end for
```

Algorithm 4 Revised UOGD for online feature updates (UOGD-R). See the original version in Equation 9 in Bai et al. [2022].

```
Require: The learning rate $\eta$.
for $t = 1, \cdots, T$ do
  Input at time $t$: Samples $S_1 \cup \cdots \cup S_t$, models $\{f_1, \cdots, f_t\}$, and intermediate model $\{f''_1, \cdots, f''_t\}$ from step 3 in Algorithm 1, the validation set $D'_0$, the train label marginal $q_0 := \mathcal{P}^{\text{train}}(y)$.
  1. Compute the unbiased estimator for label marginal distribution:
     $s_t = \frac{1}{|S_t|} \sum_{x_t \in S_t} C^{-1}_{f''_t, D'_0} f''_t(x_t)$.    ▷ In the original UOGD, it is $f_0$ rather than $f''_t$.
  2. Grab the weight $w_t$ from the last linear layer of $f_t$.
  3. Update $w_{t+1} := w_t - \eta \cdot \frac{\partial}{\partial w} J_w(w_t)^\top s_t$, where $J_w(w_t) = \frac{\partial}{\partial w}(\hat{R}^1_t(w), \cdots, \hat{R}^K_t(w))|_{w=w_t}$,
     $\hat{R}^k_t(w) = \frac{1}{|D^k_0|} \sum_{(x,y) \in D^k_0} \ell_{\text{ce}}(f(x|\theta^{\text{feat}}_t, \theta^{\text{linear}} = w), y)$, $D^k_0$ denotes the set of data with label $k$ in $D_0$.    ▷ In the original UOGD, it is $\theta^{\text{feat}}_0$ rather than $\theta^{\text{feat}}_t$.
  4. Let $f_{t+1}$ be $f(\cdot|\theta^{\text{feat}}_t, w_{t+1})$.
  Output at time $t$: $f'_{t+1}$.
end for
```

Algorithm 5 Revised ATLAS for online feature updates (ATLAS-R). See the original version in Equation 9 in Bai et al. [2022].

```
Require: The learning rate pool $\mathcal{H}$ with size N; Meta learning rate $\varepsilon$; $\forall i \in [N]$, $p_{1,i} = 1/N$ and $w_{1,i} = \theta^{\text{linear}}_{f_0}$.
for $t = 1, \cdots, T$ do
  Input at time $t$: Samples $S_1 \cup \cdots \cup S_t$, models $\{f_1, \cdots, f_t\}$, and intermediate model $\{f''_1, \cdots, f''_t\}$ from step 3 in Algorithm 1, the validation set $D'_0$, the train label marginal $q_0 := \mathcal{P}^{\text{train}}(y)$.
  1. Compute the unbiased estimator for label marginal distribution:
     $s_t = \frac{1}{|S_t|} \sum_{x_t \in S_t} C^{-1}_{f_t, D'_0} f''_t(x_t)$.    ▷ In the original ATLAS, it is $f_0$ rather than $f''_t$.
  for $i \in [N]$ do
    2. Update $w_{t+1,i} := w_{t,i} - \eta_i \cdot \frac{\partial}{\partial w} J_w(w_{t,i})^\top s_t$, where
       $J_w(w_{t,i}) = \frac{\partial}{\partial w}(\hat{R}^1_t(w), \cdots, \hat{R}^K_t(w))|_{w=w_{t,i}}$,
       $\hat{R}^k_t(w) = \frac{1}{|D^k_0|} \sum_{(x,y) \in D^k_0} \ell_{\text{ce}}(f(x|\theta^{\text{feat}}_t, w), y)$, $D^k_0$ denotes the set of data
       with label $k$ in $D_0$.    ▷ In the original ATLAS, it is $\theta^{\text{feat}}_0$ rather than $\theta^{\text{feat}}_t$.
  end for
  3. Update weight $p_{t+1}$ according to $p_{p_{t,i}} \propto \exp(-\varepsilon \sum_{\tau=1}^{t-1} \hat{R}_\tau(\mathbf{w}_{\tau,i}))$
  3. Compute $w_{t+1} = \sum_{i=1}^{N} p_{t+1,i} w_{t+1,i}$. Let $f_{t+1}$ be $f(\cdot|\theta^{\text{feat}}_t, w_{t+1})$.
  Output at time $t$: $f'_{t+1}$.
end for
```

B Additional Details of OLS-OFU

B.1 OLS-OFU Details

(1) Running the Revised OLS. First, we review common OLS methods (FLHFTL, FTH, ROGD, UOGD, and ATLAS) and identify two specific points in the algorithm where we can employ the updated prediction model f''_t (with the improved feature extractor[4]) to supplant the pretrained model f_0, hence enhancing the original OLS algorithm. Denote $C_{f,D} \in [0,1]^{K \times K}$ the confusion matrix evaluated on dataset D for the model f with $C_{f,D}[i,j] = \mathbb{P}_{(x,y) \sim D}\big(\arg\max(f(x)) = j | y = i\big)$. At the outset, all OLS methods rely on an unbiased estimator s_t of the label distribution q_t with $q_t[y] = \mathcal{P}^{\text{test}}_t(y)$, where $s_t := \frac{1}{|S_t|} \sum_{x_t \in S_t} C^{-1}_{f_0, D_0} f_0(x_t)$. This is the first point that we can replace f_0 with the improved prediction model f''_t to enhance the estimation of label marginal distribution. For the second point,

[4] We detail how to obtain this enhanced model f''_t using SSL techniques in the following paragraphs.

Algorithm 6 Revised FLHFTL for online feature updates (FLHFTL-R); See the original version in Algorithm 2 in Baby et al. [2023].

```
Require: Online regression oracle ALG.
for $t = 1, \cdots, T$ do
    Input at time $t$: Samples $S_1 \cup \cdots \cup S_t$, models $\{f_1, \cdots, f_t\}$, and intermediate model $\{f''_1, \cdots, f''_t\}$ from step 3 in Algorithm 1, the validation set $D'_0$, the train label marginal $q_0 := \mathcal{P}^{\text{train}}(y)$.
    1. Compute the unbiased estimator for label marginal distribution:
        $s_t = \frac{1}{|S_t|} \sum_{x_t \in S_t} C^{-1}_{f''_t, D'_0} f''_t(x_t)$        ▷ In the original FLHFTL, it is $f_0$ rather than $f''_t$.
    2. Compute the online estimator $\tilde{q}_{t+1} := \text{ALG}(s_1, \cdots, s_t)$
    3. Let $f'_{t+1}$ be a reweighting version of $f''_t$ by the weight
        $\left(\frac{\tilde{q}_{t+1}[k]}{q_0[k]} : k = 1, \cdots K\right)$        ▷ In the original FLHFTL, it is $f_0$ rather than $f''_t$.
    Output at time $t$: $f'_{t+1}$.
end for
```

- FLHFTL and FTH subsequently employ a vector $\tilde{q}_t$[5] to reweight the initial pretrained model f_0. Now, instead of reweighting the original pretrained model f_0, the algorithm reweights the improved model f''_t, utilizing its enhanced predictive performance.
- ROGD, UOGD, and ATLAS initially update the model through an unbiased gradient estimator within a hypothesis space that either includes a linear model after a fixed feature extractor in f_0 or a reweighted version of f_0. Now, we have the flexibility to shift to a comparable hypothesis space, replacing f_0 with the improved model f''_t, and continue applying updates using the same unbiased gradient estimator.

The revisions for ROGD, FTH, UOGD, and ATLAS can be found in Section B.2. We use f'_{t+1} to denote the model after running the revised OLS.

(2) Updating the Feature Extractor. We now introduce how to utilize an SSL loss ℓ_{ssl} to update the feature extractor for any incoming unlabeled test data batch S_t at timestep t. Specifically, let θ^{feat}_t denote the parameters of the feature extractor in f'_{t+1}. The update of $\theta^{\text{feat}}_{t+1}$ at time t is given by:

$$\theta^{\text{feat}}_{t+1} := \theta^{\text{feat}}_t - \eta \cdot \nabla_{\theta^{\text{feat}}} \ell_{\text{ssl}}(S_t; f'_{t+1}).$$

We replace the feature extractor in f'_{t+1} by $\theta^{\text{feat}}_{t+1}$.

(3) Re-training Last Linear Layer. Given the updated feature extractor $\theta^{\text{feat}}_{t+1}$, it is necessary to re-train the last linear layer $\theta^{\text{linear}}_{t+1}$ to adapt to the new feature extractor. We start the re-raining from random initialization, while keeping the feature extractor frozen. The train objective of $\theta^{\text{linear}}_{t+1}$ is to minimize the average cross-entropy loss under train data D_0. We denote the model with the frozen feature extractor $\theta^{\text{feat}}_{t+1}$ as $f(\cdot|\theta^{\text{feat}}_{t+1}, \theta^{\text{linear}})$. The objective for re-training last linear layer can be written as follows:

$$\theta^{\text{linear}}_{t+1} := \arg\min_{\theta_{\text{linear}}} \sum_{(x,y) \in D_0} \ell_{\text{ce}}\Big(f(x|\theta^{\text{feat}}_{t+1}, \theta^{\text{linear}}), y\Big).$$

We calibrate the model $f(\cdot|\theta^{\text{feat}}_{t+1}, \theta^{\text{linear}}_{t+1})$ by temperature calibration Guo et al. [2017] using the validation set D'_0 and denote the model after calibration as f''_{t+1}. This re-training step is needed to ensure the model is a calibrated model of estimating $\mathcal{P}^{\text{train}}(y|x)$ for any given input x.

In the end, we are going to define f_{t+1} for the next time step. If the parameter space of the original OLS is a reweighting version of the prediction model (ROGD, FTH, FLHFTL), suppose the reweighting vector in f'_{t+1} is p_{t+1} and we define $f_{t+1} := g(\cdot; f''_{t+1}, p_{t+1})$; else (UOGD, ATLAS), we define $f_{t+1} := f'_{t+1}$.

[5] The reweighting factor $\tilde{q}_t$ is a function of unbiased estimators $s_1, \cdots, s_t$ in FLHFTL and FTH.

B.2 The Revision for Previous Online Label Shift Adaptation Algorithms

The revised algorithms to be used in the main algorithm OLS-OFU (Algorithm 1) are FTH-R (Algorithm 3), UOGD-R (Algorithm 4), ROGD-R (Algorithm 2), ATLAS-R (Algorithm 5), FLHFTL-R (Algorithm 6).

C Theorems for OLS and Proofs

C.1 Performance Guarantee for FLHFTL-OFU

The original Online Label Shift (OLS) methods exhibit theoretical guarantees in terms of regret convergence for online label shift setting, where $\mathcal{P}_t^{\text{test}}(x|y) = \mathcal{P}^{\text{train}}(x|y)$ is invariant. With the incorporation of the additional online feature update step, OLS-OFU demonstrates analogous theoretical results. By comparing the theoretical results between OLS and OLS-OFU, we can gain insights into potential enhancements from OLS to OLS-OFU.

Theorem 1. ***[Regret convergence for FLHFTL-OFU]*** *Suppose we choose the OLS subroutine in Algorithm 6 to be FLH-FTL from Baby et al. [2023]. Let $f_t^{\text{flhftl}-\text{ofu}}$ be the output at time step $t-1$ from Algorithm 1, that is $g(\cdot; f_t'', \tilde{q}_t/q_0)$. Let σ be the smallest among the the minimum singular values of invertible confusion matrices $\{C_{f_1'',D_0'}, \cdots C_{f_T'',D_0'}\}$. Then under Assumptions 1 and 2 in Baby et al. [2023], FLHFTL-OFU has the guarantee for online label shift below:*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\ell(f_t^{\text{flhftl}-\text{ofu}}; \mathcal{P}_t^{\text{test}}) - \frac{1}{T}\sum_{t=1}^{T}\ell(g(\cdot; f_t'', q_t/q_0); \mathcal{P}_t^{\text{test}})\right] \leq O\left(\frac{K^{1/6}V_T^{1/3}}{\sigma^{2/3}T^{1/3}} + \frac{K}{\sigma\sqrt{T}}\right), \tag{3}$$

where $V_T := \sum_{t=1}^{T}\|q_t - q_{t-1}\|_1$, K is the number of classes, and the expectation is taken w.r.t. randomness in the revealed co-variates. This result is attained without prior knowledge of V_T.

To ease comparison, we state the theorem for the original OLS algorithm FLHFTL.

Theorem 2. ***[Regret convergence for FLHFTL Baby et al. [2023]]*** *Under Assumptions 1 and 2 in Baby et al. [2023], FLHFTL has the guarantee for online label shift below:*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\ell(f_t^{\text{flhftl}}; \mathcal{P}_t^{\text{test}})\right] - \frac{1}{T}\sum_{t=1}^{T}\ell(g(\cdot; f_0, q_t/q_0); \mathcal{P}_t^{\text{test}}) \leq O\left(\frac{K^{1/6}V_T^{1/3}}{\sigma^{2/3}T^{1/3}} + \frac{K}{\sigma\sqrt{T}}\right), \tag{4}$$

where $V_T := \sum_{t=1}^{T}\|q_t - q_{t-1}\|_1$, σ denotes the minimum singular value of invertible confusion matrices $C_{f_0,D_0'}$, K is the number of classes, and the expectation is taken with respect to randomness in the revealed co-variates. Further, this result is attained without prior knowledge of V_T.

Recall that the objective function for the online label shift problem is defined as the average loss in Equation 1. Both theorems establish the convergence of this average loss. In the event that f_t'' $(t \in [T])$ from the online feature updates yield improvements:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\ell(g(\cdot; f_t'', q_t/q_0); \mathcal{P}_t^{\text{test}})\right] < \frac{1}{T}\sum_{t=1}^{T}\ell(g(\cdot; f_0, q_t/q_0); \mathcal{P}_t^{\text{test}}), \tag{5}$$

then it guarantees that the loss of FLHFTL-OFU will converge to a smaller value, resulting in enhanced performance compared to FLHFTL. We substantiate this improvement through empirical evaluation in Section 4. For other OLS algorithms such as ROGD-OFU, FTH-OFU, UOGD-OFU, and ATLAS-OFU, a similar analysis can be derived and we present them in next subsections. The proofs are mostly the same as the proofs for the original algorithms with small adjustments. As our results are not straight corollaries for the original theorems, we write the full proofs here for the completeness.

C.2 Theorem for FLHFTL-OFU

Before proving Theorem 1 (in Section C.1) we recall the assumption from Baby et al. [2023] for convenience. We refer the reader to Baby et al. [2023] for justifications and further details of the assumptions.

Assumption 1. *Assume access to the true label marginals $q_0 \in \Delta_K$ of the offline train data and the true confusion matrix $C \in \mathbb{R}^{K \times K}$. Further the minimum singular value $\sigma_{min}(C) = \Omega(1)$ is bounded away from zero.*

Assumption 2 (Lipschitzness of loss functions)**.** *Let $\mathcal{D}$ be a compact and convex domain. Let r_t be any probabilistic classifier. Assume that $L_t(p) := E\left[\ell(g(\cdot; r_t, p/q_0)|x_t\right]$ is G Lipschitz with $p \in \mathcal{D} \subseteq \Delta_K$, i.e, $L_t(p_1) - L_t(p_2) \leq G\|p_1 - p_2\|_2$ for any $p_1, p_2 \in \mathcal{D}$. The constant G need not be known ahead of time.*

Theorem 1. ***[Regret convergence for FLHFTL-OFU]*** *Suppose we choose the OLS subroutine in Algorithm 6 to be FLH-FTL from Baby et al. [2023]. Let $f_t^{\mathrm{flhftl-ofu}}$ be the output at time step $t-1$ from Algorithm 1, that is $g(\cdot; f_t'', \tilde{q}_t/q_0)$. Let σ be the smallest among the the minimum singular values of invertible confusion matrices $\{C_{f_1'',D_0'}, \cdots C_{f_T'',D_0'}\}$. Then under Assumptions 1 and 2 in Baby et al. [2023], FLHFTL-OFU has the guarantee for online label shift below:*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\ell(f_t^{\mathrm{flhftl-ofu}}; \mathcal{P}_t^{\mathrm{test}}) - \frac{1}{T}\sum_{t=1}^{T}\ell(g(\cdot; f_t'', q_t/q_0); \mathcal{P}_t^{\mathrm{test}})\right] \leq O\left(\frac{K^{1/6}V_T^{1/3}}{\sigma^{2/3}T^{1/3}} + \frac{K}{\sigma\sqrt{T}}\right), \tag{3}$$

where $V_T := \sum_{t=1}^{T}\|q_t - q_{t-1}\|_1$, K is the number of classes, and the expectation is taken w.r.t. randomness in the revealed co-variates. This result is attained without prior knowledge of V_T.

Proof:

The algorithm in Baby et al. [2023] requires that the estimate s_t in Line 1 of Algorithm 6 is unbiased estimate of the label marginal q_t. Since f_t'' in Algorithm 6 is independent of the sample S_t, and since we are working under the standard label shift assumption, due to Lipton et al. [2018] we have that $C_{f_t'',D_0'}^{-1} \cdot \frac{1}{|S_t|}\sum_{x_t \in S_t} f_t''(x_t)$ forms an unbiased estimate of $E_{x \sim \mathcal{P}_t^{\mathrm{test}}}[f_t''(x)]$. Further, from Lipton et al. [2018], the reciprocal of standard deviation of this estimate is bounded below by minimum of the singular values of confusion matrices $\{C_{f_1'',D_0'}, \cdots C_{f_T'',D_0'}\}$.

Let $\tilde{q}_t$ be the estimate of the label marginal maintained by FLHFTL. By Lipschitzness, we have that

$$\begin{aligned}
E[\ell(f_t^{\mathrm{flhftl-ofu}}; \mathcal{P}_t^{\mathrm{test}}) - \ell(g(\cdot; f_t'', p/q_0)] &= E[L_t(\tilde{q}_t)] - E[L_t(q_t)] && (6)\\
&\leq G \cdot E[\|\tilde{q}_t - q_t\|_2], && (7)
\end{aligned}$$

where the last line is via Assumption 2. Rest of the proof is identical to that of Baby et al. [2023]. We reproduce it below for completeness.

$$\begin{aligned}
\sum_{t=1}^{T} E[\ell(f_t^{\mathrm{flhftl-ofu}}; \mathcal{P}_t^{\mathrm{test}}) - \ell(g(\cdot; f_t'', p/q_0)] &\leq \sum_{t=1}^{T} G \cdot E[\|\tilde{q}_t - q_t\|_2] && (8)\\
&\leq \sum_{t=1}^{T} G\sqrt{E\|\tilde{q}_t - q_t\|_2^2} && (9)\\
&\leq G\sqrt{T\sum_{t=1}^{T} E[\|\tilde{q}_t - q_t\|_2^2]} && (10)\\
&= \tilde{O}\left(K^{1/6}T^{2/3}V_T^{1/3}(1/\sigma_{min}^{2/3}(C)) + \sqrt{KT}/\sigma_{min}(C)\right), && (11)
\end{aligned}$$

where the second line is due to Jensen's inequality, third line by Cauchy-Schwartz and last line by Proposition 16 in Baby et al. [2023]. This finishes the proof.

Proof of Theorem 2: The proof is similar to the arguments in the proof of Theorem 1. The only point of deviation is that we choose $r_t = f_0$ instead of f_t'' in Assumption 2. The rest of the arguments follow via Lipschitzness.

C.3 Theorem for ROGD-OFU

We state the assumptions first for the later theorems. These assumptions are similar to Assumption 1-3 in Wu et al. [2021].

Assumption 3. $\forall \mathcal{P} \in \{\mathcal{P}^{\text{train}}, \mathcal{P}_1^{\text{test}}, \cdots, \mathcal{P}_T^{\text{test}}\}$, $\operatorname{diag}(C_{f,\mathcal{P}})$ *is differentiable with respect to* f.

Assumption 4. $\forall t \in [T]$, $\ell(g(\cdot; f_t'', p/q_0); \mathcal{P}_t^{\text{test}})$ *is convex in* p*, where* f_t'' *is defined in Algorithm 1.*

Assumption 5. $\sup_{p\in\Delta^{K-1}, i\in[K], t\in[T]} \|\nabla_p \ell(g(\cdot; f_t'', p/q_0); \mathcal{P}_t^{\text{test}})\|_2$ *is finite and bounded by* L*.*

Theorem 3 (Regret convergence for ROGD-OFU)**.** *If we run Algorithm 1 with ROGD-R (Algorithm 2) and* $\eta = \sqrt{\frac{2}{T}}\frac{1}{L}$*, under Assumption 3, 4, 5, ROGD-OFU satisfies the guarantee*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \ell(f_t^{\text{ogd-ofu}}; \mathcal{P}_t^{\text{test}})\right] - \min_{p\in\Delta_K} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \ell(g(\cdot; f_t'', p/q_0); \mathcal{P}_t^{\text{test}})\right] \le \sqrt{\frac{2}{T}}L. \tag{12}$$

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \ell(f_t^{\text{ogd}}; \mathcal{Q}_t)\right] - \min_{p\in\Delta_K} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \ell(g(\cdot; p, f_0, q_0); \mathcal{Q}_t)\right] \le \sqrt{\frac{2}{T}}L. \tag{13}$$

Proof: For any fixed p,

$$\begin{aligned}
\ell(f_t^{\text{rogd}-\text{ofu}}; \mathcal{P}_t^{\text{test}}) - \ell(g(\cdot; f_t'', p/q_0); \mathcal{P}_t^{\text{test}}) &= \ell(g(\cdot; f_t'', p_t/q_0); \mathcal{P}_t^{\text{test}}) - \ell(g(\cdot; f_t'', p/q_0); \mathcal{P}_t^{\text{test}}) \\
&\le (p_t - p) \cdot \nabla_p \ell(g(\cdot; f_t'', p_t/q_0); \mathcal{P}_t^{\text{test}}) \\
&= (p_t - p) \cdot J_{p,f_t''}(p_t)^\top \mathbb{E}_{S_t}[s_t | S_1, \cdots, S_{t-1}] \\
&= \mathbb{E}_{S_t}[(p_t - p) \cdot J_{p,f_t''}(p_t)^\top s_t | S_1, \cdots, S_{t-1}],
\end{aligned}$$

where the last inequality holds by the fact that $(p_t - p) \cdot J_{p,f_t''}(p_t)^\top$ is independent of $\{S_1, \cdots, S_{t-1}\}$. To bound $(p_t - p) \cdot J_{p,f_t''}(p_t)^\top s_t$,

$$\begin{aligned}
\|p_{t+1} - p\|_2^2 &= \|\text{Prof}_{\Delta^{K-1}}(p_t - \eta \cdot J_{p,f_t''}(p_t)^\top s_t) - p\|_2^2 \\
&\le \|p_t - \eta \cdot J_{p,f_t''}(p_t)^\top s_t - p\|_2^2 \\
&= \|p_t - p\|_2^2 + \eta^2 \|J_{p,f_t''}(p_t)^\top s_t\|_2^2 - 2\eta (p_t - p) \cdot (J_{p,f_t''}(p_t)^\top s_t).
\end{aligned}$$

This implies

$$(p_t - p) \cdot (J_{p,f_t''}(p_t)^\top s_t) \le \frac{1}{2\eta}(\|p_t - p\|_2^2 - \|p_{t+1} - p\|_2^2) + \frac{\eta}{2}\|J_{p,f_t''}(p_t)^\top s_t\|_2^2$$

Thus

$$\begin{aligned}
&\mathbb{E}_{S_1,\cdots,S_T}\left[\frac{1}{T}\sum_{t=1}^{T} \ell(f_t^{\text{rogd}-\text{ofu}}; \mathcal{P}_t^{\text{test}}) - \frac{1}{T}\sum_{t=1}^{T} \ell(g(\cdot; f_t'', p/q_0); \mathcal{P}_t^{\text{test}})\right] \\
&\le \mathbb{E}_{S_1,\cdots,S_T}\left[\frac{1}{T}\sum_{t=1}^{T} \frac{1}{2\eta}(\|p_t - p\|_2^2 - \|p_{t+1} - p\|_2^2) + \frac{\eta}{2}\|J_{p,f_t''}(p_t)^\top s_t\|_2^2\right] \\
&\le \frac{1}{2\eta T}\|p_1 - p\|_2^2 + \frac{\eta}{2T}\sum_{t=1}^{T} \mathbb{E}_{S_1,\cdots,S_t}[\|J_{p,f_t''}(p_t)^\top s_t\|_2^2] \\
&\le \frac{1}{\eta T} + \frac{\eta L^2}{2} = \sqrt{\frac{2}{T}}L.
\end{aligned}$$

This bound holds for any p. Thus,

$$\mathbb{E}_{S_1,\cdots,S_T}\left[\frac{1}{T}\sum_{t=1}^{T} \ell(f_t^{\text{rogd}-\text{ofu}}; \mathcal{P}_t^{\text{test}})\right] - \min_{p\in\Delta^{K-1}} \mathbb{E}_{S_1,\cdots,S_T}\left[\frac{1}{T}\sum_{t=1}^{T} \ell(g(\cdot; f_t'', p/q_0); \mathcal{P}_t^{\text{test}})\right] \le \sqrt{\frac{2}{T}}L.$$

C.4 Theorem for FTH-OFU

We begin with two assumptions.

Assumption 6. *For any $\mathcal{P}^{\text{test}}$ s.t. $\mathcal{P}^{\text{test}}(x|y) = \mathcal{P}^{\text{train}}(x|y)$, denote $q_t := (\mathcal{P}_t^{\text{test}}(y=k) : k \in [K])$ and then*

$$\|q_t - \arg\min_{p \in \Delta^{K-1}} \ell(g(\cdot; f_t'', p/q_0); \mathcal{P}^{\text{test}})\| \le \delta.$$

Assumption 7. $\forall \mathcal{P}^{\text{test}}$ *s.t.* $\mathcal{P}^{\text{test}}(x|y) = \mathcal{P}^{\text{train}}(x|y)$, $\sup_p \|\nabla_p \ell(g(\cdot; f_t'', p/q_0); \mathcal{P}^{\text{test}})\| \le L$

Theorem 4 (Regret convergence for FTH-OFU)**.** *If we run Algorithm 1 with FTH-R (Algorithm 3) and assume σ is no larger than the minimum singular value of invertible confusion matrices $\{C_{f_1'', D_0'}, \cdots C_{f_T'', D_0'}\}$, under Assumption 6 and 7 with $\delta = 0$, FTH-OFU satisfies the guarantee that with probability at least $1 - 2KT^{-7}$ over samples $S_1 \cup \cdots \cup S_T$,*

$$\frac{1}{T}\sum_{t=1}^{T} \ell(f_t^{\text{fth-ofu}}; \mathcal{P}_t^{\text{test}}) - \min_{p \in \Delta_K} \frac{1}{T}\sum_{t=1}^{T} \ell(g(\cdot; f_t'', p/q_0); \mathcal{P}_t^{\text{test}}) \le O\left(\frac{\log T}{T} + \frac{1}{\sigma}\sqrt{\frac{K \log T}{T}}\right), \tag{14}$$

where K is the number of classes.

Proof: Denote $q_t := (\mathcal{P}_t^{\text{test}}(y=k) : k \in [K])$. By the Hoeffding and union bound, we have

$$\mathbb{P}\left(\forall t \le T, \|p_{t+1} - \frac{1}{t}\sum_{\tau=1}^{t} q_\tau\| \le \sqrt{K}\varepsilon_t\right) \ge 1 - \sum_{t=1}^{T} 2M \exp\left(-2\varepsilon_t^2 t/\sigma^2\right).$$

This implies that with probability at least $1 - \sum_{t=1}^{T} 2M \exp\left(-2\varepsilon_t^2 t/\sigma^2\right)$, $\forall p$,

$$\begin{aligned}
&\sum_{t=1}^{T} \ell(p_t; \mathcal{P}_t^{\text{test}}) - \sum_{t=1}^{T} \ell(g(\cdot; f_t'', p/q_0); \mathcal{P}_t^{\text{test}}) \\
&\le \sum_{t=1}^{T} \ell(g(\cdot; f_t'', \frac{1}{t}\sum_{\tau=1}^{t} q_\tau/q_0); \mathcal{P}_t^{\text{test}}) - \sum_{t=1}^{T} \ell(g(\cdot; f_t'', p/q_0); \mathcal{P}_t^{\text{test}}) + L\sqrt{M} \cdot \sum_{t=1}^{T} \varepsilon_t \\
&\le \sum_{t=1}^{T} \ell(g(\cdot; f_t'', \frac{1}{t-1}\sum_{\tau=1}^{t-1} q_\tau/q_0); \mathcal{P}_t^{\text{test}}) - \sum_{t=1}^{T} \ell(g(\cdot; f_t'', \frac{1}{t}\sum_{\tau=1}^{t} q_\tau/q_0); \mathcal{P}_t^{\text{test}}) + L\sqrt{M} \cdot \sum_{t=1}^{T} \varepsilon_t \\
&\le \sum_{t=1}^{T} L \left\| \frac{1}{t-1}\sum_{\tau=1}^{t-1} q_\tau - \frac{1}{t}\sum_{\tau=1}^{t} q_\tau \right\| + L\sqrt{M} \cdot \sum_{t=1}^{T} \varepsilon_t \\
&\le \sum_{t=1}^{T} \frac{L}{t} \left\| \frac{1}{t-1}\sum_{\tau=1}^{t-1} q_\tau - q_t \right\| + L\sqrt{M} \cdot \sum_{t=1}^{T} \varepsilon_t \\
&\le \sum_{t=1}^{T} \frac{2L}{t} + L\sqrt{M} \cdot \sum_{t=1}^{T} \varepsilon_t.
\end{aligned}$$

If we take $\varepsilon_t = 2\sigma\sqrt{\frac{\ln T}{T}}$, the above is equivalent to: with probability at least $1 - 2KT^{-7}$

$$\frac{1}{T}\sum_{t=1}^{T} \ell(p_t; \mathcal{P}_t^{\text{test}}) - \min_p \frac{1}{T}\sum_{t=1}^{T} \ell(g(\cdot; f_t'', p/q_0); \mathcal{P}_t^{\text{test}}) \le 2L\frac{\ln T}{T} + 4L\sigma\sqrt{\frac{K \ln T}{T}}$$

C.5 Theorems for UOGD-OFU and ATLAS-OFU

Theorem 5. *[Regret convergence for UOGD-OFU] Let $f(\cdot; \theta^{\text{feat}}_{f_t''}, w)$ denote a network with the same feature extractor as that of f_t'' and a last linear layer with weight w. Let $f^{\text{uogd-ofu}} = f(\cdot; \theta^{\text{feat}}_{f_t''}, w_t)$, where w_t is the weight maintained at round t by Algorithm 4. If we run Algorithm 1 with UOGD in*

Bai et al. [2022] and let step size be η, then under the same assumptions as Lemma 1 in Bai et al. [2022], UOGD-OFU satisfies that

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\ell(f^{\mathrm{uogd-ofu}};\mathcal{P}_t^{\mathrm{test}})-\frac{1}{T}\sum_{t=1}^{T}\min_{w\in\mathcal{W}}\ell(f(\cdot;\theta^{\mathrm{feat}}_{f_t''},w);\mathcal{P}_t^{\mathrm{test}})\right]\leq O\left(\frac{K\eta}{\sigma^2}+\frac{1}{\eta T}+\sqrt{\frac{V_{T,\ell}}{T\eta}}\right), \tag{15}$$

where $V_{T,\ell}:=\sum_{t=2}^{T}\sup_{w\in\mathcal{W}}|\ell(f(\cdot;\theta^{\mathrm{feat}}_{f_t''},w);\mathcal{P}_t^{\mathrm{test}})-\ell(f(\cdot;\theta^{\mathrm{feat}}_{f_{t-1}''},w);\mathcal{P}_{t-1}^{\mathrm{test}})|$, σ denotes the minimum singular value of the invertible confusion matrices $\{C_{f_1'',D_0'},\cdots C_{f_T'',D_0'}\}$ and K is the number of classes and the expectation is taken with respect to randomness in the revealed co-variates.

Proof Sketch: Recall that $\ell(f(\cdot;\theta^{\mathrm{feat}}_{f_t''},w);\mathcal{P}_t^{\mathrm{test}}):=E_{(x,y)\sim\mathcal{P}_t^{\mathrm{test}}}\ell_{\mathrm{ce}}\left(f(x|\theta^{\mathrm{feat}}_{f_t''},w),y\right)$.

This guarantee follows from the arguments in Bai et al. [2022] from two basic facts below:

1. The risk $\ell(f(\cdot;\theta^{\mathrm{feat}}_{f_t''},w);\mathcal{P}_t^{\mathrm{test}})$ is convex in w over a convex and compact domain $\mathcal{W}$.
2. It is possible to form unbiased estimates $\hat{G}_t(w)\in\mathbb{R}^K$ such that $E[\hat{G}_t(w)|S_{1:t-1}]=E_{((x,y)\sim\mathcal{P}_t^{\mathrm{test}})}\nabla_w\ell_{\mathrm{ce}}\left(f(x|\theta^{\mathrm{feat}}_{f_t''},w),y\right)$.

Hence we proceed to verify these two facts in our setup. Fact 1 is true because the cross-entropy loss is convex in any subset of the simplex and the last linear layer weights only defines an affine transformation which preserves convexity.

For fact 2, note that the f_t'' only uses the data until round $t-1$. So by the same arguments in Bai et al. [2022], using the BBSE estimator defined from the classifier f_t'', the unbiased estimate of risk gradient can be defined.

Let w_t be the weight of the last layer maintained by UOGD at round t. Let $u_{1:T}$ be any sequence in $\mathcal{W}$. Consequently we have for any round,

$$\begin{aligned}\ell(f^{\mathrm{uogd-ofu}};\mathcal{P}_t^{\mathrm{test}})-\ell(f(\cdot;\theta^{\mathrm{feat}}_{f_t''},u_t))&=\ell(f(\cdot;\theta^{\mathrm{feat}}_{f_t''},w_t)-\ell(f(\cdot;\theta^{\mathrm{feat}}_{f_t''},u_t)) && (16)\\ &\leq\langle\nabla_w\ell(f(\cdot;\theta^{\mathrm{feat}}_{f_t''},w_t),w_t-u_t\rangle && (17)\\ &=\langle E[\hat{G}_t(w_t)|S_{1:t-1}],w_t-u_t\rangle. && (18)\end{aligned}$$

Rest of the proof is identical to Bai et al. [2022].

Theorem 6 (Regret convergence for ATLAS-OFU)**.** *Let $f(\cdot;\theta^{\mathrm{feat}}_{f_t''},w)$ denote a network with the same feature extractor as that of f_t'' and a last linear layer with weight w. Let $f^{\mathrm{atlas-ofu}}=f(\cdot;\theta^{\mathrm{feat}}_{f_t''},w_t)$, where w_t is the weight maintained at round t by Algorithm 5. If we run Algorithm 1 with ATLAS in Bai et al. [2022] and set up the step size pool $\mathcal{H}=\{\eta_i=O\left(\frac{\sigma}{\sqrt{KT}}\right)\cdot 2^{i-1}|i\in[N]\}$ ($N=1+\lceil\frac{1}{2}\log_2(1+2T)\rceil$), then under the same assumptions as Lemma 1 in Bai et al. [2022], UOGD-OFU satisfies that*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\ell(f^{\mathrm{atlas-ofu}};\mathcal{P}_t^{\mathrm{test}})-\frac{1}{T}\sum_{t=1}^{T}\min_{w\in\mathcal{W}}\ell(f(\cdot;\theta^{\mathrm{feat}}_{f_{t+1}''},w);\mathcal{P}_t^{\mathrm{test}})\right]\leq O\left(\left(\frac{K^{1/3}}{\sigma^{2/3}}+1\right)\frac{V_{T,\ell}^{1/3}}{T^{1/3}}+\sqrt{\frac{K}{\sigma^2T}}\right), \tag{19}$$

where $V_{T,\ell}:=\sum_{t=2}^{T}\sup_{w\in\mathcal{W}}|\ell(f(\cdot;\theta^{\mathrm{feat}}_{f_t''},w);\mathcal{P}_t^{\mathrm{test}})-\ell(f(\cdot;\theta^{\mathrm{feat}}_{f_{t-1}''},w);\mathcal{P}_{t-1}^{\mathrm{test}})|$, σ denotes the minimum singular value of the invertible confusion matrices $\{C_{f_1'',D_0'},\cdots C_{f_T'',D_0'}\}$ and K is the number of classes and the expectation is taken with respect to randomness in the revealed co-variates.

The proof is similar to that of Theorem 5 and hence omitted.

Discussion about the assumption. In the theorems for UOGD and ATLAS, the definition of $V_{T,\ell}$ is shift severity from $\mathcal{P}_t^{\mathrm{test}}$. However, in the theorems for UOGD-OFU and ATLAS-OFU above, $V_{T,\ell}$ is shift severity from both $\mathcal{P}_t^{\mathrm{test}}$ and $\theta^{\mathrm{feat}}_{f_t''}$, which can be much larger. This might lead to harder convergence of the regret.

D Additional experiments

D.1 Experiment Set-Up

Dataset and shift process set-up. For online label shift, we evaluate the efficacy of our algorithm on CIFAR-10, which has 10 categories of images. We split the original train set of CIFAR-10 into the offline train and validation sets, which have 40,000 and 10,000 images respectively. At the online test stage, the unlabeled batches are sampled from the test set of CIFAR-10. For online generalized label shift, the offline train and validation sets are the same CIFAR-10 images, but the test unlabeled batches are sampled from CIFAR-10C. CIFAR-10C is the benchmark that has the same objects in CIFAR-10 but with multiple types of corruption. We experiment with three types of corruptions (i.e., domain shifts): Gaussian noise, Fog and Pixelate. Besides CIFAR-10 and CIFAR-10C, we also experiment with three additional datasets for the setting of online label shift and present the results in Section D.4. See more details of dataset set-up in Section D.3.

We follow Bai et al. [2022] and Baby et al. [2023] to simulate the online label distribution shift in two online shift patterns: Sinusoidal shift and Bernoulli shift. Given two label distribution vectors q and q', the label marginal distributions at time t is $q_t := \alpha_t q + (1 - \alpha_t)q'$. In Sinusoidal shift, $\alpha_t = \sin \frac{i\pi}{L}$ (periodic length $L = \sqrt{T}$, $i = t \bmod L$) while in Bernoulli shift, α_t is a random bit (either 0 or 1), where the bit switches $\alpha_t = \alpha_t - 1$ if the coinflip probability exceeds $p = \frac{1}{\sqrt{T}}$. The q and q' are $\frac{1}{K}(1, \cdots, 1)$ and $(1, 0, \cdots, 0)$ in the experiment. To sample the batch test data at time t, we first sample a batch of labels (not revealed to the learner) according to q_t. Then given each label we can sample an image from the test set, and collect this batch of images without labels as S_t. We experiment with $T = 1000$ and batch size $B = 10$ at each time step, following Baby et al. [2023].

Evaluation metric. We report the average error $\frac{1}{TB}\sum_{t=1}^{T}\sum_{x_t \in S_t} \mathbb{1}\left(f_t(x_t) \neq y_t\right)$, where $y_t \sim \mathcal{P}_t^{\text{test}}(y|x_t)$, to approximate $\frac{1}{T}\sum_{t=1}^{T} \ell(f_t; \mathcal{P}_t^{\text{test}})$ for the evaluation efficiency. This approximation is valid for large T due to its exponential concentration rate by the Azuma–Hoeffding inequality.

Online algorithms set-up. We perform an extensive evaluation of 6 OLS algorithms in the literature: FTFWH, FTH, ROGD, UOGD, ATLAS, and FLH-FTL. FTFWH is an empirical OLS proposed in Wu et al. [2021]. We further report the performance of OLS-OFU (Algorithm 1) on top of each OLS. OLS-OFU is implemented with 3 common SSL methods: rotation degree prediction, entropy minimization, and MoCo. Additionally, we report two baseline scores. The first, denoted as Base, uses the fixed pretrained model f_0 to predict the labels at all test time steps. The second is online feature updates (OFU) only, where at time step t we only update the features (Step 2 in Algorithm 1) without utilizing OLS algorithms.

D.2 Additional Experiments

How does OLS-OFU perform under high severity of domain shift in the setting of generalized online label shift? In Figure 2, it is clear that as the domain shift severity increases, OLS-OFU significantly enlarges the gap compared with OLS. However, it might be worth pointing out that neither OLS nor OLS-OFU are better than OFU. This is because the label shift assumption is violated so severely — even OFU cannot reduce the violation under an acceptable level, especially when the OLS module exists in OLS-OFU, as the adaptation to distribution shift is far off. For results involving higher levels of severity, different corruption types, and SSL techniques, please refer to Section D.7.

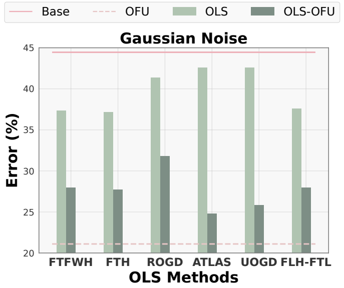


Figure 2: Results on CIFAR-10C for a high level of domain shift severity.

Does Equation 5 hold empirically? In Section C.1, we argued that when the inequality in Equation 5 holds, the loss of FLHFTL-OFU exhibits a tighter upper bound compared to FLHFTL. Figure 3 presents the RHS (corresponds to OLS) and LHS (corresponds to OLS-OFU with SSL loss as rotation degree prediction) of Equation 5. We perform the study over cross eight different settings, varying types of domain shift and online shift pattern, which empirically validates that OLS-OFU yields improvements on the *baseline* of the regret as shown in Equation 5. Section D.8 validates this inequality for other SSL techniques.

Does the order of prediction and update matter? In the default online distribution shift framework (Figure **??**), model updates occur after making predictions for samples at timestep t. This raises the question of whether the model should be updated before making predictions. We conducted empirical evaluations for both the "predict first" and "update first" approaches and found no compelling evidence to favor one over the other (additional results in Section D.9). However, it's noteworthy that within the "predict first" framework, OLS and OLS-OFU benefit from robust theoretical guarantees, hence we recommend the "predict first" approach in practical applications.

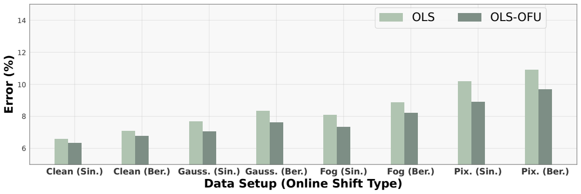


Figure 3: Empirical examination for the holdness of Equation 5. Clean denotes the experiment on CIFAR-10. Gaussion, Fog and Pixelate denote various domain shifts in CIFAR-10C. They are paired with two online shift patterns: Sinusoidal and Bernoulli.

D.3 Additional details of datasets

Severity of CIFAR-10C in the experiment. For each type of corruption in CIFAR-10C, we select a mild level and a high level of severity in the experiment section. Here we introduce the exact parameters of mild and high levels of severity for those corruptions. For Gaussian Noise, the severity levels for [mild, high] are [0.03, 0.07]. For Fog, the severity levels for [mild, high] are [(0.75,2.5), (1.5,1.75)]. For Pixelate, the severity levels for [mild, high] are [0.75, 0.65].

Details of additional datasets. In addition to CIFAR-10 and CIFAR-10C, we evaluate on three more datasets: STL10 [Coates et al., 2011], CINIC [Darlow et al., 2018], and EuroSAT [Helber et al., 2019]. Similar to CIFAR-10, we split the original train sets of these datasets into the train set and the validation set by the ratio $4:1$ and use the original test sets for sampling test images in the online test stage.

D.4 Results on Additional Datasets

Figure 4 shows the results for three additional datasets: STL10, CINIC, and EuroSAT, also with OLS-OFU (rotation degree prediction) under sinusoidal shift setting. In-line with CIFAR-10, we observe that OLS and OLS-OFU can perform better than Base and OFU, and OLS-OFU can outperform OLS. We find this pattern to be more consistent for EuroSAT and STL10 than CINIC.

D.5 More Results on CIFAR-10

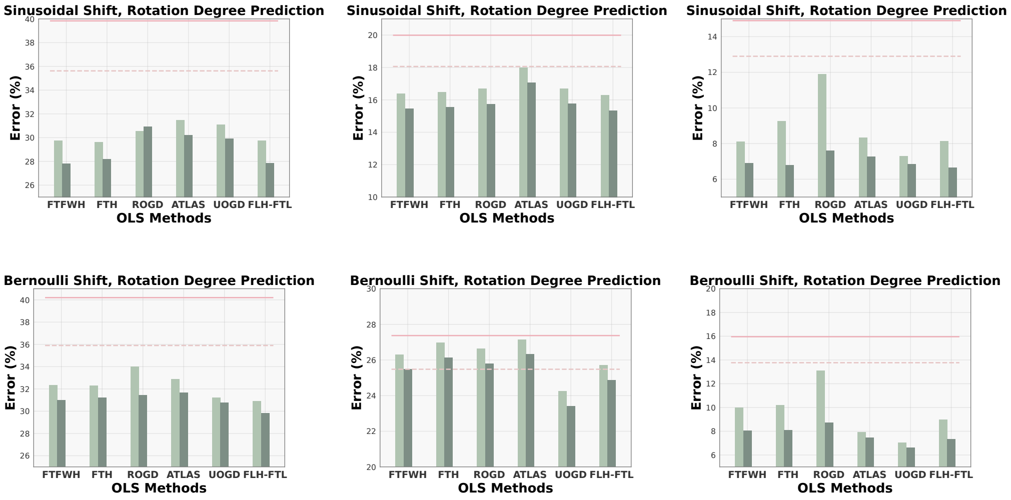
Figure 5 shows the results on CIFAR-10 for Bernoulli shift cross three SSL methods in OLS-OFU. Similar to Figure 1(a), OLS-OFU within all three SSL methods outperforms all baseline methods.

D.6 More Results on CIFAR-10C

We evaluate three SSL methods in OLS-OFU on CIFAR-10C for two online shift patterns. We pick moderately high severity levels for evaluating CIFAR-10C. In Figure 6 we can observe the consistent improvement from OLS to OLS-OFU and the SOTA performance of OLS-OFU.

D.7 More Results on CIFAR-10C with High Severity

We evaluate three SSL methods in OLS-OFU on CIFAR-10C with *high severity* for two online shift patterns. In Figure 7, we can observe when SSL in OLS-OFU is rotation degree prediction or MoCo, the improvement from OLS to OLS-OFU is very significant but OLS-OFU cannot outperform OFU. The conclusion is similar to the discussion for Figure 2 in Section 4. However, OLS-OFU with SSL entropy minimization has different behavior. When the corruption of CIFAR-10C becomes



(a) STL10

(b) CINIC

(c) EuroSAT

Figure 4: Results for additional datasets STL10, CINIC and EuroSAT.

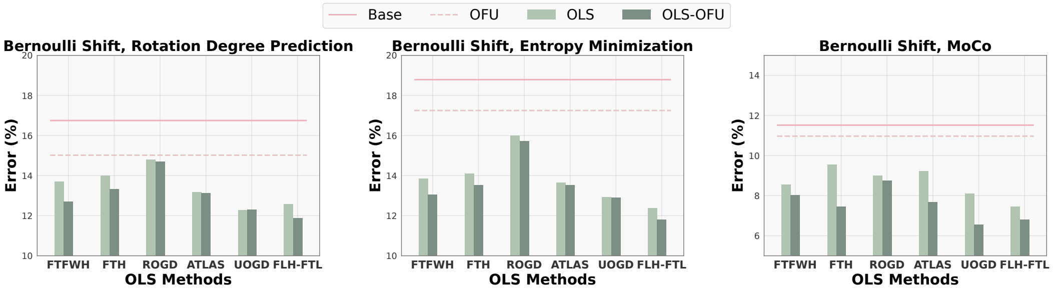


Figure 5: Results of Bernoulli shifts on CIFAR-10. OLS-OFU is evaluated with three SSL methods.

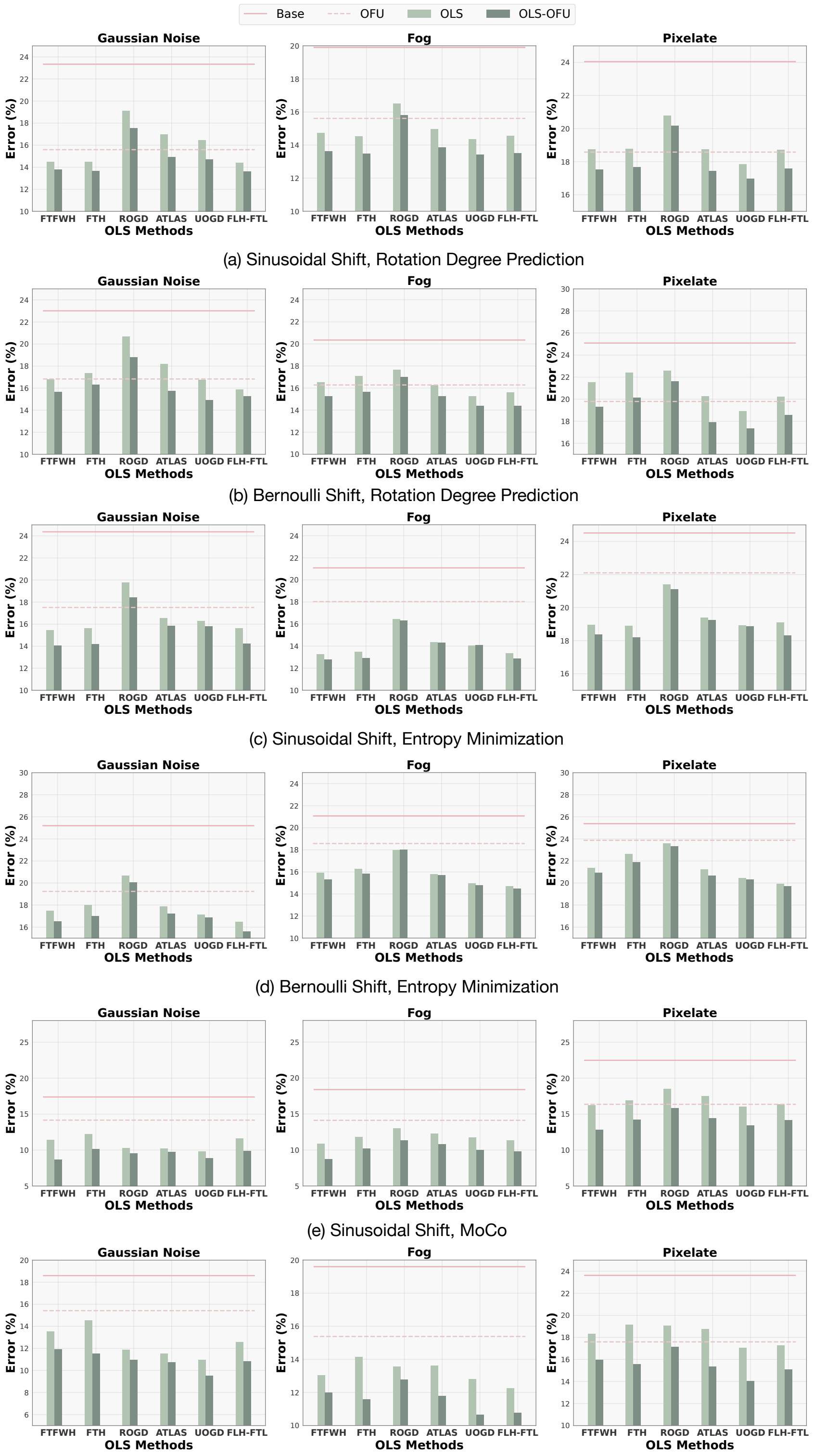
more severe, OLS-OFU entropy minimization shows less improvement from OLS, if we compare the transition: Figure 6 (clean CIFAR-10), Figure 6 (CIFAR-10C with mild severity), Figure 7 (CIFAR-10C with high severity). This suggests that rotation degree prediction and MoCo are more appropriate SSL to address the domain shift from CIFAR-10 to CIFAR-10C.

D.8 Empirical evaluation for Equation 5

Figure 8 shows the comparison between LHS (OLS-OFU) and RHS (OLS) of the inequality in Equation 5 when the SSL in OLS-OFU is entropy minimization or MoCo. Similar to what we observe in Figure 3, the inequality in Equation 5 holds cross 4 data settings and two online shift patterns when OLS-OFU is implemented with entropy minimization or MoCo.

D.9 Ablation for the order of updates and predictions

In the default framework of online distribution shift as shown in Figure **??**, the model updates happen after the predictions for the samples at time step t. We would like to see if the model updated before the predictions would bring benefit. Figure 9 shows the comparison between "predict first" and "update first". We can observe that there is no strong evidence to demonstrate the advantage of "predict first" or "update first" – the difference is indeed insignificant. However, because in the



(a) Sinusoidal Shift, Rotation Degree Prediction

(b) Bernoulli Shift, Rotation Degree Prediction

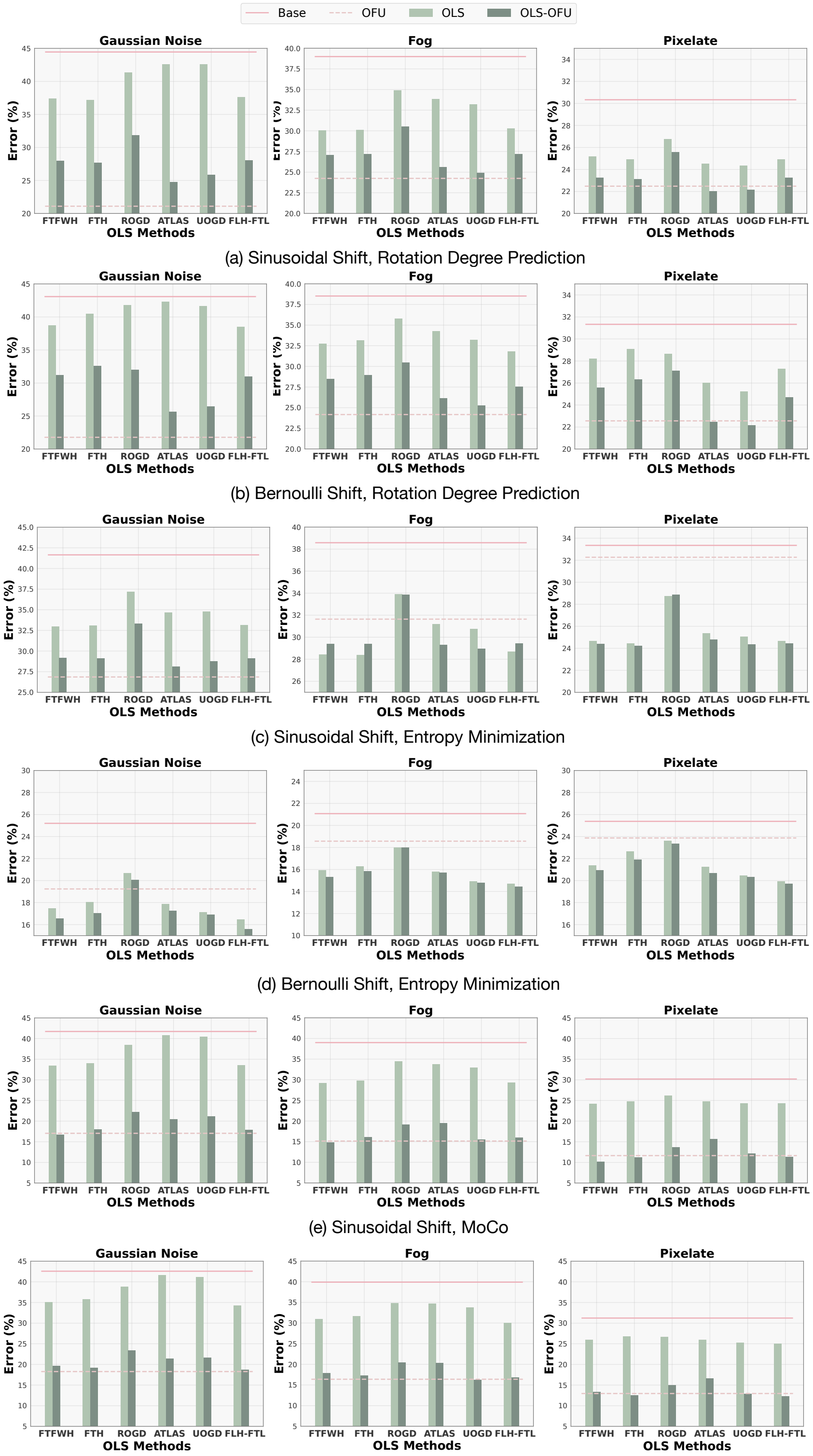
(c) Sinusoidal Shift, Entropy Minimization

(d) Bernoulli Shift, Entropy Minimization

(e) Sinusoidal Shift, MoCo

(f) Bernoulli Shift, MoCo

Figure 6: Results of two online shift patterns on CIFAR-10C and three SSL methods in OLS-OFU.



(a) Sinusoidal Shift, Rotation Degree Prediction

(b) Bernoulli Shift, Rotation Degree Prediction

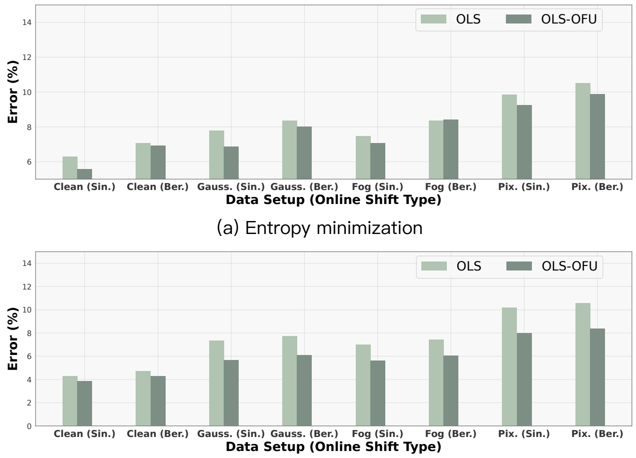
(c) Sinusoidal Shift, Entropy Minimization

(d) Bernoulli Shift, Entropy Minimization

(e) Sinusoidal Shift, MoCo

(f) Bernoulli Shift, MoCo

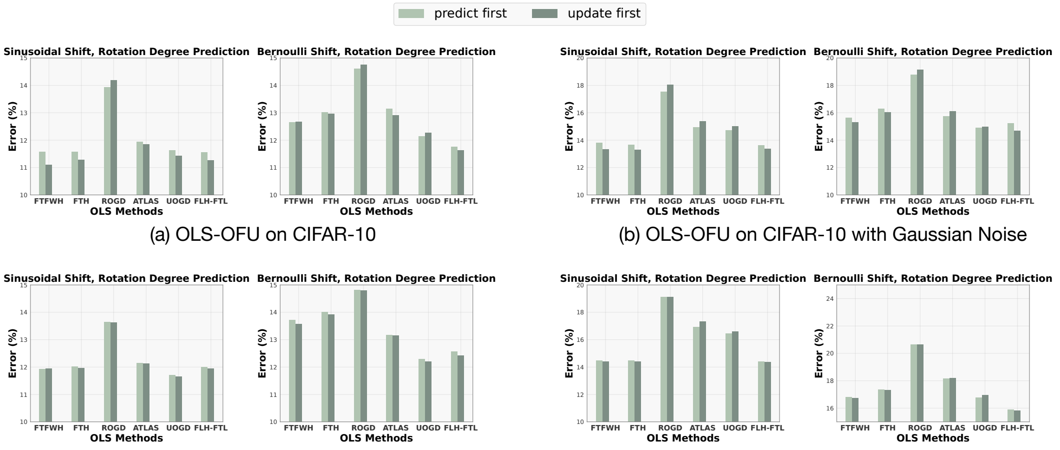
Figure 7: Results of two online shift patterns on CIFAR-10C (high severity) and three SSL methods in OLS-OFU.



(a) Entropy minimization

(b) MoCo

Figure 8: Sanity check for the inequality $\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\ell(g(\cdot; f''_t, q_t/q_0); \mathcal{P}_t^{\text{test}})\right] < \frac{1}{T}\sum_{t=1}^{T}\ell(g(\cdot; f_0, q_t/q_0); \mathcal{P}_t^{\text{test}})$ (Equation 5) when SSL in OLS-OFU is entropy minimization or MoCo.



(a) OLS-OFU on CIFAR-10

(b) OLS-OFU on CIFAR-10 with Gaussian Noise

(c) OLS on CIFAR-10

(d) OLS on CIFAR-10 with Gaussian Noise

Figure 9: Ablation for the order of updates and predictions.

framework of "predict first" OLS and OLS-OFU enjoy strong theoretical guarantees, we recommend "predict first" in practice.

D.10 Details of SSL Methods

Rotation degree prediction involves initially rotating a given image by a specific degree from the set $\{0, 90, 180, 270\}$ and the classifier is required to determine the degree by which the image has been rotated. It requires another network f^{deg} to predict the rotation degree, sharing the same feature extractor θ^{feat} as f_0 but with a different set of downstream layers. Its SSL loss $\ell_{\text{ssl}}(S; f)$ is defined as $\sum_{x\in S}\ell_{ce}(f^{\text{deg}}(R(x,i)), i)$, where i is an integer uniformly sampled from $[4]$, and $R(x,i)$ is to rotate x with degree $\text{DL}[i]$ from a list of degrees $\text{DL} = [0, 90, 180, 270]$.

Entropy minimization utilizes a minimum entropy regularizer, with the motivation that unlabeled examples are mostly beneficial when classes have a small overlap. $\ell_{\text{ssl}}(S; f)$ would be the entropy $\sum_{x\in S}\sum_{k=1}^{K} f(x)_k \log f(x)_k$.

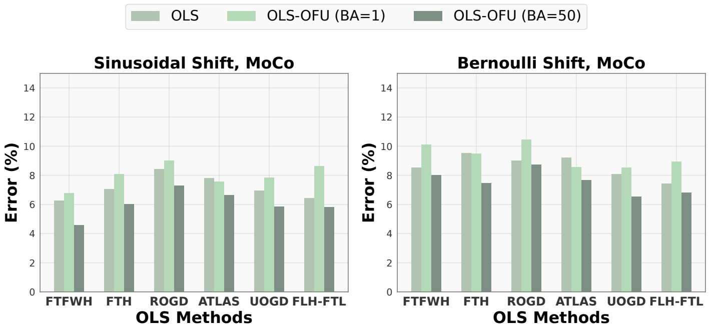


Figure 10: Evaluating OLS-OFU with different batch accummulations for MoCo on CIFAR-10.

Moreover, MoCo is a more advanced representation learning technique, using a query and momentum encoder to learn representations from unlabeled data by maximizing the similarity of positive pairs and minimizing the similarity of negative pairs. The SSL loss would be a contrastive loss (InfoNCE) where the positive example x' is an augmented version of x and other samples in the same time step can be the negative examples. However, the batch size for a time step is small, e.g. 10 in our experiment. MoCo updates with such a small batch won't work. Thus, we experiment with MoCo by applying a batch accumulation strategy, which we introduce next.

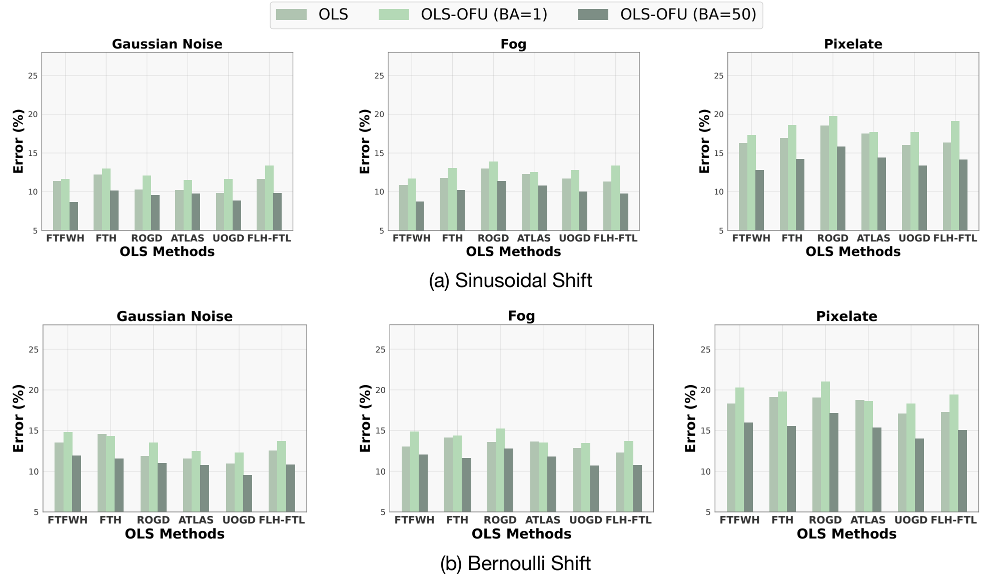
D.10.1 Batch accumulation for MoCo

Prior test-time training (rotation degree prediction, entropy minimization) methods operate with the OFU framework easily, with feature extractor updates taking place in each time step. Rotation degree prediction originates as a self-supervised training method [Gidaris et al., 2018], thus naturally we evaluate more recent self-supervised training methods, namely MoCo [He et al., 2019, Chen et al., 2020b, 2021]. No prior work shows how to use MoCo (or self-supervised learning in general) in a test-time training setting. Given that self-supervised training is sensitive to batch size, the intuition is that a larger batch size (much larger than the number of online samples available per time step) is required to perform a valid gradient update for a MoCo checkpoint. As such, we evaluated a *batch accumulation* strategy OLS-OFU (BA=τ), where we continue evaluating online samples per time step, but only perform the online feature update at every τ steps (having accumulated the online samples throughout the τ steps in one batch). In particular, we perform feature extractor update every $\tau = 50$ steps (for 1000 steps, feature update occurs 20 times, online samples evaluation occurs 1000 times), evaluating with 10 online samples per time step, using a smaller learning rate (0.0005) but test-time train with 10 epochs. Notice that $\tau = 1$ is the default setting in Algorithm 1. OLS-OFU with MoCo presented in Figure 1, Figure 5 and Figure 6 is equivalent to OLS-OFU (BA=50).

To show the necessity of large τ, we evaluate both OLS-OFU (BA=1) and OLS-OFU (BA=50) on CIFAR-10 (Figure 10) and CIFAR-10C (Figure 11). Firstly, we can observe that OLS-OFU (BA=1) is even worse than OLS. We hypothesize this is because small batch size of MoCo will hurt the performance and larger batch size in MoCo is necessary. Hence, we increase τ from 1 to 50 and then we can observe the significant improvement from OLS-OFU (BA=1) to OLS-OFU (BA=50). Now, OLS-OFU (BA=50) can outperform OLS.

D.11 Self-training

Pseudo labelling [Lee, 2013], a common self-training technique, generates pseudo labels for unlabelled data and uses them to update the model. Though we are not able to use ground-truth labels to compute feature extractor updates, we can use the model at time t to make predictions with respect to the online samples at time t, and train on the inputs with their assigned (pseudo) labels. An issue that arises in self-training is confirmation bias, where the model repeatedly overfits to incorrect pseudo-labels. As such, different methods can be used to select which samples will be pseudo-labelled and used in updating the model, e.g. using data augmentation [Arazo et al., 2020], using regularization



(a) Sinusoidal Shift

(b) Bernoulli Shift

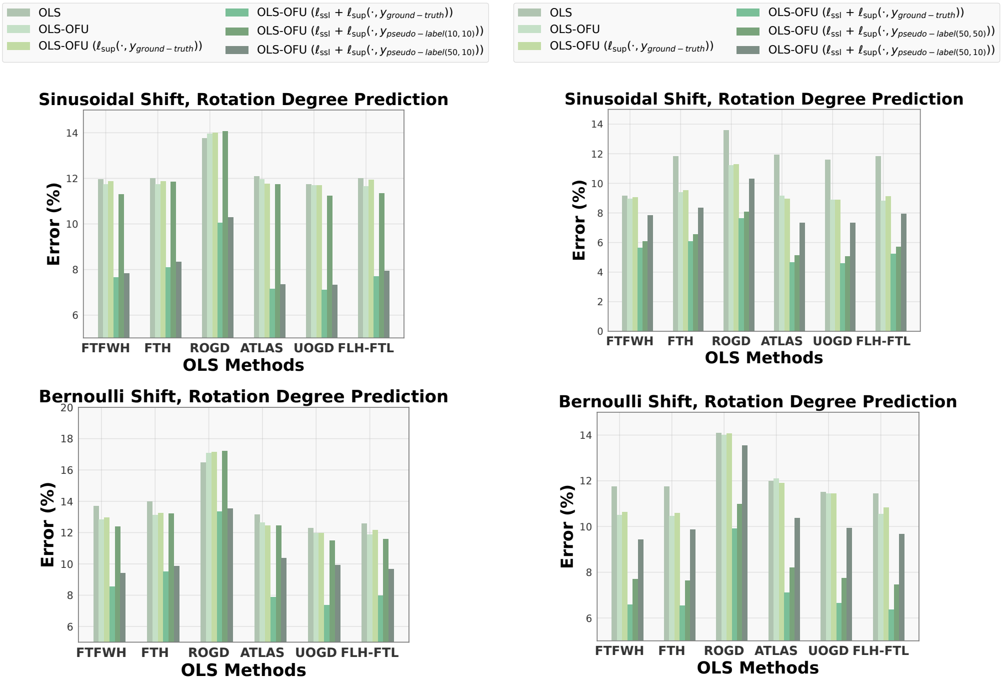
Figure 11: Evaluating OLS-OFU with different batch accummulations for MoCo on CIFAR-10C.

to induce confident low-entropy pseudo-labelling [Grandvalet and Bengio, 2004a], using softmax thresholds to filter out noisy low-confidence predictions [Xie et al., 2020]. We make use of ensembles to identify noisy low-confidence/entropy pseudo-label predictions, though other various alternatives can also be used. In addition to OLS and OLS-OFU, we highlight the methods under comparison:

- *OLS-OFU ($\ell_{\text{sup}}(\cdot, y_{\text{ground-truth}})$)*: Instead of computing pseudo-labels, we make use of the correct ground-truth labels $y_{\text{ground-truth}}$. Recall ℓ_{sup} is the supervised learning loss. We update the feature extractor with the supervised loss w.r.t. ground-truth labels $\ell_{\text{sup}}(\cdot, y_{\text{ground-truth}})$.
- *OLS-OFU (ℓ_{ssl} + $\ell_{\text{sup}}(\cdot, y_{\text{ground-truth}})$)*: Instead of computing pseudo-labels, we make use of the correct ground-truth labels $y_{\text{ground-truth}}$. Recall ℓ_{ssl} and ℓ_{sup} are the self-supervised and supervised learning losses respectively. We update the feature extractor with both the self-supervised loss ℓ_{ssl} as well as the supervised loss w.r.t. ground-truth labels $\ell_{\text{sup}}(\cdot, y_{\text{ground-truth}})$.
- *OLS-OFU (ℓ_{ssl} + $\ell_{\text{sup}}(\cdot, y_{\text{pseudo-label(\#samples=, \#FU-samples=)}})$)*: Recall ℓ_{ssl} and ℓ_{sup} are the self-supervised and supervised learning losses respectively. We compute pseudo-labels $y_{\text{pseudo-label}})$, and update the feature extractor with both the self-supervised loss ℓ_{ssl} as well as the supervised loss w.r.t. pseudo-labels $\ell_{\text{sup}}(\cdot, y_{\text{pseudo-label}})$.

How to compute pseudo-labels? We now describe the procedure to compute pseudo-labels for $\ell_{\text{sup}}(\cdot, y_{\text{pseudo-label(\#samples=, \#FU-samples=)}})$. The seed used to train our model is 4242, and we train an additional 4 models on seeds 4343, 4545, 4646, 4747. With this ensemble of 5 models, we keep sampling inputs at each online time step until we have `#FU-samples` samples, or we reach a limit of `#samples` samples. We accept an input when the agreement between the ensembles exceeds a threshold $e = 1.0$ (i.e. we only accept samples where all 5 ensembles agree on the label of the online sample). In the default online learning setting, there are only `#samples=10`, and therefore there may not be enough accepted samples to perform feature update with, thus we evaluate with a continuous sampling setup, where we sample `#samples=50` (and evaluate on all these samples), but only use the first 10 samples (`#FU-samples=10`) to perform the feature extractor update.

Results on pseudo-labelling. First, we find that *OLS-OFU (ℓ_{ssl} + $\ell_{\text{sup}}(\cdot, y_{\text{ground-truth}})$)* attains the lowest error and is the lower bound we are attaining towards. Evaluating *OLS-OFU (ℓ_{ssl} + $\ell_{\text{sup}}(\cdot, y_{\text{pseudo-label(\#samples=10, \#FU-samples=10)}})$)*, we find that the performance does not outperform OLS-OFU, and is not near *OLS-OFU (ℓ_{ssl} + $\ell_{\text{sup}}(\cdot, y_{\text{ground-truth}})$)*. If we set the threshold e too high, there



(a) number of online samples: 10

(b) number of online samples: 50

Figure 12: Results on pseudo-labelling.

may not be enough online samples to update the feature extractor. If we set the threshold e too low, there may be too many incorrect labels and we incorrectly update our feature extractor. As such, we would like to sample more inputs at each online time step such that we can balance this tradeoff. We sample `#samples=50` at each online time step, and update with `#FU-samples` $\leq$ `10`. For fair comparison, we also show the comparable methods in both `#samples=10, #FU-samples=10` and `#samples=50, #FU-samples=50` settings.

With this sampling setup, we find that *OLS-OFU* $(\ell_{\text{ssl}} + \ell_{\text{sup}}(\cdot, y_{\text{pseudo-label(\#samples=50, \#FU-samples=10)}}))$ can outperform both *OLS-OFU (`#samples=10`)* and *OLS-OFU (`#samples=50`)*. Though it does not exceed neither *OLS-OFU* $(\ell_{\text{ssl}} + \ell_{\text{sup}}(\cdot, y_{\text{ground-truth}}))$ for `#samples=10` nor `#samples=50`, it lowers the gap considerably.