Online Maximization of Non-Decomposable Test and Population Utilities

Wojciech Kotłowski
Poznan University of Technology
wkotlowski@cs.put.poznan.pl

Marek Wydmuch
Snowflake AI Research /
Poznan University of Technology
marek.wydmuch@snowflake.com

Krzysztof Dembczyński
Yahoo Research /
Poznan University of Technology
kdembczynski@cs.put.poznan.pl

Abstract

We study sequential learning for classification metrics that are non-decomposable across instances and are general functions of the confusion matrix, such as the Jaccard index and the F-measure. The learner observes instances sequentially, makes irrevocable predictions, and is evaluated by the final empirical performance metric, leading to an online-regret objective. After observing the sequence, the learner also returns a classifier for future population use, leading to a population-regret objective. This single protocol connects two central frameworks for optimizing non-decomposable metrics: Expected Test Utility (ETU) and Population Utility (PU). For smooth concave utilities of the confusion matrix, we show that the ETU benchmark is controlled, in expectation, by the population PU comparator, and that an algorithm with small online regret also yields PU guarantees by an online-to-batch conversion. We then turn to non-concave linear-fractional metrics. We devise a general stochastic Dinkelbach-root method that tracks the relevant metric parameter online and gives online and population regret of order $n^{-1/2}$, up to logarithmic factors, with explicit dependence on conditional-probability estimation error. Our empirical studies evaluate these methods on benchmark datasets.

1 Introduction

In many classification tasks, the objective that matters operationally is not the average loss of isolated predictions, but a utility metric obtained after all decisions on an evaluation collection have been pooled. Retrieval and screening systems are often tuned for precision, recall, or the F-measure [Lewis, 1995]; segmentation and set-prediction tasks use the Jaccard index (also known as intersection over union) [Jaccard, 1901]; and imbalanced classification benchmarks commonly report balanced accuracy, geometric mean [Drummond and Holte, 2005, Wang and Yao, 2012, Menon et al., 2013, Cao et al., 2019], or Matthews correlation coefficient [Baldi et al., 2000]. These utilities are non-decomposable over instances and are general functions of the aggregate confusion matrix. As a result, the contribution of a prediction cannot be assigned independently at the moment it is made: it depends on the other predictions, the realized label composition, and the normalization induced by the final utility.

The literature has developed two standard statistical viewpoints on this problem. In the *Population Utility* (PU) framework, the goal is to construct a classifier whose *population confusion matrix* has high utility [Ye et al., 2012]. This is the natural deployment objective when the learned classifier will

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

be used on future examples. In the *Expected Test Utility* (ETU) framework, the utility is evaluated on an *empirical confusion matrix* over a finite test set and then averaged over the randomness in the labels, and possibly also the test inputs [Lewis, 1995, Jansche, 2007, Ye et al., 2012]. PU and ETU differ in the order in which the expectation, the metric, and the optimization over classifiers are taken. They are asymptotically close under standard conditions [Ye et al., 2012, Dembczynski et al., 2017], but at a finite horizon they can lead to very different benchmarks [Schultheis et al., 2023, 2024].

This paper studies these objectives in a general online prediction protocol, which covers binary, multi-class, and multi-label classification setups. Following [Kotlowski et al., 2024], the algorithm observes a sequence of instances, drawn *i.i.d.* from some underlying distribution. At a given round t, it observes the input vector, predicts the output (label vector), and then observes the true label, which may or may not be used in the state update. After n rounds, the online sequence is scored by applying the target utility to the empirical confusion matrix of all labels and predictions. This creates a challenge that motivates the study of online maximization of non-decomposable utilities: the algorithm is evaluated by a nonlinear function of the entire trajectory, but each prediction must be made immediately and cannot be revised once later examples reveal that a different tradeoff would have been better.

Our protocol has two outputs. First, the online predictions themselves are evaluated by ETU (online) regret against the expected-utility-maximizing prediction sequence. Second, the trajectory may be converted into a classifier whose future population performance is measured by PU (population) regret. This "online-to-batch" viewpoint is more than a technical convenience: it lets us ask whether an algorithm that behaves well on the current test sequence also learns enough about the metric-specific decision rule to return a useful classifier.

As in the original framework of [Kotlowski et al., 2024], we do not restrict the predictor to a parametric class such as linear models. Instead, the algorithm is given a conditional probability estimator (CPE), which returns an estimate of the conditional label distribution at the currently observed input. This modeling choice is natural in batch optimization of non-decomposable metrics, where optimal rules are usually obtained by turning conditional probabilities into cost-sensitive decisions. For decomposable losses, such a reduction would nearly solve the problem pointwise. For non-decomposable utilities, the decisions on different examples are coupled through the final confusion matrix, so the cost vector used at the current example depends on the current state of the online trajectory and on the metric being optimized.

We first revisit smooth concave utilities of the confusion matrix. In this setting, the OMMA algorithm of Kotlowski et al. [2024] predicts by linearizing the utility at the current empirical confusion matrix and solving the induced cost-sensitive classification problem. Its original guarantee is stated against a PU comparator. We show that for concave utilities this comparator is also sufficient for the ETU benchmark: the expected utility of the ETU prediction sequence is controlled by the PU value. As a result, the same linearization idea controls online ETU regret, and a randomized online-to-batch classifier obtained from the online trajectory has low PU regret of order $O(n^{-1/2})$.

Next we go beyond concavity. Many of the most common confusion-matrix metrics are linear-fractional (LF), being ratios of affine functions of the confusion matrix; this class includes precision, recall, F-measure, the Jaccard index, and the class-wise rates used in balanced and macro-averaged criteria. Such utilities need not be concave, and in this case the finite-test benchmark can genuinely exploit the realized input distribution. We handle this class with the Dinkelbach transform [Dinkelbach, 1967], a classical tool from fractional programming [Schaible and Ibaraki, 1983, Stancu-Minasian, 1997], also used in the previous papers on the batch LF analysis [Narasimhan et al., 2015a, 2024, Zhang and Agarwal, 2024, Mao et al., 2025]. The transform represents the optimal ratio as the root of a one-dimensional map. Each candidate root induces a linear cost-sensitive prediction problem, and the root can be tracked online by stochastic approximation [Robbins and Monro, 1951, Kushner and Yin, 2003, Zhang et al., 2018]. The resulting algorithm, called Stochastic Online Linear-fractional Optimization (SOLO), achieves ETU regret of order $\tilde{O}(n^{-1/2})$, and its final prediction rule can be used as a classifier that has PU regret of the same order. The analysis is different from that for concave utilities: it relies on the strong-sign structure of the Dinkelbach root, stochastic approximation for tracking this root online, and a smoothness argument that converts root error into ETU and PU regret.

Our contribution. We first formulate a sequential learning framework for non-decomposable utilities that contains both ETU regret for the online predictions and PU regret for a returned classifier.

We then strengthen the analysis of [Kotlowski et al., 2024] for smooth concave utilities by relating the ETU benchmark to the PU value and deriving online regret and online-to-batch PU guarantees. Next, for LF metrics, we propose SOLO, a metric-optimization SGD method that tracks a Dinkelbach root online. Like OMMA, its predictions are cost-sensitive and based on the current CPE, but its state and analysis are one-dimensional and ratio-specific. We prove $\tilde{O}(n^{-1/2})$ ETU and PU regret bounds for linear-fractional metrics; the one-dimensional strong-sign condition required for stochastic approximation follows automatically from denominator positivity. Finally, we show that the $O(n^{-1/2})$ PU rate for a returned classifier is unimprovable in general, even when the algorithm knows the exact conditional label distribution.

Related work. The batch theory of complex performance metrics is extensive, covering binary classification [Ye et al., 2012, Koyejo et al., 2014, Narasimhan et al., 2014, Dembczynski et al., 2017, Singh and Khim, 2022, Han et al., 2024, Mao et al., 2025], multi-class classification [Narasimhan et al., 2015b, 2024, Zhang and Agarwal, 2024], and multi-label classification [Waegeman et al., 2014, Koyejo et al., 2015, Kotłowski and Dembczyński, 2017, Mao et al., 2024]. Many batch methods reduce metric optimization to finding an appropriate cost-sensitive classifier, threshold, or mixture of such classifiers, often through Frank-Wolfe or related first-order methods. The recent ETU work on generalized test utilities [Schultheis et al., 2023] emphasizes that optimal predictions can depend on the entire set of test inputs.

Online work on non-decomposable metrics is much more limited. One line of research focuses on optimization of convex surrogates for structured prediction or for rate-based criteria using online updates [Joachims, 2005, Kar et al., 2014, Narasimhan et al., 2015a]. These methods optimize a surrogate objective over a parametric class, so the surrogate optimum and the target metric optimum need not coincide. Another line of research exploits the special structure of the F-measure. Busa-Fekete et al. [2015] jointly learn a CPE and adapt the F-measure threshold, establishing consistency but not a finite-time regret rate. The algorithm of Zhang et al. [2018] is closer in spirit to our LF method: it simultaneously learns a conditional probability model and an optimal threshold, and it obtains a finite-sample rate by casting binary F-measure thresholding as stochastic optimization of a one-dimensional strongly convex objective. This construction is, however, metric-specific. It relies on the special characterization of the F-measure optimum by a posterior-probability threshold, and its guarantee is a threshold-convergence and population-consistency statement rather than an ETU/PU regret bound for a general utility class. The closest general online result is the OMMA algorithm of Kotlowski et al. [2024], which provides fast regret bounds for smooth concave utilities by repeatedly linearizing the metric at the current confusion matrix. We use that mechanism as one component of the present paper, but change both the benchmark and the utility class: concave utilities are connected to ETU and PU regret, while LF utilities require a separate ratio-root analysis.

2 Problem setup

Let $\boldsymbol{x} \in \mathcal{X}$ be the input instance and $\boldsymbol{y} \in \mathcal{Y} \subseteq \{0,1\}^m$ the output label vector, jointly distributed according to $(\boldsymbol{x}, \boldsymbol{y}) \sim \mathbb{P}$. We let $\mathbb{P}_{\mathcal{X}}$ denote the marginal distribution of $\boldsymbol{x}$, and $\boldsymbol{\eta}(\boldsymbol{x})$ the *label conditional expectation*, $\boldsymbol{\eta}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{y}|\boldsymbol{x}}[\boldsymbol{y}]$. In *multi-class classification*, the label vector $\boldsymbol{y}$ is a one-hot encoding of one out of m classes, $\mathcal{Y} = \{\boldsymbol{y} \in \{0,1\}^m \colon \sum_j y_j = 1\}$, and $\boldsymbol{\eta}(\boldsymbol{x}) = (\eta_1(\boldsymbol{x}), \ldots, \eta_m(\boldsymbol{x}))$ becomes the label conditional distribution, with $\eta_j(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{y}|\boldsymbol{x}}[y_j] = \mathbb{P}(y_j = 1|\boldsymbol{x})$. In *multi-label classification*, $\boldsymbol{y} \in \{0,1\}^m$ denotes a vector of relevant labels, and $\boldsymbol{\eta}(\boldsymbol{x})$ is a vector of *label marginal conditional probabilities*, $\eta_j(\boldsymbol{x}) = \mathbb{P}(y_j = 1|\boldsymbol{x})$. Note that in the multi-label case, $\boldsymbol{\eta}(\boldsymbol{x})$ *is not* the conditional distribution $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})$ (which is a *joint* distribution over $\{0,1\}^m$), but the algorithms will always operate only on the marginals. We will sometimes use $\boldsymbol{y} \sim \boldsymbol{\eta}(\boldsymbol{x})$ to denote a multi-label vector drawn from $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})$ with marginals $\boldsymbol{\eta}(\boldsymbol{x})$ if no other property of the distribution matters.

We consider an online stochastic setting in which the algorithm observes a sequence of instances[1] $(\boldsymbol{x}, \boldsymbol{y})^n = (\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)$, drawn *i.i.d.* from $\mathbb{P}$. At each iteration $t = 1, 2, \ldots, n$, after receiving input $\boldsymbol{x}_t$ drawn from $\mathbb{P}_{\mathcal{X}}$, the algorithm makes a prediction $\widehat{\boldsymbol{y}}_t$, and next observes the true output $\boldsymbol{y}_t$ drawn from $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x}_t)$. We assume that the algorithm has access to a *conditional probability estimator* (CPE) $\widehat{\boldsymbol{\eta}}_t$, which at trial t returns an estimate $\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t)$ of the true conditional

[1] In the paper, we use $\boldsymbol{v}^t$ to denote a sequence $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t)$ and extend this convention with $f(\boldsymbol{v}^t)$ denoting $(f(\boldsymbol{v}_1), \ldots, f(\boldsymbol{v}_t))$ and $(\boldsymbol{x}, \boldsymbol{y})^t$ denoting $(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_t, \boldsymbol{y}_t)$.

expectation $\boldsymbol{\eta}(\boldsymbol{x}_t)$, which the algorithm will use for issuing its prediction $\widehat{\boldsymbol{y}}_t$. We put the time subscript in $\widehat{\boldsymbol{\eta}}_t$ as we allow the CPE to change over time (e.g., it can be produced by an external online algorithm run on the same sequence of instances); we assume, however, that $\widehat{\boldsymbol{\eta}}_t$ can only depend on the observed data $\boldsymbol{x}_t$, $(\boldsymbol{x}, \boldsymbol{y})^{t-1}$ (i.e., it does not depend on $\boldsymbol{y}_t$ and all the future instances). The CPE could be, for instance, a neural network or boosted decision trees trained (possibly online) by minimizing some proper scoring loss. In this work, we do not focus on the way the CPE is learned, and all our bounds on the performance of the algorithm will depend on the estimation error of $\widehat{\boldsymbol{\eta}}_t$ with respect to the true conditional expectation $\boldsymbol{\eta}_t$. Thus, our focus is the metric-optimization decision layer, rather than learning or calibrating the CPE. We outline the online protocol in Figure 1.

For any label-prediction pair $(\boldsymbol{y}, \widehat{\boldsymbol{y}})$, let $\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}})$ denote its *single-example confusion matrix*. In multi-class classification, it is an $m \times m$ matrix $\boldsymbol{C}$ with entries defined as $C_{ij} = y_i \widehat{y}_j \in \{0, 1\}$, while in multi-label classification, it is an $m \times 2 \times 2$ tensor $\boldsymbol{C}$, where $C_{juv} = \mathbf{1}\{y_j = u\}\mathbf{1}\{\widehat{y}_j = v\} \in \{0, 1\}$, with $j \in \{1, \dots, m\}$ and $u, v \in \{0, 1\}$, and $\mathbf{1}\{\cdot\}$ being an indicator function. Given a label sequence $\boldsymbol{y}^n = (\boldsymbol{y}_1, \dots, \boldsymbol{y}_n)$ and a prediction sequence $\widehat{\boldsymbol{y}}^n = (\widehat{\boldsymbol{y}}_1, \dots, \widehat{\boldsymbol{y}}_n)$, the *empirical confusion matrix* $\boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n) = \frac{1}{n}\sum_{t=1}^{n} \boldsymbol{C}(\boldsymbol{y}_t, \widehat{\boldsymbol{y}}_t)$ is the average of these single-example contributions. In the multi-class case, its (i, j)-entry contains the fraction of times an instance from class i was predicted as being in class j. In the multi-label case, the empirical confusion matrix is a sequence of m binary confusion matrices, one such matrix for each label. In this work, we also evaluate the population version of the confusion matrix for *randomized classifiers* $\boldsymbol{h} \colon \mathcal{X} \to \Delta(\mathcal{Y})$, where $\Delta(\mathcal{Y})$ is a set of distributions over $\mathcal{Y}$; its *population confusion matrix* is defined as $\boldsymbol{C}(\boldsymbol{h}) = \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \sim \mathbb{P}, \widehat{\boldsymbol{y}} \sim \boldsymbol{h}(\boldsymbol{x})}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}})]$. In a given problem, we let $\mathcal{C}$ denote the convex hull of all achievable confusion matrices, that is, the set of all confusion matrices that can be formed from all allowed label and prediction vectors, as well as any of their convex combinations.

Utility metrics. We maximize metrics that do not decompose into a sum over instances but are functions $\psi : \mathcal{C} \to \mathbb{R}$ of the empirical confusion matrix. We focus on two classes. The first consists of differentiable utilities $\psi(\boldsymbol{C})$ *concave* in $\boldsymbol{C}$. The second consists of *linear-fractional* (LF) utilities,

$$\psi(\boldsymbol{C}) = \frac{N(\boldsymbol{C})}{D(\boldsymbol{C})}, \qquad N(\boldsymbol{C}) = \boldsymbol{A} \cdot \boldsymbol{C}, \qquad D(\boldsymbol{C}) = \boldsymbol{B} \cdot \boldsymbol{C}, \tag{1}$$

where $\boldsymbol{A}, \boldsymbol{B}$ have the same shape as $\boldsymbol{C}$. LF utilities are sometimes defined with additional constants in the numerator and denominator (e.g., regularization), but due to the sum-to-one constraints on $\boldsymbol{C}$, these constants can be incorporated into $\boldsymbol{A}$ and $\boldsymbol{B}$ by adding a constant to each of their entries. Examples of common utility metrics are summarized in Appendix A. The main difficulty in maximizing ψ comes from the fact that the algorithm is only evaluated at the end of the sequence by a function of all the labels and predictions, while each prediction $\widehat{\boldsymbol{y}}_t$ must be made immediately upon observing $\boldsymbol{x}_t$, and cannot be changed in the future.

ETU and PU regret. The online prediction sequence is evaluated by the expected empirical utility $\mathbb{E}[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n))]$. Since, without assumptions on the data distribution, the utility cannot be meaningfully bounded in absolute terms, we compare the performance of the algorithm to the best sequence of predictions that could have been obtained if the inputs $\boldsymbol{x}^n$ were known in hindsight, that is, to the expected utility maximizer (ETU):

$$\boldsymbol{y}^{\star n} = \boldsymbol{y}^{\star n}(\boldsymbol{x}^n) \in \operatorname*{argmax}_{\boldsymbol{z}^n} \mathbb{E}_{\boldsymbol{y}^n | \boldsymbol{x}^n} \left[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{z}^n))\right]. \tag{2}$$

We note that the expectation over the labels is essential in the definition above: otherwise, letting the optimal prediction depend on the observed labels $\boldsymbol{y}^n$ would lead to the trivial and unreachable solution

---

for $t = 1, \dots, n$ **do**
  Observe input instance $\boldsymbol{x}_t$ drawn from $\mathbb{P}_{\mathcal{X}}$
  Receive conditional probability estimate $\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t)$
  Predict label $\widehat{\boldsymbol{y}}_t$ based on $\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t)$ and the current metric state
  Receive true label $\boldsymbol{y}_t$ drawn from $\mathbb{P}(\cdot|\boldsymbol{x}_t)$
Evaluate by $\psi(\boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n))$, and return a randomized classifier $\widehat{\boldsymbol{h}}_n \colon \mathcal{X} \to \Delta(\mathcal{Y})$.

Figure 1: The online prediction protocol

Algorithm 1 OMMA for concave utilities (label-adaptive and label-oblivious versions)

Initialize $\boldsymbol{C}_0 \in \mathcal{C}$.
for $t = 1, \ldots, n$ **do**
  Observe $\boldsymbol{x}_t$ and predict $\widehat{\boldsymbol{y}}_t \in \operatorname{argmax}_{\widehat{\boldsymbol{y}}} \nabla\psi(\boldsymbol{C}_{t-1}) \cdot \mathbb{E}_{\boldsymbol{y}\sim\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t)}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}})]$.
  Observe $\boldsymbol{y}_t$ and define $\boldsymbol{\Delta}_t^{\mathrm{ad}} = \boldsymbol{C}(\boldsymbol{y}_t, \widehat{\boldsymbol{y}}_t)$ and $\boldsymbol{\Delta}_t^{\mathrm{obl}} = \mathbb{E}_{\boldsymbol{y}\sim\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t)}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}}_t)]$.
  Update $\boldsymbol{C}_t = \frac{t-1}{t}\boldsymbol{C}_{t-1} + \frac{1}{t}\boldsymbol{\Delta}_t$ with $\boldsymbol{\Delta}_t = \boldsymbol{\Delta}_t^{\mathrm{ad}}$ (adaptive) or $\boldsymbol{\Delta}_t = \boldsymbol{\Delta}_t^{\mathrm{obl}}$ (oblivious)

$\boldsymbol{y}^{\star n} = \boldsymbol{y}^n$. Let $\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n) = \mathbb{E}_{\boldsymbol{y}^n|\boldsymbol{x}^n}[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{y}^{\star n}))]$ denote the attained maximum expected utility in (2), where we explicitly indicate its dependence on the input sample $\boldsymbol{x}^n$. The corresponding *ETU (online) regret* is then defined as

$$R_n^{\mathrm{ETU}} := \mathbb{E}[\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n)] - \mathbb{E}[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n))]. \tag{3}$$

We analyze two versions of the online algorithm. The *label-adaptive* version may use past labels through the current empirical confusion matrix, that is at any time t, $\widehat{y}_t$ depends on $\boldsymbol{x}^t$ and $\boldsymbol{y}^{t-1}$. The *label-oblivious* version uses the inputs and CPE values but does not update its metric state with observed labels; that is, $\widehat{y}_t$ only depends on $\boldsymbol{x}^t$. While the label-adaptive algorithm is potentially more powerful, as it can adapt its predictions based on past labels, the label-oblivious version can be used even when the labels are not available until the end of the prediction process, as in test-set prediction. Interestingly, our analysis shows that both methods achieve comparable regret bounds.

After observing n examples, the online algorithm returns a (possibly randomized) classifier $\widehat{\boldsymbol{h}}_n$, which is evaluated by means of its population utility $\psi(\boldsymbol{C}(\widehat{\boldsymbol{h}}_n))$, where we recall that $\boldsymbol{C}(\widehat{\boldsymbol{h}}_n) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathbb{P},\ \widehat{\boldsymbol{y}}\sim\widehat{\boldsymbol{h}}_n(\boldsymbol{x})}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}})]$. This is compared with the *optimal (PU) classifier*:[2]

$$\boldsymbol{h}^\star \in \operatorname*{argmax}_{\boldsymbol{h}} \psi(\boldsymbol{C}(\boldsymbol{h})), \qquad \psi^\star_{\mathrm{PU}} := \psi(\boldsymbol{C}(\boldsymbol{h}^\star)) \tag{4}$$

and leads to the notion of the *PU (population) regret*:

$$R_n^{\mathrm{PU}} := \psi^\star_{\mathrm{PU}} - \mathbb{E}[\psi(\boldsymbol{C}(\widehat{\boldsymbol{h}}_n))]. \tag{5}$$

The online algorithm is equipped with the CPE $\widehat{\boldsymbol{\eta}}_t$ of the label conditionals $\boldsymbol{\eta}$, which need not be exact. All of our bounds will be phrased in terms of its *expected estimation error*, defined as:

$$\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{E}[\|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|_1] \tag{6}$$

To state our results together for the multi-class and multi-label settings, we define constants $\mathfrak{m}$ and $\mathfrak{c}$, the values of which depend on the setting: $\mathfrak{m} = \mathfrak{c} = 1$ for the multi-class, and $\mathfrak{m} = m, \mathfrak{c} = 2$ for the multi-label case.

3 Concave Utilities

We first consider differentiable concave utilities of the confusion matrix. Throughout this section, we assume that ψ is Lipschitz and smooth on $\mathcal{C}$. We use the OMMA algorithm of Kotlowski et al. [2024]: at round t, OMMA linearizes the utility at the current confusion-matrix state $\boldsymbol{C}_{t-1}$ and solves the induced cost-sensitive problem using the current CPE. The label-adaptive version updates $\boldsymbol{C}_t$ using the observed label; the label-oblivious version uses the CPE-conditional confusion contribution instead. The algorithm is presented in Algorithm 1. Kotlowski et al. [2024] prove a bound on the expected empirical utility of the online sequence relative to the optimal PU classifier:

Theorem 3.1 (OMMA bound against the PU comparator, adapted from Kotlowski et al. [2024])**.** *Assume that ψ is L-Lipschitz and M-smooth on $\mathcal{C}$. The label-adaptive OMMA algorithm satisfies*

$$\psi^\star_{\mathrm{PU}} - \mathbb{E}[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n))] \le \frac{M\mathfrak{m}(1+\ln n)}{n} + 2\sqrt{\mathfrak{c}}L\ \mathrm{Err}(\widehat{\boldsymbol{\eta}}^n).$$

[2]The existence of the optimal population classifier is shown by Schultheis et al. [2024] for a very general class of metrics; here we assume the PU classifier exists (for clarity of presentation), but it is not necessary for our results to hold.

The theorem above compares OMMA with the PU benchmark (4) (population-optimal), rather than the ETU benchmark (2) (optimal predictions for the realized inputs $\boldsymbol{x}^n$). Below, we show that, surprisingly, for concave utilities this PU comparison is in fact stronger: even though ETU predictions may adapt to the observed inputs while the PU classifier cannot, the ETU benchmark is still controlled by the PU value. Thus OMMA also enjoys the same bound on its ETU regret.

Theorem 3.2 (The ETU benchmark is controlled by the PU benchmark)**.** *Let ψ be concave. Then,*

$$\mathbb{E}[\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n)] \leq \psi^\star_{\mathrm{PU}}$$

As a consequence, label-adaptive OMMA has its ETU regret bounded as in Theorem 3.1.

The key observation is that the optimal PU value is concave as a function of the *input distribution*. In Appendix D we prove a similar bound (Theorem D.2) with additional term $\frac{M\mathfrak{m}}{2n}$ for the label-oblivious version of OMMA; Kotlowski et al. [2024] treated this case only for the exact CPE.

The previous bounds concern the expected empirical utility of the sequence produced by OMMA. They are thus online guarantees, not yet PU guarantees: after the sequence has ended, the online algorithm has produced a trajectory of cost-sensitive rules, but it has not selected a single classifier for future predictions. To obtain such a classifier in the PU setting, we use an online-to-batch conversion: choose one of the cost-sensitive rules generated along the online trajectory uniformly at random and deploy that rule on future inputs. This conversion shows that a single pass over the data is enough to obtain PU regret bounds of the same order $O(n^{-1/2})$ and with a much better dependence on m as those delivered by more involved batch reductions for complex metrics, including Frank-Wolfe and related cost-sensitive optimization methods [Narasimhan et al., 2015b, 2024, Schultheis et al., 2024].

Theorem 3.3 (Online-to-batch conversion for concave utilities)**.** *Let $\widehat{\boldsymbol{h}}_n$ draw T uniformly from $\{1, \ldots, n\}$ and use the cost-sensitive rule generated by the OMMA state $\boldsymbol{C}_{T-1}$, that is for any $\boldsymbol{x}$, $\widehat{\boldsymbol{h}}_n(\boldsymbol{x}) \in \operatorname{argmax}_{\widehat{\boldsymbol{y}}} \nabla\psi(\boldsymbol{C}_{T-1}) \cdot \mathbb{E}_{\boldsymbol{y}\sim\widehat{\boldsymbol{\eta}}_T(\boldsymbol{x})}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}})]$. Under the assumptions of Theorem 3.1,*

$$R^{\mathrm{PU}}_n \leq \frac{M\mathfrak{m}(r + \ln n)}{n} + L\sqrt{\frac{\mathfrak{m}}{n}} + 2\sqrt{\mathfrak{c}}L \operatorname{Err}(\widehat{\boldsymbol{\eta}}^n).$$

where $r = 1$ for the label-adaptive version, and $r = \frac{3}{2}$ for the label oblivious version of the algorithm.

The uniform draw in Theorem 3.3 converts control of the *average* online trajectory into a PU guarantee. It mixes decision rules, rather than averaging their parameters, and does not guarantee the same rate for the final OMMA iterate. In contrast, Theorem 4.3 directly bounds SOLO's final rule.

4 Linear-Fractional Utilities

We now turn to LF utilities, for which the analysis is motivated from the PU perspective. In this section we assume $0 \leq \boldsymbol{A} \cdot \boldsymbol{C} \leq 1$, $\gamma \leq \boldsymbol{B} \cdot \boldsymbol{C} \leq 1$ and $\psi(\boldsymbol{C}) \leq 1$ for all $\boldsymbol{C} \in \mathcal{C}$ for some $\gamma > 0$, and that every entry of $\boldsymbol{A}, \boldsymbol{B}$ is bounded by 1; all these upper bounds on $\boldsymbol{A}$ and $\boldsymbol{B}$ can always be achieved by rescaling both matrices by a constant factor which does not affect their ratio. The lower bound $\boldsymbol{B} \cdot \boldsymbol{C} \geq \gamma > 0$, while standard in the analysis of LF metrics [Kotłowski and Dembczyński, 2017, Narasimhan et al., 2024, Mao et al., 2025], need not hold globally, that is, for every confusion matrix, for commonly used utilities (Table 3). However, adding a small constant to the denominator in the utility definition stabilizes its values and guarantees the condition holds. This is also the approach we use in the experiments. Importantly, the global bound is not intrinsic to our approach: in Appendix F we give a distribution-dependent version with a local constant γ, which is satisfied by many metrics of interest, including F-measure and Jaccard. We keep the global assumption $\gamma > 0$ in the main text for clarity of presentation.

The LF utilities are generally not concave, so the analysis from the previous section no longer applies. Instead, we handle this class with the Dinkelbach transform [Dinkelbach, 1967, Schaible and Ibaraki, 1983], a classical reduction from fractional programming that has also been used in batch analyses of LF metrics [Narasimhan et al., 2015a, 2024, Zhang and Agarwal, 2024, Mao et al., 2025]. The key point is that a ratio can be optimized by finding the root of a one-dimensional value function. Specifically for LF metric (1), define the linear surrogate, parameterized by $\lambda \in [0, 1]$:

$$\psi_\lambda(\boldsymbol{C}) = N(\boldsymbol{C}) - \lambda D(\boldsymbol{C}) = (\boldsymbol{A} - \lambda\boldsymbol{B}) \cdot \boldsymbol{C}. \tag{7}$$

Algorithm 2 SOLO for LF utilities

Initialize $\lambda_0 \in [0,1]$ and set $\alpha_t = 1/(\gamma(t+1))$.
for $t = 1, \dots, n$ **do**
  Set $\boldsymbol{G}_t = \boldsymbol{A} - \lambda_{t-1}\boldsymbol{B}$.
  Observe $\boldsymbol{x}_t$ and predict $\widehat{\boldsymbol{y}}_t \in \operatorname{argmax}_{\widehat{\boldsymbol{y}}} \boldsymbol{G}_t \cdot \mathbb{E}_{\boldsymbol{y}\sim\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t)}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}})]$.
  Observe $\boldsymbol{y}_t$ and define $\boldsymbol{\Delta}_t^{\mathrm{ad}} = \boldsymbol{C}(\boldsymbol{y}_t, \widehat{\boldsymbol{y}}_t)$ and $\boldsymbol{\Delta}_t^{\mathrm{obl}} = \mathbb{E}_{\boldsymbol{y}\sim\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t)}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}}_t)]$.
  Update $\lambda_t = \mathrm{clip}_{[0,1]}\{\lambda_{t-1} + \alpha_t(\boldsymbol{G}_t \cdot \boldsymbol{\Delta}_t)\}$ with $\boldsymbol{\Delta}_t = \boldsymbol{\Delta}_t^{\mathrm{ad}}$ (adaptive) or $\boldsymbol{\Delta}_t = \boldsymbol{\Delta}_t^{\mathrm{obl}}$ (oblivious).

For any candidate $\lambda \in [0,1]$, define the optimal population value of the surrogate

$$\Phi(\lambda) := \max_{\boldsymbol{h}} \psi_\lambda(\boldsymbol{C}(\boldsymbol{h})). \tag{8}$$

Note that for any classifier $\boldsymbol{h}$ and any scalar λ, $\psi(\boldsymbol{C}(\boldsymbol{h})) = \frac{N(\boldsymbol{C}(\boldsymbol{h}))}{D(\boldsymbol{C}(\boldsymbol{h}))} \geq \lambda$ if and only if $\psi_\lambda(\boldsymbol{C}) \geq 0$. Thus the optimal value of the ratio $\lambda^\star = \max_{\boldsymbol{h}} \psi(\boldsymbol{C}(\boldsymbol{h}))$ is the threshold at which the best achievable linear surrogate $\Phi(\lambda)$ changes sign:

Proposition 4.1 (Dinkelbach root)**.** *Let $\lambda^\star := \psi(\boldsymbol{C}(\boldsymbol{h}^\star)) = \max_{\boldsymbol{h}} \psi(\boldsymbol{C}(\boldsymbol{h}))$. Then $\Phi(\lambda^\star) = 0$, $\Phi(\lambda) > 0$ for $\lambda < \lambda^\star$, and $\Phi(\lambda) < 0$ for $\lambda > \lambda^\star$. Moreover,*

$$(\lambda - \lambda^\star)\Phi(\lambda) \leq -\gamma(\lambda - \lambda^\star)^2. \tag{9}$$

Finally, any maximizer $\boldsymbol{h}^\star \in \operatorname{argmax}_{\boldsymbol{h}} \psi_{\lambda^\star}(\boldsymbol{C}(\boldsymbol{h}))$ of $\psi_{\lambda^\star}$ is also a PU classifier for ψ.

The proposition has two consequences. First, if we knew $\lambda^\star$, then optimizing the linear surrogate $\psi_{\lambda^\star}$ would recover a PU classifier for the original LF utility. Second, $\lambda^\star$ is the unique root of Φ, and the strong-sign inequality (9) says that $\Phi(\lambda)$ points toward the root: it is positive below $\lambda^\star$ and negative above it, with slope controlled by γ. This is the structure that the proposed stochastic approximation, SOLO (Algorithm 2) exploits.

We now explain how a single online example provides an estimate of $\Phi(\lambda)$ needed for stochastic approximation. For a fixed λ, let $\boldsymbol{h}_\lambda \in \operatorname{argmax}_{\boldsymbol{h}} \psi_\lambda(\boldsymbol{C}(\boldsymbol{h}))$. Since ψ_λ is linear in the confusion matrix, this classifier decomposes into pointwise cost-sensitive decisions:

$$\boldsymbol{h}_\lambda(\boldsymbol{x}) \in \operatorname*{argmax}_{\widehat{\boldsymbol{y}}} (\boldsymbol{A} - \lambda\boldsymbol{B}) \cdot \mathbb{E}_{\boldsymbol{y}\sim\boldsymbol{\eta}(\boldsymbol{x})}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}})]. \tag{10}$$

Thus, if the CPE is exact, the prediction rule of SOLO at trial t is precisely the pointwise rule (10) of $\boldsymbol{h}_{\lambda_{t-1}}$ at $\boldsymbol{x}_t$. Writing $\boldsymbol{G}_t = \boldsymbol{A} - \lambda_{t-1}\boldsymbol{B}$ and $\widehat{\Phi}_t = \boldsymbol{G}_t \cdot \boldsymbol{\Delta}_t$, we have, conditionally on the past,

$$\mathbb{E}[\widehat{\Phi}_t \mid (\boldsymbol{x}, \boldsymbol{y})^{t-1}] = \boldsymbol{G}_t \cdot \boldsymbol{C}(\boldsymbol{h}_{\lambda_{t-1}}) = \psi_{\lambda_{t-1}}(\boldsymbol{C}(\boldsymbol{h}_{\lambda_{t-1}})) = \Phi(\lambda_{t-1}).$$

Thus the update $\lambda_t = \mathrm{clip}_{[0,1]}\{\lambda_{t-1} + \alpha_t\widehat{\Phi}_t\}$ is exactly a Robbins–Monro stochastic approximation step for the equation $\Phi(\lambda) = 0$.

SOLO is close in form to OMMA, but it uses a different scalar state. If one set $\lambda_{t-1} = \psi(\boldsymbol{C}_{t-1})$, then $\nabla\psi(\boldsymbol{C}_{t-1}) = \frac{\boldsymbol{A} - \lambda_{t-1}\boldsymbol{B}}{D(\boldsymbol{C}_{t-1})}$, so the induced cost-sensitive rule would coincide with the OMMA rule. For LF utilities, SOLO instead estimates the global Dinkelbach root.

Theorem 4.2 (ETU regret for LF utilities)**.** *Run SOLO in Algorithm 2. Then, for both label-adaptive and label-oblivious updates,*

$$R_n^{\mathrm{ETU}} \leq \frac{4}{\gamma^2\sqrt{n}} + \frac{4}{\gamma}\sqrt{\frac{2\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)} + \frac{4\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n) + \frac{2\mathfrak{c}^2 m^2}{\gamma^2 n}.$$

Theorem 4.3 (PU regret for LF utilities)**.** *Return the classifier induced by the final root estimate:*

$$\widehat{\boldsymbol{h}}_n(\boldsymbol{x}) \in \operatorname*{argmax}_{\widehat{\boldsymbol{y}}} (\boldsymbol{A} - \lambda_n\boldsymbol{B}) \cdot \mathbb{E}_{\boldsymbol{y}\sim\widehat{\boldsymbol{\eta}}_{n+1}(\boldsymbol{x})}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}})].$$

Let

$$\mathrm{Err}(\widehat{\boldsymbol{\eta}}_{n+1}) = \mathbb{E}\,\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}_{n+1}(\boldsymbol{x})\|_1$$

be the one-step analogue of (6). Then

$$R_n^{\mathrm{PU}} \leq \frac{1}{\gamma}\left[\frac{1}{\gamma\sqrt{n+1}} + 2\sqrt{\frac{2\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)}\right] + \frac{4\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}_{n+1}).$$

Note that, contrary to concave utilities, here it suffices to return the final cost-sensitive rule of the online algorithm. The square-root dependence on the CPE error comes from the fact that the same approximate probabilities are used to track the Dinkelbach root. With exact CPEs, both the online ETU regret and the PU regret of the returned classifier scale as $O(n^{-1/2})$.

Theorem 4.4 (PU lower bound for returned classifier)**.** *For regularized precision $\psi(C) = \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}+1/2}$, there is $c > 0$, such that for every algorithm that observes n i.i.d. examples, knows the exact conditional probability $\eta(\boldsymbol{x})$, and returns a classifier $\widehat{\boldsymbol{h}}_n$, there exists a distribution on which*

$$R_n^{\mathrm{PU}} = \psi(\boldsymbol{C}(\boldsymbol{h}^\star)) - \mathbb{E}[\psi(\boldsymbol{C}(\widehat{\boldsymbol{h}}_n))] \geq \frac{c}{\sqrt{n}}.$$

Thus the LF population rate is unimprovable in the general case.

The lower bound applies to all learning algorithms, including randomized ones, even when given access to the exact CPE. Since the regularized precision is a differentiable, Lipschitz, and smooth LF utility, Theorem 4.4 establishes an $\Omega(n^{-1/2})$ minimax PU regret lower bound *over the class of such LF problems*, matching, up to logarithmic factors, the rate achieved by SOLO using exact CPEs. This result does not imply a corresponding lower bound for online ETU regret.

5 Empirical Evaluation

The primary contribution of this work is theoretical, aiming to formalize the ETU-to-PU connection and close known gaps in online non-decomposable metric optimization, particularly through the introduction of SOLO for LF metrics. We do not expect the proposed algorithms to yield substantial empirical gains over existing state-of-the-art heuristics, which often reach similar optimal operating points in practice but lack these theoretical foundations. We specifically focus on common LF utilities, the F-measure and the Jaccard index, to verify whether the theoretically grounded online rules match strong threshold baselines, and whether their final states retain competitive performance on the PU test set. Notably, while OMMA's formal guarantees apply strictly to concave utilities, we include it in these evaluations to demonstrate that its local linearization strategy remains a highly effective and versatile heuristic for non-concave LF metrics in practice. In this section, we include experimental results for both multi-label and multi-class settings.[3]

Multi-label experiments. We evaluate the online algorithms on four real-world multi-label datasets: YouTube and Flickr [Tang and Liu, 2009] with DeepWalk features [Perozzi et al., 2014], Mediamill [Snoek et al., 2006], and RCV1X [Lewis et al., 2004]. For each dataset, we pool the original train and test sets and split them into CPE train, online test, and PU test sets with proportions 70%/15%/15%. Detailed dataset split statistics are provided in Table 4.

First, we train a CPE on the CPE train set using a simple logistic regression model. Next, we run each online method on the online test set, where the online methods see the examples sequentially and update their state after each example. Finally, we evaluate the final state of the online method on the held-out PU test set, for which the algorithm must make independent predictions using its final state. We report macro- and micro-averaged F_1 and Jaccard. The micro metrics are linear-fractional; for macro metrics, SOLO maintains a separate root for each labelwise ratio. We run the methods with metric-specific objectives and report the results averaged over 10 random dataset splits, with each algorithm evaluated on the same splits and shuffles to ensure a fair comparison. Further details about the experimental setup and protocol are provided in Appendix G.

We compare OMMA and SOLO against a set of baselines. The default baseline predicts labels with $\hat{\eta}_j(x) > 0.5$. We also consider two threshold-tuning baselines that are not online methods: Thr. (y) tunes per-label thresholds for macro metrics and a single threshold for micro metrics on the online labels using the estimated $\hat{\boldsymbol{\eta}}$, and therefore serves as an optimistic PU-style reference. Thr. $(\hat{\eta})$ replaces labels with CPE probabilities and finds the optimal threshold on the online test set, serving as an ETU-style reference. In addition, we report results for two online baselines. OFO [Busa-Fekete et al., 2015, Jasinska et al., 2016] is a consistent online algorithm for the (micro- and macro-averaged) F-measure that produces a threshold which can be used to predict on the PU set. Greedy is an

[3] Code to replicate the experiments is available at `https://github.com/mwydmuch/MOLI`.

Table 1: Results for multi-label online and PU test sets (%), averaged over 10 runs. The best mean in each column is in **bold underlined**, the second best in ***bold italic***. Ma and Mi denote macro and micro averaging; Ja. denotes the Jaccard index. $\star$ marks a batch baseline; $\times$ denotes an unsupported metric for that baseline. Standard deviations are reported in Table 5.

Method	Online Ma-F_1	Online Mi-F_1	Online Ma-Ja.	Online Mi-Ja.	PU Ma-F_1	PU Mi-F_1	PU Ma-Ja.	PU Mi-Ja.	Online Ma-F_1	Online Mi-F_1	Online Ma-Ja.	Online Mi-Ja.	PU Ma-F_1	PU Mi-F_1	PU Ma-Ja.	PU Mi-Ja.
	YouTube								MediaMill							
$\hat{\boldsymbol{\eta}}>0.5$	22.79	30.97	15.27	18.32	22.90	30.86	15.42	18.25	4.32	53.64	3.18	36.65	4.35	53.65	3.20	36.66
OFO	36.10	43.28	×	×	36.35	***43.29***	×	×	13.29	58.01	×	×	13.77	57.83	×	×
Greedy	36.38	×	24.14	×	36.37	×	24.20	×	13.48	×	8.63	×	13.90	×	8.88	×
OMMA	36.34	**43.44**	24.12	**27.75**	***36.38***	43.28	***24.21***	***27.62***	13.48	**58.11**	8.64	**40.95**	13.87	57.84	8.86	40.69
SOLO	**36.56**	***43.42***	**24.27**	***27.73***	**36.54**	**43.29**	**24.31**	**27.63**	***14.02***	58.07	***8.92***	40.92	***14.54***	***57.84***	***9.22***	***40.69***
Thr. $(\boldsymbol{y})^\star$	×	×	×	×	36.01	43.26	23.94	27.60	×	×	×	×	**16.16**	**57.85**	**10.14**	**40.70**
Thr. $(\hat{\boldsymbol{\eta}})^\star$	***36.45***	43.42	***24.19***	27.73	×	×	×	×	**16.31**	***58.10***	**10.24**	***40.95***	×	×	×	×
	Flickr								RCV1X							
$\hat{\boldsymbol{\eta}}>0.5$	18.86	29.54	12.13	17.33	18.66	29.75	12.03	17.48	15.67	69.79	10.70	53.60	15.43	69.75	10.53	53.55
OFO	30.31	40.65	×	×	***30.68***	***41.22***	×	×	22.21	70.75	×	×	23.87	***70.72***	×	×
Greedy	30.93	×	19.96	×	30.67	×	***19.83***	×	24.86	×	16.69	×	25.03	×	***16.88***	×
OMMA	***30.94***	40.91	***19.97***	25.71	30.66	41.15	19.83	25.91	24.74	70.73	16.61	54.72	24.99	70.67	16.85	54.64
SOLO	30.90	***40.94***	19.96	***25.74***	**30.73**	**41.23**	**19.86**	**25.97**	**25.77**	**70.79**	***17.21***	**54.79**	**25.41**	**70.72**	**17.07**	**54.70**
Thr. $(\boldsymbol{y})^\star$	×	×	×	×	29.95	41.20	19.31	***25.94***	×	×	×	×	***25.10***	70.72	16.77	***54.70***
Thr. $(\hat{\boldsymbol{\eta}})^\star$	**31.14**	**40.95**	**20.15**	**25.75**	×	×	×	×	***25.66***	***70.76***	**17.27**	***54.75***	×	×	×	×

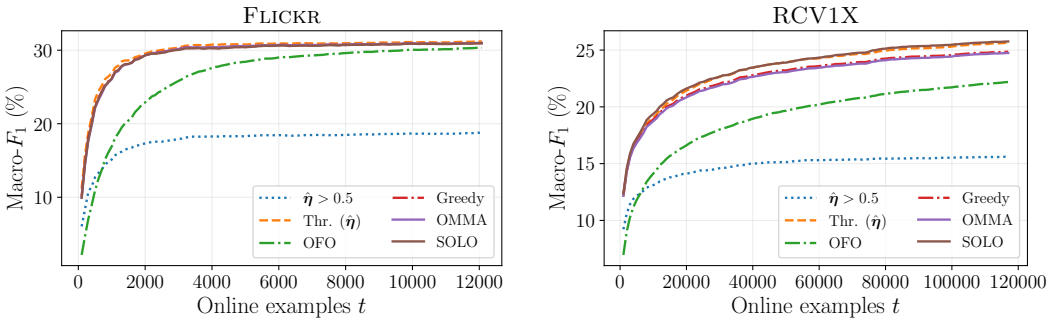


Figure 2: Incremental online macro-F_1 (%) on Flickr and RCV1X, averaged over 10 runs. The first point is at $t = 100$. Plots for all metrics and datasets are given in Figures 4 and 5.

algorithm that, given the current confusion matrix, selects the prediction maximizing the expected utility after the next step; it is easily implemented for macro metrics.

For all online algorithms that use the confusion matrix (OFO, Greedy, and OMMA), we add a small regularization constant to each entry to bound the, away from zero. We use $1/t$ in the normalized matrix, as in [Kotlowski et al., 2024]. For SOLO we use decaying step sizes $\alpha_t = 1/(t+1)$ and initial $\lambda_0 = 0.5$. Table 1 reports the results, and Figure 2 shows the incremental online performance. We study the step-size, initial-root, and stabilization choices in Appendix I.

Multi-class experiments. We also evaluate on Caltech-256 [Griffin et al., 2007] with VGG16 features [Simonyan and Zisserman, 2014] and LEDGAR from LexGLUE [Chalkidis et al., 2022] with TF-IDF features, both with logistic-regression CPEs. In single-label multi-class classification, micro-F_1 equals accuracy and micro-Jaccard is a strictly increasing function of accuracy. Both are therefore optimized by posterior Top-1 prediction and do not provide an informative test of LF root tracking. We therefore optimize the following class-weighted F_β objective, which is a single nontrivial LF ratio:

$$\text{CW-}F_\beta = \frac{(1+\beta^2)\sum_j w_j \text{tp}_j}{(1+\beta^2)\sum_j w_j \text{tp}_j + \beta^2 \sum_j w_j \text{fn}_j + \sum_j w_j \text{fp}_j}, \tag{11}$$

where $\beta = 0.25$ and $w_j \propto (n_j+1)^{-2}$, normalized to a mean of one, with n_j being the number of training examples from class j in CPE train set. These class marginals are supplied as task information before streaming, and the weights remain fixed within each split. The resulting objective generally requires cost-sensitive predictions beyond posterior Top-1.

Table 2: Results for multi-class online and PU test sets, measured by CW-$F_{0.25}$ (%) and averaged over 10 runs. The best mean in each column is in **bold underlined**, the second best in ***bold italic***. $\star$ marks a batch baseline. Standard deviations are reported in Table 6.

	CALTECH-256		LEDGAR	
Method	Online	PU	Online	PU
Top-1	78.34	80.47	77.72	79.59
Greedy	89.49	91.26	***86.53***	***87.79***
OMMA	89.49	91.26	***86.53***	**87.80**
SOLO	***91.47***	***92.00***	86.32	87.11
Batch plug-in*	**92.27**	**92.34**	**87.48**	87.63

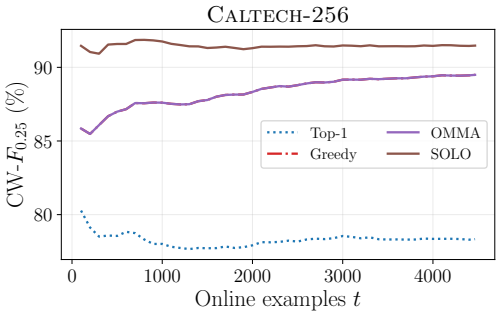


Figure 3: Incremental online CW-$F_{0.25}$ (%) on Caltech-256, averaged over 10 runs. The first point is at $t = 100$.

We use the same pooled 70%/15%/15% CPE-training/online/PU split protocol as for multi-label experiments, with 10 repetitions. We compare Top-1, Greedy, OMMA, and SOLO with a batch plug-in baseline that tunes a scalar cost parameter in the ratio-of-linear classifier family [Narasimhan et al., 2015b]. It selects the parameter maximizing empirical utility on the complete online partition and freezes it for PU prediction, analogously to multi-label threshold tuning. Its online score is an in-sample calibration reference. Table 2 and Figure 3 report the final and incremental results.

Results summary. The results show that the online OMMA and SOLO algorithms consistently recover competitive metric-specific performance for a single pass over the online set in comparison to baselines. At the same time, they present little degradation after transferring to the held-out PU set. This supports the main theoretical message of our work: the same state that yields good finite-test performance can be used as a population classifier, while the LF-specific Dinkelbach update is at least as competitive as the generic OMMA linearization on the fractional metrics tested here. OMMA and SOLO remain close to the CPE-threshold ETU, PU, and batch plug-in baselines. Additional sensitivity experiments show that SOLO is substantially less sensitive to confusion matrix regularization than the methods using the confusion matrix directly. Tuning the step-size schedule may improve final performance on the online set. However, the results show that the default schedule used in all main experiments is a robust choice.

6 Conclusion

This work establishes a unified sequential learning framework that seamlessly bridges the ETU and PU objectives for non-decomposable classification metrics. We show that for smooth concave utilities, the ETU benchmark is fundamentally controlled by the PU value, allowing local linearization methods like OMMA to achieve strong online and online-to-batch PU guarantees. To address the broader, non-concave class of linear-fractional metrics, we introduce SOLO, a stochastic approximation algorithm that tracks the Dinkelbach root online. This approach yields near-optimal ETU and PU regret bounds of $O(n^{-1/2})$ while explicitly accounting for CPE errors.

Despite these strong theoretical guarantees, several limitations remain. A primary bottleneck of both OMMA and SOLO is their explicit reliance on the accuracy of the CPE; in scenarios with highly imbalanced data, obtaining well-calibrated probabilities is notoriously difficult and directly bounds the achievable regret. Second, while SOLO theoretically justifies using the final classifier iterate for LF metrics, OMMA's theoretical PU guarantees for concave metrics still rely on a randomized online-to-batch conversion, which is often sub-optimal for practical deployment. Furthermore, the LF theoretical guarantees currently rely on a global denominator bound ($\gamma > 0$), necessitating slight regularization in practice. Finally, the theoretical picture is not yet fully closed: while our lower bound confirms the $O(n^{-1/2})$ rate is unimprovable for returned classifiers, deriving a matching unrestricted lower bound for adaptive online ETU regret remains a compelling open challenge for future research.

Acknowledgments and Disclosure of Funding

W. Kotłowski was supported from the HORIZON-MSCA-2021-DN-01 project LEMUR no. 101073307, funded by the European Union.

References

Pierre Baldi, Søren Brunak, Yves Chauvin, Claus AF Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.

Róbert Busa-Fekete, Balázs Szörényi, Krzysztof Dembczynski, and Eyke Hüllermeier. Online F-measure optimization. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 595–603, 2015.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. LexGLUE: A benchmark dataset for legal language understanding in English. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.297.

Krzysztof Dembczynski, Wojciech Kotlowski, Oluwasanmi Koyejo, and Nagarajan Natarajan. Consistency analysis for binary classification revisited. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 961–969. PMLR, 2017.

Werner Dinkelbach. On nonlinear fractional programming. *Management Science*, 13(7):492–498, 1967.

Chris Drummond and Robert C. Holte. Severe class imbalance: Why better algorithms aren't the answer. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 539–546. Springer, 2005.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Greg Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

Doheon Han, Nuno Moniz, and Nitesh V. Chawla. Anyloss: Transforming classification metrics into loss functions. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, page 992–1003. Association for Computing Machinery, 2024.

Paul Jaccard. Distribution de la flore alpine dans le bassin des dranses et dans quelques regions voisines. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 37:241–272, 1901.

Martin Jansche. A maximum expected utility framework for binary sequence labeling. In Annie Zaenen and Antal van den Bosch, editors, *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 736–743, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Kalina Jasinska, Krzysztof Dembczynski, Róbert Busa-Fekete, Karlson Pfannschmidt, Timo Klerx, and Eyke Hüllermeier. Extreme F-measure maximization using sparse probability estimates. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1435–1444. JMLR.org, 2016.

Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005), August 7-11, 2005, Bonn, Germany*, 2005.

Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Online and stochastic gradient methods for non-decomposable loss functions. *Advances in Neural Information Processing Systems*, 27, 2014.

Wojciech Kotłowski and Krzysztof Dembczyński. Surrogate regret bounds for generalized classification performance metrics. *Machine Learning*, 10:549–572, 2017.

Wojciech Kotlowski, Marek Wydmuch, Erik Schultheis, Rohit Babbar, and Krzysztof Dembczynski. A general online algorithm for optimizing complex performance metrics. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024.

Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S. Dhillon. Consistent binary classification with generalized performance metrics. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2744–2752, 2014.

Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S. Dhillon. Consistent multilabel classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3321–3329, 2015.

Harold J. Kushner and G. George Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, New York, NY, 2 edition, 2003. doi: 10.1007/b97441.

Lucien Le Cam and Grace Lo Yang. *Asymptotics in Statistics: Some Basic Concepts*. Springer Series in Statistics. Springer, New York, 2 edition, 2000. ISBN 978-0387989125. doi: 10.1007/978-1-4757-2764-4.

David D Lewis. Evaluating and optimizing autonomous text classification systems. In *Proceedings of the 18th Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 246–254. ACM, 1995.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. Multi-label learning with stronger consistency guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 2378–2406, 2024.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. Principled algorithms for optimizing generalized metrics in binary classification. In *International Conference on Machine Learning (ICML)*, 2025.

Aditya K. Menon, Harikrishna Narasimhan, Shivani Agarwal, and Sanjay Chawla. On the statistical consistency of algorithms for binary classification under class imbalance. In *International Conference on Machine Learning (ICML)*, 2013.

Harikrishna Narasimhan, Rohit Vaish, and Shivani Agarwal. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1493–1501, 2014.

Harikrishna Narasimhan, Purushottam Kar, and Prateek Jain. Optimizing non-decomposable performance measures: A tale of two classes. In *ICML*, volume 37, pages 199–208, 2015a.

Harikrishna Narasimhan, Harish Ramaswamy, Aadirupa Saha, and Shivani Agarwal. Consistent multiclass algorithms for complex performance measures. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2398–2407, Lille, France, 07 2015b. PMLR.

Harikrishna Narasimhan, Harish G. Ramaswamy, Shiv Kumar Tavker, Drona Khurana, Praneeth Netrapalli, and Shivani Agarwal. Consistent multiclass algorithms for complex metrics and constraints. *Journal of Machine Learning Research*, 25(367):1–81, 2024.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 701–710. ACM, 2014.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. doi: 10.1214/aoms/1177729586.

Siegfried Schaible and Toshihide Ibaraki. Fractional programming. *European Journal of Operational Research*, 12(4):325–338, 1983. doi: 10.1016/0377-2217(83)90127-2.

Erik Schultheis, Marek Wydmuch, Wojciech Kotlowski, Rohit Babbar, and Krzysztof Dembczynski. Generalized test utilities for long-tail performance in extreme multi-label classification. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 22269–22303. Curran Associates, Inc., 2023.

Erik Schultheis, Wojciech Kotłowski, Marek Wydmuch, Rohit Babbar, Strom Borman, and Krzysztof Dembczyński. Consistent algorithms for multi-label classification with macro-at-k metrics. In *The Twelfth International Conference on Learning Representations*, 2024.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Shashank Singh and Justin T Khim. Optimal binary classification beyond accuracy. *Advances in Neural Information Processing Systems*, 35:18226–18240, 2022.

Cees G. M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *ACM International Conference on Multimedia*, page 421–430. Association for Computing Machinery, 2006.

Ioan V. Stancu-Minasian. *Fractional Programming: Theory, Methods and Applications*, volume 409 of *Mathematics and Its Applications*. Kluwer Academic Publishers, Dordrecht, 1997. ISBN 978-0792345682.

Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, page 817–826. Association for Computing Machinery, 2009.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

Willem Waegeman, Krzysztof Dembczynski, Arkadiusz Jachnik, Weiwei Cheng, and Eyke Hüllermeier. On the bayes-optimality of F-measure maximizers. *Journal of Machine Learning Research*, 15(1):3333–3388, 2014.

Shuo Wang and Xin Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(4):1119–1130, 2012.

Nan Ye, Kian Ming Adam Chai, Wee Sun Lee, and Hai Leong Chieu. Optimizing F-measure: A tale of two approaches. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012.

Mingyuan Zhang and Shivani Agarwal. Multiclass learning from noisy labels for non-decomposable performance measures. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 238, pages 2170–2178. PMLR, 2024.

Xiaoxuan Zhang, Mingrui Liu, Xun Zhou, and Tianbao Yang. Faster online learning of optimal threshold for consistent F-measure optimization. In *Advances in Neural Information Processing Systems*, volume 31, pages 3889–3899, 2018.

Table 3: Examples of binary performance metrics. The LF column indicates whether the displayed binary metric is a linear-fractional function of the confusion matrix.

Metric	$\psi(\boldsymbol{C}^{2\times 2})$	LF?	Metric	$\psi(\boldsymbol{C}^{2\times 2})$	LF?
Accuracy	$\mathrm{tp}+\mathrm{tn}$	yes	Balanced accuracy	$\frac{1}{2}\frac{\mathrm{tp}}{\mathrm{tp+fn}}+\frac{1}{2}\frac{\mathrm{tn}}{\mathrm{tn+fp}}$	no
Recall/TPR	$\frac{\mathrm{tp}}{\mathrm{tp+fn}}$	yes	TNR	$\frac{\mathrm{tn}}{\mathrm{tn+fp}}$	yes
Precision	$\frac{\mathrm{tp}}{\mathrm{tp+fp}}$	yes	Jaccard/IoU	$\frac{\mathrm{tp}}{\mathrm{tp+fp+fn}}$	yes
F_β-measure	$\frac{(1+\beta^2)\mathrm{tp}}{(1+\beta^2)\mathrm{tp}+\beta^2\mathrm{fn}+\mathrm{fp}}$	yes	G-mean	$\sqrt{\frac{\mathrm{tptn}}{(\mathrm{tp+fn})(\mathrm{tn+fp})}}$	no
H-mean	$2\left(\frac{\mathrm{tp+fn}}{\mathrm{tp}}+\frac{\mathrm{tn+fp}}{\mathrm{tn}}\right)^{-1}$	no	Q-mean	$1-\sqrt{\frac{1}{2}\left(\left(\frac{\mathrm{fn}}{\mathrm{tp+fn}}\right)^2+\left(\frac{\mathrm{fp}}{\mathrm{tn+fp}}\right)^2\right)}$	no
Matthews coeff.	$\frac{\mathrm{tptn}-\mathrm{fpfn}}{\sqrt{(\mathrm{tp+fp})(\mathrm{tp+fn})(\mathrm{tn+fp})(\mathrm{tn+fn})}}$	no	Min	$\min\left\{\frac{\mathrm{tp}}{\mathrm{tp+fn}},\frac{\mathrm{tn}}{\mathrm{tn+fp}}\right\}$	no

A Examples of Utility Metrics

This section collects common non-decomposable performance criteria and indicates which of them fall into the LF class used in Section 4. Table 3 gives binary definitions in terms of true positives ($\mathrm{tp}=C_{11}$), false negatives ($\mathrm{fn}=C_{10}$), false positives ($\mathrm{fp}=C_{01}$), and true negatives ($\mathrm{tn}=C_{00}$).

Most confusion-matrix utilities have multi-class and multi-label extensions through micro- and macro-averaging. By rewriting the multi-class $\boldsymbol{C}^{m\times m}$ confusion matrix into a multi-label confusion tensor $\boldsymbol{C}^{m\times 2\times 2}$,

$$\boldsymbol{C}^{m\times 2\times 2}(\boldsymbol{C}^{m\times m}) := \left[\begin{pmatrix} \sum_{i\neq j,\ell\neq j} C_{i\ell} & \sum_{i\neq j} C_{ij} \\ \sum_{i\neq j} C_{ji} & C_{jj} \end{pmatrix}\right]_{j=1}^{m},$$

one can define the micro- and macro-averaged versions for both multi-class and multi-label classification as

$$\text{Micro-}\psi(\boldsymbol{C}^{m\times 2\times 2}) := \psi\left(\frac{1}{m}\sum_{j=1}^{m}\boldsymbol{C}_j\right),$$

$$\text{Macro-}\psi(\boldsymbol{C}^{m\times 2\times 2}) := \frac{1}{m}\sum_{j=1}^{m}\psi\left(\boldsymbol{C}_j\right).$$

Micro-averaged variants of LF metrics remain LF. Macro-averaged variants are generally averages of LF ratios and hence need not be LF. Concave macro objectives are covered by our concave analysis, while more general composite LF objectives require separate treatment [Schultheis et al., 2023]. In unconstrained multi-label classification, however, both the prediction space and the macro objective decompose across labels. Thus, running SOLO independently for each label, with a separate root estimate, is theoretically justified: the ETU and PU optima factorize label-wise, and averaging the single-label regret bounds yields corresponding guarantees for the macro-averaged metric. This is the approach used in our multi-label experiments.

B Notation

We denote the Euclidean (Frobenius), ℓ_1, and entrywise maximum norms of matrices/tensors by

$$\|\boldsymbol{A}\|_2=\Big(\sum_{ij}A_{ij}^2\Big)^{1/2},\quad \|\boldsymbol{A}\|_1=\sum_{ij}|A_{ij}|,\quad \|\boldsymbol{A}\|_{\max}=\max_{ij}|A_{ij}|\quad \text{(multi-class)}$$

$$\|\boldsymbol{A}\|_2=\Big(\sum_{iuv}A_{iuv}^2\Big)^{1/2},\quad \|\boldsymbol{A}\|_1=\sum_{iuv}|A_{iuv}|,\quad \|\boldsymbol{A}\|_{\max}=\max_{iuv}|A_{iuv}|\quad \text{(multi-label)}$$

For any prediction $\widehat{\boldsymbol{y}}$ and any $\boldsymbol{x}$, we use:

$$\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}),\widehat{\boldsymbol{y}})=\mathbb{E}_{\boldsymbol{y}\sim\boldsymbol{\eta}(\boldsymbol{x})}[\boldsymbol{C}(\boldsymbol{y},\widehat{\boldsymbol{y}})],\qquad \text{and similarly}\quad \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}),\widehat{\boldsymbol{y}})=\mathbb{E}_{\boldsymbol{y}\sim\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x})}[\boldsymbol{C}(\boldsymbol{y},\widehat{\boldsymbol{y}})].$$

Likewise, for any randomized classifier $\boldsymbol{h}$ and any $\boldsymbol{x}$,

$$\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \boldsymbol{h}(\boldsymbol{x})) = \mathbb{E}_{\boldsymbol{y}\sim\boldsymbol{\eta}(\boldsymbol{x}),\widehat{\boldsymbol{y}}\sim\boldsymbol{h}(\boldsymbol{x})}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}})],$$

and similarly for $\widehat{\boldsymbol{\eta}}_t$.

C Auxiliary Facts

Lemma C.1 (Empirical utility versus population utility)**.** *Let $\boldsymbol{h}$ be any fixed classifier independent of the data sequence $(\boldsymbol{x}, \boldsymbol{y})^n$, and let $\widehat{\boldsymbol{y}}_t = \boldsymbol{h}(\boldsymbol{x}_t)$. If ψ is M-smooth, then*

$$|\mathbb{E}[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n))] - \psi(\mathbb{E}[\boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n)])| \le \frac{M\mathfrak{m}}{2n}.$$

Proof. Write

$$\boldsymbol{C}_n = \frac{1}{n}\sum_{t=1}^{n} \boldsymbol{Z}_t \quad \text{with} \quad \boldsymbol{Z}_t := \boldsymbol{C}(\boldsymbol{y}_t, \boldsymbol{h}(\boldsymbol{x}_t)), \qquad \text{and} \quad \bar{\boldsymbol{C}} := \mathbb{E}[\boldsymbol{C}_n].$$

By M-smoothness, for any two confusion matrices $\boldsymbol{C}, \boldsymbol{C}'$,

$$-\frac{M}{2}\|\boldsymbol{C} - \boldsymbol{C}'\|^2 \le \psi(\boldsymbol{C}) - \psi(\boldsymbol{C}') - \nabla\psi(\boldsymbol{C}') \cdot (\boldsymbol{C} - \boldsymbol{C}') \le \frac{M}{2}\|\boldsymbol{C} - \boldsymbol{C}'\|^2.$$

Setting $\boldsymbol{C} = \boldsymbol{C}_n$ and $\boldsymbol{C}' = \bar{\boldsymbol{C}}$ in the above and then taking expectations, gives

$$|\mathbb{E}[\psi(\boldsymbol{C}_n)] - \psi(\mathbb{E}[\boldsymbol{C}_n])| \le \frac{M}{2}\,\mathbb{E}\,\|\boldsymbol{C}_n - \mathbb{E}[\boldsymbol{C}_n]\|_2^2,$$

as the first-order term disappears:

$$\mathbb{E}[\nabla\psi(\bar{\boldsymbol{C}}) \cdot (\boldsymbol{C}_n - \bar{\boldsymbol{C}})] = \nabla\psi(\bar{\boldsymbol{C}}) \cdot (\mathbb{E}[\boldsymbol{C}_n] - \bar{\boldsymbol{C}}) = 0.$$

It remains to bound the variance term. Since the examples are i.i.d. and the classifier is fixed independently of the data, the random matrices $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n$ are i.i.d. Therefore

$$\begin{aligned}\mathbb{E}\,\|\boldsymbol{C}_n - \mathbb{E}[\boldsymbol{C}_n]\|_2^2 &= \mathbb{E}\left\|\frac{1}{n}\sum_{t=1}^{n}(\boldsymbol{Z}_t - \mathbb{E}[\boldsymbol{Z}_t])\right\|_2^2 \\ &= \frac{1}{n}\,\mathbb{E}\,\|\boldsymbol{Z}_1 - \mathbb{E}[\boldsymbol{Z}_1]\|_2^2 \le \frac{1}{n}\,\mathbb{E}\,\|\boldsymbol{Z}_1\|_2^2.\end{aligned}$$

A single-example multi-class confusion matrix has squared norm 1 (as there is only one non-zero entry equal to 1); a single-example multi-label confusion tensor has at most m nonzero entries (one per label), each equal to one, and hence squared norm at most m. Thus $\mathbb{E}\,\|\boldsymbol{Z}_1\|_2^2 \le \mathfrak{m}$, which proves the claim. □

Lemma C.2 (Smoothness transfer for label-oblivious strategies)**.** *Consider a label-oblivious prediction strategy $\widehat{\boldsymbol{y}}_1, \ldots, \widehat{\boldsymbol{y}}_n$, where $\widehat{\boldsymbol{y}}_t$ is a function of the inputs $\boldsymbol{x}^n$ and auxiliary randomness ξ^n (independent of the data), but not of the labels $\boldsymbol{y}^n$. Let*

$$\boldsymbol{C}_n = \frac{1}{n}\sum_{t=1}^{n} \boldsymbol{C}(\boldsymbol{y}_t, \widehat{\boldsymbol{y}}_t)$$

denote the empirical confusion matrix and let

$$\widetilde{\boldsymbol{C}}_n := \frac{1}{n}\sum_{t=1}^{n} \mathbb{E}[\boldsymbol{C}(\boldsymbol{y}_t, \widehat{\boldsymbol{y}}_t) \mid \boldsymbol{x}^n, \xi^n]$$

denote the average of conditional means. If ψ is M-smooth, then

$$\left|\mathbb{E}[\psi(\boldsymbol{C}_n)] - \mathbb{E}[\psi(\widetilde{\boldsymbol{C}}_n)]\right| \le \frac{M\mathfrak{m}}{2n}.$$

Proof. Define the realized and conditional single-round contributions

$$\boldsymbol{Z}_t := \boldsymbol{C}(\boldsymbol{y}_t, \widehat{\boldsymbol{y}}_t), \qquad \boldsymbol{U}_t := \mathbb{E}[\boldsymbol{Z}_t \mid \boldsymbol{x}^n, \xi^n], \qquad \boldsymbol{D}_t := \boldsymbol{Z}_t - \boldsymbol{U}_t.$$

Then

$$\boldsymbol{C}_n - \widetilde{\boldsymbol{C}}_n = \frac{1}{n}\sum_{t=1}^{n} \boldsymbol{D}_t, \qquad \mathbb{E}[\boldsymbol{D}_t \mid \boldsymbol{x}^n, \xi^n] = 0.$$

Similarly as in the proof of Lemma C.1, the M-smoothness of ψ gives

$$-\frac{M}{2}\|\boldsymbol{C}_n - \widetilde{\boldsymbol{C}}_n\|_2^2 \le \psi(\boldsymbol{C}_n) - \psi(\widetilde{\boldsymbol{C}}_n) - \nabla\psi(\widetilde{\boldsymbol{C}}_n)\cdot(\boldsymbol{C}_n - \widetilde{\boldsymbol{C}}_n) \le \frac{M}{2}\|\boldsymbol{C}_n - \widetilde{\boldsymbol{C}}_n\|_2^2.$$

Taking expectations, the first-order term vanishes:

$$\begin{aligned}\mathbb{E}[\nabla\psi(\widetilde{\boldsymbol{C}}_n)\cdot(\boldsymbol{C}_n - \widetilde{\boldsymbol{C}}_n)] &= \frac{1}{n}\sum_{t=1}^{n}\mathbb{E}[\nabla\psi(\widetilde{\boldsymbol{C}}_n)\cdot\boldsymbol{D}_t] \\ &= \frac{1}{n}\sum_{t=1}^{n}\mathbb{E}\Big[\nabla\psi(\widetilde{\boldsymbol{C}}_n)\cdot\mathbb{E}[\boldsymbol{D}_t \mid \boldsymbol{x}^n, \xi^n]\Big] = 0.\end{aligned}$$

Therefore

$$\Big|\mathbb{E}[\psi(\boldsymbol{C}_n)] - \mathbb{E}[\psi(\widetilde{\boldsymbol{C}}_n)]\Big| \le \frac{M}{2}\,\mathbb{E}\,\|\boldsymbol{C}_n - \widetilde{\boldsymbol{C}}_n\|_2^2.$$

It remains to compute the second moment. Expanding the square,

$$\mathbb{E}\,\|\boldsymbol{C}_n - \widetilde{\boldsymbol{C}}_n\|_2^2 = \frac{1}{n^2}\sum_{t=1}^{n}\mathbb{E}\,\|\boldsymbol{D}_t\|_2^2 + \frac{2}{n^2}\sum_{1\le s<t\le n}\mathbb{E}[\boldsymbol{D}_s\cdot\boldsymbol{D}_t].$$

We now use the assumption that the classifier is label-oblivious, which implies that for $s \neq t$, $\boldsymbol{D}_s$ and $\boldsymbol{D}_t$ are independent given $(\boldsymbol{x}^n, \xi^n)$, and each has conditional mean zero by construction. The cross terms thus vanish:

$$\mathbb{E}[\boldsymbol{D}_s\cdot\boldsymbol{D}_t] = \mathbb{E}[\mathbb{E}[\boldsymbol{D}_s\cdot\boldsymbol{D}_t \mid \boldsymbol{x}^n, \xi^n]] = 0,$$

For the diagonal terms,

$$\mathbb{E}\,\|\boldsymbol{D}_t\|_2^2 = \mathbb{E}\,\|\boldsymbol{Z}_t - \mathbb{E}[\boldsymbol{Z}_t \mid \boldsymbol{x}^n, \xi^n]\|_2^2 \le \mathbb{E}\,\|\boldsymbol{Z}_t\|_2^2 \le \mathfrak{m},$$

where we used the same bound on a single-example confusion contribution as in Lemma C.1. Therefore $\mathbb{E}\,\|\boldsymbol{C}_n - \widetilde{\boldsymbol{C}}_n\|_2^2 \le \mathfrak{m}/n$, which proves the result. □

Lemma C.3 (Lipschitz transfer for adaptive strategies)**.** *Consider a predictable sequence of classifiers $\boldsymbol{h}_t : \mathcal{X} \to \Delta(\mathcal{Y})$, where $\boldsymbol{h}_t$ is a function of $(\boldsymbol{x}, \boldsymbol{y})^{t-1}$ and auxiliary randomness ξ^t (independent of the data), but not of the fresh pair $(\boldsymbol{x}_t, \boldsymbol{y}_t)$. Let $\widehat{\boldsymbol{y}}_t \sim \boldsymbol{h}_t(\boldsymbol{x}_t)$ and let*

$$\boldsymbol{C}_n = \frac{1}{n}\sum_{t=1}^{n}\boldsymbol{C}(\boldsymbol{y}_t, \widehat{\boldsymbol{y}}_t)$$

denote the empirical confusion matrix and let

$$\widetilde{\boldsymbol{C}}_n := \frac{1}{n}\sum_{t=1}^{n}\boldsymbol{C}(\boldsymbol{h}_t)$$

denote the average population confusion matrix of these classifiers. If ψ is L-Lipschitz, then

$$\Big|\mathbb{E}[\psi(\boldsymbol{C}_n)] - \mathbb{E}[\psi(\widetilde{\boldsymbol{C}}_n)]\Big| \le L\sqrt{\frac{\mathfrak{m}}{n}}.$$

Proof. Define the realized and conditional single-round contributions

$$\boldsymbol{Z}_t := \boldsymbol{C}(\boldsymbol{y}_t, \widehat{\boldsymbol{y}}_t), \qquad \boldsymbol{U}_t := \boldsymbol{C}(\boldsymbol{h}_t), \qquad \boldsymbol{D}_t := \boldsymbol{Z}_t - \boldsymbol{U}_t.$$

Then

$$\boldsymbol{C}_n - \widetilde{\boldsymbol{C}}_n = \frac{1}{n}\sum_{t=1}^{n}\boldsymbol{D}_t, \qquad \mathbb{E}[\boldsymbol{D}_t \mid (\boldsymbol{x}, \boldsymbol{y})^{t-1}, \xi^t] = 0.$$

Moreover,

$$\mathbb{E}\,\|\boldsymbol{C}_n - \widetilde{\boldsymbol{C}}_n\|_2^2 = \mathbb{E}\left\|\frac{1}{n}\sum_{t=1}^{n}\boldsymbol{D}_t\right\|_2^2 = \frac{1}{n^2}\sum_{t=1}^{n}\mathbb{E}\,\|\boldsymbol{D}_t\|_2^2 + \frac{2}{n^2}\sum_{1\le s<t\le n}\mathbb{E}[\boldsymbol{D}_s\cdot\boldsymbol{D}_t].$$

For $s < t$, $\boldsymbol{D}_s$ is measurable with respect to $((\boldsymbol{x},\boldsymbol{y})^{t-1},\xi^t)$, while

$$\mathbb{E}[\boldsymbol{D}_t \mid (\boldsymbol{x},\boldsymbol{y})^{t-1},\xi^t] = 0.$$

Hence

$$\mathbb{E}[\boldsymbol{D}_s\cdot\boldsymbol{D}_t] = \mathbb{E}\big[\boldsymbol{D}_s\cdot\mathbb{E}[\boldsymbol{D}_t \mid (\boldsymbol{x},\boldsymbol{y})^{t-1},\xi^t]\big] = 0.$$

For the diagonal terms,

$$\mathbb{E}\,\|\boldsymbol{D}_t\|_2^2 = \mathbb{E}\,\|\boldsymbol{Z}_t - \boldsymbol{C}(\boldsymbol{h}_t)\|_2^2 \le \mathbb{E}\,\|\boldsymbol{Z}_t\|_2^2 \le \mathfrak{m},$$

where the last inequality uses the same bound on a single-example confusion contribution as in Lemma C.1, so that

$$\mathbb{E}\,\|\boldsymbol{C}_n - \widetilde{\boldsymbol{C}}_n\|_2^2 \le \frac{\mathfrak{m}}{n}.$$

Finally, using Lipschitzness of ψ and Jensen's inequality,

$$\Big(\mathbb{E}[\psi(\boldsymbol{C}_n)] - \mathbb{E}[\psi(\widetilde{\boldsymbol{C}}_n)]\Big)^2 \le \mathbb{E}\left[(\psi(\boldsymbol{C}_n) - \psi(\widetilde{\boldsymbol{C}}_n))^2\right] \le L^2\,\mathbb{E}\left[\|\boldsymbol{C}_n - \widetilde{\boldsymbol{C}}_n\|_2^2\right] \le \frac{L^2\mathfrak{m}}{n},$$

and the statement follows by taking square root on both sides. □

Lemma C.4 (Perturbation bound)**.** *For any input $\boldsymbol{x}$ and prediction $\widehat{\boldsymbol{y}}$:*

$$\|\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}),\widehat{\boldsymbol{y}}) - \boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),\widehat{\boldsymbol{y}})\|_2 \le \sqrt{\mathfrak{c}}\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_2,$$
$$\|\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}),\widehat{\boldsymbol{y}}) - \boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),\widehat{\boldsymbol{y}})\|_1 \le \mathfrak{c}\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_1,$$

Proof. Denote:

$$\boldsymbol{\Delta} = \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}),\widehat{\boldsymbol{y}}) - \boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}),\widehat{\boldsymbol{y}})$$

For multi-class classification, $\boldsymbol{\Delta}$ is a matrix with entries equal to $\boldsymbol{\Delta}_{ij} = (\eta_i(\boldsymbol{x}) - \widehat{\eta}_i(\boldsymbol{x}))\widehat{y}_j$. Due to one-hot constraint $\sum_j \widehat{y}_j = 1$, we get that for any $\widehat{\boldsymbol{y}}$, $\|\boldsymbol{\Delta}\|_2^2 = \sum_{i=1}^m |\eta_i(\boldsymbol{x}) - \widehat{\eta}_i(\boldsymbol{x})|^2 = \|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_2^2$, or equivalently, $\|\boldsymbol{\Delta}\|_2 = \|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_2$. Similarly, $\|\boldsymbol{\Delta}\|_1 = \|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_1$. For multi-label classification $\boldsymbol{\Delta}$ is a sequence of m matrices of size 2×2, with i-th matrix having entries

$$\begin{pmatrix} (\widehat{\eta}_i(\boldsymbol{x}) - \eta_i(\boldsymbol{x}))(1-\widehat{y}_i) & (\widehat{\eta}_i(\boldsymbol{x}) - \eta_i(\boldsymbol{x}))\widehat{y}_i \\ (\eta_i(\boldsymbol{x}) - \widehat{\eta}_i(\boldsymbol{x}))(1-\widehat{y}_i) & (\eta_i(\boldsymbol{x}) - \widehat{\eta}_i(\boldsymbol{x}))\widehat{y}_i \end{pmatrix},$$

so that $\|\boldsymbol{\Delta}\|_2^2 = 2\sum_{i=1}^m |\eta_i(\boldsymbol{x}) - \widehat{\eta}_i(\boldsymbol{x})|^2 = 2\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_2^2$, or equivalently, $\|\boldsymbol{\Delta}\|_2 = \sqrt{2}\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_2$; and likewise $\|\boldsymbol{\Delta}\|_1 = 2\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}(\boldsymbol{x})\|_1$ □

D Concave Utilities: Proofs

We first compare the ETU and PU benchmarks (Theorem 3.2), then prove the online guarantee for label-oblivious OMMA (Theorem D.2), and finally give the online-to-batch conversion (Theorem 3.3).

D.1 Comparing the ETU and PU benchmarks

The key observation is that the optimal PU value is *concave as a function of the input distribution*, with the conditional label distribution held fixed.

For any input distribution ν over $\mathcal{X}$ and any randomized classifier $\boldsymbol{h}$, let

$$\boldsymbol{C}_\nu(\boldsymbol{h}) := \mathbb{E}_{\boldsymbol{x}\sim\nu}[\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}),\boldsymbol{h}(\boldsymbol{x}))] \tag{12}$$

be the population confusion matrix of a possibly randomized classifier $\boldsymbol{h}$, and define

$$F_\psi(\nu) := \sup_{\boldsymbol{h}} \psi(\boldsymbol{C}_\nu(\boldsymbol{h})). \tag{13}$$

In particular, when $\nu = \mathbb{P}_\mathcal{X}$, $\boldsymbol{C}_\nu(\boldsymbol{h})$ is the usual population confusion matrix $\boldsymbol{C}(\boldsymbol{h})$, while $F_\psi(\nu)$ is the utility achieved by the PU classifier.

Proposition D.1 (Concavity of the value functional)**.** *If ψ is concave, then the map $\nu \mapsto F_\psi(\nu)$ defined in (13) is concave in the input distribution ν.*

Proof. Fix two input distributions ν_1, ν_2, a mixing coefficient $\alpha \in (0,1)$, and write their convex combination as $\nu := \alpha\nu_1 + (1-\alpha)\nu_2$. Let $\boldsymbol{h}_1, \boldsymbol{h}_2$ be two possibly randomized classifiers, and let

$$\boldsymbol{C}_{\nu_i}(\boldsymbol{h}_i) = \mathbb{E}_{\boldsymbol{x}\sim\nu_i}[\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \boldsymbol{h}_i(\boldsymbol{x}))] \qquad (i = 1, 2),$$

be the confusion matrix of classifier $\boldsymbol{h}_i$ over distribution ν_i.

Under the mixture law ν, define a randomized classifier $\bar{\boldsymbol{h}}$ as follows: at input $\boldsymbol{x}$, choose the prediction of $\boldsymbol{h}_1$ with probability $p_1(\boldsymbol{x}) = \alpha \frac{\mathrm{d}\nu_1}{\mathrm{d}\nu}(\boldsymbol{x})$ and that of $\boldsymbol{h}_2$ with probability $p_2(\boldsymbol{x}) = (1-\alpha)\frac{\mathrm{d}\nu_2}{\mathrm{d}\nu}(\boldsymbol{x})$ (since $\alpha\nu_1 \ll \nu$ and $(1-\alpha)\nu_2 \ll \nu$, the Radon-Nikodym derivatives exist, and these probabilities are well-defined and satisfy $p_1(\boldsymbol{x}) + p_2(\boldsymbol{x}) = 1$ ν-almost surely).

The resulting confusion matrix of $\bar{\boldsymbol{h}}$ under ν is

$$\begin{aligned}
\boldsymbol{C}_\nu(\bar{\boldsymbol{h}}) &= \mathbb{E}_{\boldsymbol{x}\sim\nu}[\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \bar{\boldsymbol{h}}(\boldsymbol{x}))] = \mathbb{E}_{\boldsymbol{x}\sim\nu}\left[p_1(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \boldsymbol{h}_1(\boldsymbol{x})) + p_2(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \boldsymbol{h}_2(\boldsymbol{x}))\right] \\
&= \int \alpha \frac{\mathrm{d}\nu_1}{\mathrm{d}\nu}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \boldsymbol{h}_1(\boldsymbol{x}))\mathrm{d}\nu(\boldsymbol{x}) + \int (1-\alpha)\frac{\mathrm{d}\nu_2}{\mathrm{d}\nu}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \boldsymbol{h}_2(\boldsymbol{x}))\mathrm{d}\nu(\boldsymbol{x}) \\
&= \alpha\,\mathbb{E}_{\boldsymbol{x}\sim\nu_1}[\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \boldsymbol{h}_1(\boldsymbol{x}))] + (1-\alpha)\,\mathbb{E}_{\boldsymbol{x}\sim\nu_2}[\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \boldsymbol{h}_2(\boldsymbol{x}))] \\
&= \alpha\boldsymbol{C}_{\nu_1}(\boldsymbol{h}_1) + (1-\alpha)\boldsymbol{C}_{\nu_2}(\boldsymbol{h}_2).
\end{aligned}$$

Therefore

$$\begin{aligned}
F_\psi(\nu) = \sup_{\boldsymbol{h}} \psi(\boldsymbol{C}_\nu(\boldsymbol{h})) &\geq \psi(\boldsymbol{C}_\nu(\bar{\boldsymbol{h}})) = \psi\left(\alpha\boldsymbol{C}_{\nu_1}(\boldsymbol{h}_1) + (1-\alpha)\boldsymbol{C}_{\nu_2}(\boldsymbol{h}_2)\right) \\
&\geq \alpha\psi(\boldsymbol{C}_{\nu_1}(\boldsymbol{h}_1)) + (1-\alpha)\psi(\boldsymbol{C}_{\nu_2}(\boldsymbol{h}_2)),
\end{aligned}$$

where the last inequality follows from the concavity of ψ. Since $\boldsymbol{h}_1, \boldsymbol{h}_2$ were chosen arbitrarily, the inequality holds for any $\boldsymbol{h}_1, \boldsymbol{h}_2$, and thus

$$\begin{aligned}
F_\psi(\nu) &\geq \sup_{\boldsymbol{h}_1, \boldsymbol{h}_2} \{\alpha\psi(\boldsymbol{C}_{\nu_1}(\boldsymbol{h}_1)) + (1-\alpha)\psi(\boldsymbol{C}_{\nu_2}(\boldsymbol{h}_2))\} \\
&= \alpha \sup_{\boldsymbol{h}_1} \{\psi(\boldsymbol{C}_{\nu_1}(\boldsymbol{h}_1))\} + (1-\alpha) \sup_{\boldsymbol{h}_2} \{\psi(\boldsymbol{C}_{\nu_2}(\boldsymbol{h}_2))\} \\
&= \alpha F_\psi(\nu_1) + (1-\alpha)F_\psi(\nu_2),
\end{aligned}$$

which was to be shown. □

Theorem 3.2 (The ETU benchmark is controlled by the PU benchmark)**.** *Let ψ be concave. Then,*

$$\mathbb{E}[\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n)] \leq \psi^\star_{\mathrm{PU}}$$

As a consequence, label-adaptive OMMA has its ETU regret bounded as in Theorem 3.1.

Proof. Given inputs $\boldsymbol{x}^n$, define the ETU optimal predictions (2):

$$\boldsymbol{y}^{\star n} = \operatorname*{argmax}_{\widehat{\boldsymbol{y}}^n} \mathbb{E}_{\boldsymbol{y}^n|\boldsymbol{x}^n}\left[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n))\right],$$

where we keep the dependence of $\boldsymbol{y}^{\star n} = \boldsymbol{y}^{\star n}(\boldsymbol{x}^n)$ implicit for the clarity of presentation. By Jensen's inequality,

$$\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n) = \mathbb{E}_{\boldsymbol{y}^n|\boldsymbol{x}^n}\left[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{y}^{\star n}))\right] \leq \psi\left(\mathbb{E}_{\boldsymbol{y}^n|\boldsymbol{x}^n}[\boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{y}^{\star n})]\right) \tag{14}$$

Note that the predictions $\boldsymbol{y}^{\star n}$ might not correspond to any classifier $\boldsymbol{h}\colon \mathcal{X} \to \mathcal{Y}$, as if the same input appears more than once, e.g. $\boldsymbol{x}_i = \boldsymbol{x}_j$ for some $i \neq j$, we could still have $\boldsymbol{y}^\star_i \neq \boldsymbol{y}^\star_j$, while we would need to have $\boldsymbol{h}(\boldsymbol{x}_i) = \boldsymbol{h}(\boldsymbol{x}_j)$ for any classifier $\boldsymbol{h}$. However, we can show that the confusion matrix $\mathbb{E}_{\boldsymbol{y}^n|\boldsymbol{x}^n}[\boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{y}^{\star n})]$ in (14) can be realized by some *randomized* classifier. To this end, for any input $\boldsymbol{x}$, let $\mathcal{I}_{\boldsymbol{x}} = \{t : \boldsymbol{x}_t = \boldsymbol{x}\}$ (if $\boldsymbol{x}$ does not appear in the sequence $\boldsymbol{x}^n$, then $\mathcal{I}_{\boldsymbol{x}} = \emptyset$). Define a randomized classifier $\boldsymbol{h}_{\boldsymbol{x}^n}$ by choosing at any input $\boldsymbol{x}$ one of the predictions $\{\boldsymbol{y}^\star_t : t \in \mathcal{I}_{\boldsymbol{x}}\}$ uniformly at random; if $\mathcal{I}_{\boldsymbol{x}} = \emptyset$, set $\boldsymbol{h}_{\boldsymbol{x}^n}(\boldsymbol{x})$ arbitrarily. Then

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{y}^n|\boldsymbol{x}^n}[\boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{y}^{\star n})] &= \frac{1}{n}\sum_{t=1}^n \mathbb{E}_{\boldsymbol{y}\sim\boldsymbol{\eta}(\boldsymbol{x}_t)}[\boldsymbol{C}(\boldsymbol{y}, \boldsymbol{y}^\star_t)] \\
&= \frac{1}{n}\sum_{t=1}^n \mathbb{E}_{\boldsymbol{y}\sim\boldsymbol{\eta}(\boldsymbol{x}_t)}\,\mathbb{E}_{\widehat{\boldsymbol{y}}_t\sim\boldsymbol{h}_{\boldsymbol{x}^n}(\boldsymbol{x}_t)}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}}_t)] \\
&= \boldsymbol{C}_{\nu_n}(\boldsymbol{h}_{\boldsymbol{x}^n}),
\end{aligned}$$

where $\nu_n = \frac{1}{n}\sum_{t=1}^{n} \delta_{\boldsymbol{x}_t}$ is the empirical input distribution over sequence $\boldsymbol{x}^n$ and we used definition (12). By (14),

$$\mathbb{E}_{\boldsymbol{y}^n|\boldsymbol{x}^n}\left[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{y}^{\star n}(\boldsymbol{x}^n)))\right] \le \psi\left(\boldsymbol{C}_{\nu_n}(\boldsymbol{h}_{\boldsymbol{x}^n})\right) \le \sup_{\boldsymbol{h}} \psi\left(\boldsymbol{C}_{\nu_n}(\boldsymbol{h})\right) = F_\psi(\nu_n).$$

Taking expectation over $\boldsymbol{x}^n$ on both sides and using concavity of F_ψ (Proposition D.1),

$$\mathbb{E}\left[\underbrace{\psi(\boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{y}^{\star n}(\boldsymbol{x}^n)))}_{\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n)}\right] \le F_\psi(\mathbb{E}[\nu_n]) = F_\psi(\mathbb{P}_{\mathcal{X}}) = \underbrace{\sup_{\boldsymbol{h}} \psi(\boldsymbol{C}(\boldsymbol{h}))}_{\psi^\star_{\mathrm{PU}}}.$$

□

D.2 Online regret for label-oblivious OMMA

In what follows, we assume that the label-obliviousness applies to the entire prediction strategy, including the CPE, that is $\widehat{\boldsymbol{\eta}}_t$ does not depend on $\boldsymbol{y}^{t-1}$.

Theorem D.2 (Label-oblivious OMMA with approximate CPE)**.** *Assume that ψ is concave, L-Lipschitz, and M-smooth on $\mathcal{C}$. The label-oblivious version of OMMA satisfies*

$$\psi^\star_{\mathrm{PU}} - \mathbb{E}[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n))] \le \frac{M\mathfrak{m}(1+\ln n)}{n} + \frac{M\mathfrak{m}}{2n} + 2\sqrt{\mathfrak{c}}L\,\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n).$$

Consequently, its ETU regret is bounded by the same right-hand side.

Proof. Let $\boldsymbol{h}^\star$ be the optimal PU classifier, that is $\psi^\star_{\mathrm{PU}} = \psi(\boldsymbol{C}(\boldsymbol{h}^\star))$. Let $\widehat{\boldsymbol{C}}_t$ be the internal state of label-oblivious OMMA, which can be written as:

$$\widehat{\boldsymbol{C}}_t = \frac{1}{t}\sum_{s=1}^{t} \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_s(\boldsymbol{x}_s), \widehat{\boldsymbol{y}}_s).$$

Also denote

$$\boldsymbol{G}_{t-1} = \nabla\psi(\widehat{\boldsymbol{C}}_{t-1}), \qquad \text{and} \quad \boldsymbol{C}^\star = \boldsymbol{C}(\boldsymbol{h}^\star).$$

We first control the internal trajectory of the algorithm. By concavity,

$$\psi(\boldsymbol{C}^\star) - \psi(\widehat{\boldsymbol{C}}_{t-1}) \le \boldsymbol{G}_{t-1} \cdot (\boldsymbol{C}^\star - \widehat{\boldsymbol{C}}_{t-1}). \tag{15}$$

By M-smoothness and the update:

$$\widehat{\boldsymbol{C}}_t = \frac{t-1}{t}\widehat{\boldsymbol{C}}_{t-1} + \frac{1}{t}\boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t) = \widehat{\boldsymbol{C}}_{t-1} + \frac{1}{t}(\boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t) - \widehat{\boldsymbol{C}}_{t-1}),$$

we get

$$\psi(\widehat{\boldsymbol{C}}_t) \ge \psi(\widehat{\boldsymbol{C}}_{t-1}) + \frac{1}{t}\boldsymbol{G}_{t-1} \cdot (\boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t) - \widehat{\boldsymbol{C}}_{t-1}) - \frac{M}{2t^2}\|\boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t) - \widehat{\boldsymbol{C}}_{t-1}\|_2^2.$$

Both $\boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t)$ and $\widehat{\boldsymbol{C}}_{t-1}$ belong to the convex hull of single-example confusion contributions; hence $\|\boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t) - \widehat{\boldsymbol{C}}_{t-1}\|_2^2 \le 2\mathfrak{m}$, by the same arguments as in, e.g., the proof of Lemma C.1. Multiplying the last inequality by t and using (15) gives the recursion

$$t\big(\psi(\boldsymbol{C}^\star) - \psi(\widehat{\boldsymbol{C}}_t)\big) \le (t-1)\big(\psi(\boldsymbol{C}^\star) - \psi(\widehat{\boldsymbol{C}}_{t-1})\big) + \boldsymbol{G}_{t-1} \cdot (\boldsymbol{C}^\star - \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t)) + \frac{M\mathfrak{m}}{t}. \tag{16}$$

We now bound the linear term. First, condition on all past observations $(\boldsymbol{x}, \boldsymbol{y})^{t-1}$ that determine $\widehat{\boldsymbol{C}}_{t-1}$, $\boldsymbol{G}_{t-1}$, and the CPE $\widehat{\boldsymbol{\eta}}_t$. The classification rule of the algorithm at trial t selects

$$\widehat{\boldsymbol{y}}_t \in \operatorname*{argmax}_{\widehat{\boldsymbol{y}}} \boldsymbol{G}_{t-1} \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}).$$

Using Lemma C.4

$$\|\boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}) - \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}})\|_2 \le \sqrt{\mathfrak{c}}\|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_2,$$

and $\|\boldsymbol{G}_{t-1}\|_2 \le L$, which follows from the Lipschitzness of ψ, we have for any $\widehat{\boldsymbol{y}}$,

$$\begin{aligned}
\boldsymbol{G}_{t-1} \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}) &\le \boldsymbol{G}_{t-1} \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}) + \|\boldsymbol{G}_{t-1}\|_2 \|\boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}) - \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}})\|_2 \\
&\le \boldsymbol{G}_{t-1} \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}) + \sqrt{\mathfrak{c}} L \|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_2 \\
&\le \max_{\widehat{\boldsymbol{y}}} \boldsymbol{G}_{t-1} \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}) + \sqrt{\mathfrak{c}} L \|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_2 \\
&= \boldsymbol{G}_{t-1} \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t) + \sqrt{\mathfrak{c}} L \|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_2,
\end{aligned}$$

where the last equality follows from the definition of $\widehat{\boldsymbol{y}}_t$. In particular, the above applies to the prediction of the PU classifier at $\boldsymbol{x}_t$, $\widehat{\boldsymbol{y}} = \boldsymbol{h}^\star(\boldsymbol{x}_t)$, which after taking expectation over the fresh input $\boldsymbol{x}_t$ gives

$$\boldsymbol{G}_{t-1} \cdot \boldsymbol{C}^\star \le \mathbb{E}_{\boldsymbol{x}_t}[\boldsymbol{G}_{t-1} \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t)] + \sqrt{\mathfrak{c}} L \, \mathbb{E}_{\boldsymbol{x}_t} \|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_2.$$

Thus, taking expectation of (16) and plugging the above yields

$$t\big(\psi(\boldsymbol{C}^\star) - \mathbb{E}[\psi(\widehat{\boldsymbol{C}}_t)]\big) \le (t-1)\big(\psi(\boldsymbol{C}^\star) - \mathbb{E}[\psi(\widehat{\boldsymbol{C}}_{t-1})]\big) + \sqrt{\mathfrak{c}} L \, \mathbb{E}_{\boldsymbol{x}_t} \|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_2 + \frac{M\mathfrak{m}}{t}.$$

Summing over $t = 1, \dots, n$, using $\sum_{t=1}^n t^{-1} \le 1 + \ln n$, and the fact that $\|\cdot\|_2 \le \|\cdot\|_1$ gives

$$\psi(\boldsymbol{C}^\star) - \mathbb{E}[\psi(\widehat{\boldsymbol{C}}_n)] \le \frac{M\mathfrak{m}(1 + \ln n)}{n} + \sqrt{\mathfrak{c}} L \, \mathrm{Err}(\widehat{\boldsymbol{\eta}}^n). \tag{17}$$

Define

$$\widetilde{\boldsymbol{C}}_n = \frac{1}{n} \sum_{t=1}^n \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t).$$

By Lipschitzness,

$$\begin{aligned}
|\psi(\widehat{\boldsymbol{C}}_n) - \psi(\widetilde{\boldsymbol{C}}_n)| &\le L \|\widehat{\boldsymbol{C}}_n - \widetilde{\boldsymbol{C}}_n\|_2 \\
&= L \left\| \frac{1}{n} \sum_{t=1}^n (\boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t) - \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t)) \right\|_2 \\
&\le \frac{L}{n} \sum_{t=1}^n \|\boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t) - \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t)\|_2 \\
&\le \frac{\sqrt{\mathfrak{c}} L}{n} \sum_{t=1}^n \|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_2 \\
&\le \frac{\sqrt{\mathfrak{c}} L}{n} \sum_{t=1}^n \|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_1,
\end{aligned}$$

so that

$$\mathbb{E}\left[|\psi(\widehat{\boldsymbol{C}}_n) - \psi(\widetilde{\boldsymbol{C}}_n)|\right] \le \sqrt{\mathfrak{c}} L \mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)$$

Plugging it to (17) gives:

$$\psi(\boldsymbol{C}^\star) - \mathbb{E}[\psi(\widetilde{\boldsymbol{C}}_n)] \le \frac{M\mathfrak{m}(1 + \ln n)}{n} + 2\sqrt{\mathfrak{c}} L \, \mathrm{Err}(\widehat{\boldsymbol{\eta}}^n).$$

The final step is to relate $\mathbb{E}[\psi(\widetilde{\boldsymbol{C}}_n)]$ to $\mathbb{E}[\psi(\boldsymbol{C}_n)]$, the expected utility of the empirical confusion matrix of the algorithm $\boldsymbol{C}_n = \boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n)$, but this follows directly from Lemma C.2: $\mathbb{E}[\psi(\widetilde{\boldsymbol{C}}_n)] - \mathbb{E}[\psi(\boldsymbol{C}_n)] \le \frac{M\mathfrak{m}}{2n}$, thus finishing the proof. The ETU-regret claim follows from the benchmark comparison in Theorem 3.2. □

D.3 Online-to-batch conversion

Theorem 3.3 (Online-to-batch conversion for concave utilities)**.** *Let $\widehat{\boldsymbol{h}}_n$ draw T uniformly from $\{1, \dots, n\}$ and use the cost-sensitive rule generated by the OMMA state $\boldsymbol{C}_{T-1}$, that is for any $\boldsymbol{x}$, $\widehat{\boldsymbol{h}}_n(\boldsymbol{x}) \in \operatorname{argmax}_{\widehat{\boldsymbol{y}}} \nabla\psi(\boldsymbol{C}_{T-1}) \cdot \mathbb{E}_{\boldsymbol{y} \sim \widehat{\boldsymbol{\eta}}_T(\boldsymbol{x})}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}})]$. Under the assumptions of Theorem 3.1,*

$$R_n^{\mathrm{PU}} \le \frac{M\mathfrak{m}(r + \ln n)}{n} + L\sqrt{\frac{\mathfrak{m}}{n}} + 2\sqrt{\mathfrak{c}} L \, \mathrm{Err}(\widehat{\boldsymbol{\eta}}^n).$$

where $r = 1$ for the label-adaptive version, and $r = \frac{3}{2}$ for the label oblivious version of the algorithm.

Proof. We prove the theorem assuming the prediction algorithm depends on the past labels, to cover both label-adaptive and label-oblivious cases (the label-oblivious algorithm simply does not use the available labels). For each trial $t = 1, \dots, n$, let $\boldsymbol{h}_t$ denote the cost-sensitive rule generated by the corresponding OMMA state before seeing the fresh pair $(\boldsymbol{x}_t, \boldsymbol{y}_t)$, that is

$$\boldsymbol{h}_t(\boldsymbol{x}) \in \operatorname*{argmax}_{\widehat{\boldsymbol{y}}} \nabla\psi(\boldsymbol{C}_{t-1}) \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}), \widehat{\boldsymbol{y}}).$$

The online-to-batch classifier $\widehat{\boldsymbol{h}}_n$ first draws $T \sim \mathrm{Unif}\{1, \dots, n\}$, independently of all data and auxiliary randomness, and then uses $\boldsymbol{h}_T$. Conditional on the online trajectory, its population confusion matrix is therefore

$$\boldsymbol{C}(\widehat{\boldsymbol{h}}_n) = \frac{1}{n} \sum_{t=1}^{n} \boldsymbol{C}(\boldsymbol{h}_t).$$

Since $\boldsymbol{h}_t$ is determined before the fresh pair $(\boldsymbol{x}_t, \boldsymbol{y}_t)$ is drawn, Lemma C.3 applies and gives

$$\left| \mathbb{E}[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n))] - \mathbb{E}[\psi(\boldsymbol{C}(\widehat{\boldsymbol{h}}_n))] \right| \le L\sqrt{\frac{\mathfrak{m}}{n}}.$$

Combining this with the online guarantee in Theorem 3.1 (label-adaptive) or Theorem D.2 (label-oblivious) yields

$$\psi(\boldsymbol{C}(\boldsymbol{h}^\star)) - \mathbb{E}[\psi(\boldsymbol{C}(\widehat{\boldsymbol{h}}_n))] \le \frac{M\mathfrak{m}(r + \ln n)}{n} + 2\sqrt{\mathfrak{c}} L \operatorname{Err}(\widehat{\boldsymbol{\eta}}^n) + L\sqrt{\frac{\mathfrak{m}}{n}},$$

where $r = 1$ in the label-adaptive case and $r = 3/2$ in the label-oblivious case. This is the claimed PU regret bound. □

E Linear-Fractional Utilities: Proofs

Throughout this section we use the normalization and global denominator bound from Section 4. We first establish Lipschitzness and smoothness (Lemma E.1), characterize the population optimum $\lambda^\star$ as a Dinkelbach root (Proposition 4.1), and bound the effect of CPE error (Lemma E.2). We then control the online estimates λ_t (Lemma E.3) and compare the ETU and PU benchmarks (Lemma E.4). We then establish the ETU and PU guarantees (Theorems 4.2 and 4.3), and finally prove the population lower bound (Theorem 4.4).

E.1 Regularity of the utility under global denominator bound assumption

Lemma E.1 (Lipschitzness and smoothness of linear-fractional ψ)**.** *The LF utility (1) is L-Lipschitz and M-smooth on $\mathcal{C}$ with*

$$L \le \frac{2m}{\gamma},\ M \le \frac{4m^2}{\gamma^2} \quad \textit{(multi-class)}, \qquad L \le \frac{4\sqrt{m}}{\gamma},\ M \le \frac{16m}{\gamma^2} \quad \textit{(multi-label)}.$$

Proof. We identify matrices/tensors with their vectorizations when writing gradients, Hessians, and Euclidean norms. First,

$$\nabla\psi(\boldsymbol{C}) = \frac{\boldsymbol{A}D(\boldsymbol{C}) - \boldsymbol{B}N(\boldsymbol{C})}{D(\boldsymbol{C})^2} = \frac{\boldsymbol{A} - \boldsymbol{B}\frac{N(\boldsymbol{C})}{D(\boldsymbol{C})}}{D(\boldsymbol{C})} = \frac{\boldsymbol{A} - \psi(\boldsymbol{C})\boldsymbol{B}}{D(\boldsymbol{C})}.$$

Since $\psi(\boldsymbol{C}) \in [0, 1]$ and $D(\boldsymbol{C}) \ge \gamma$,

$$\|\nabla\psi(\boldsymbol{C})\|_2 \le \frac{\|\boldsymbol{A}\|_2 + \|\boldsymbol{B}\|_2}{\gamma}.$$

We now compute the Hessian exactly. Differentiating the gradient gives

$$\nabla^2\psi(\boldsymbol{C}) = -\frac{(\boldsymbol{A} - \psi(\boldsymbol{C})\boldsymbol{B})\boldsymbol{B}^\top + \boldsymbol{B}(\boldsymbol{A} - \psi(\boldsymbol{C})\boldsymbol{B})^\top}{D(\boldsymbol{C})^2}.$$

For two vectors $\boldsymbol{u}, \boldsymbol{v}$, the nonzero eigenvalues of $\boldsymbol{u}\boldsymbol{v}^\top + \boldsymbol{v}\boldsymbol{u}^\top$ are $\boldsymbol{u} \cdot \boldsymbol{v} \pm \|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2$; hence

$$\|\boldsymbol{u}\boldsymbol{v}^\top + \boldsymbol{v}\boldsymbol{u}^\top\|_{\mathrm{op}} = \|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2 + |\boldsymbol{u} \cdot \boldsymbol{v}| \le 2\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2.$$

Applying this identity with $u = \boldsymbol{A} - \psi(\boldsymbol{C})\boldsymbol{B}$ and $\boldsymbol{v} = \boldsymbol{B}$ yields

$$\|\nabla^2\psi(\boldsymbol{C})\|_{\mathrm{op}} \le \frac{2\|\boldsymbol{B}\|_2\|\boldsymbol{A} - \psi(\boldsymbol{C})\boldsymbol{B}\|_2}{D(\boldsymbol{C})^2} \le \frac{2\|\boldsymbol{B}\|_2(\|\boldsymbol{A}\|_2 + \|\boldsymbol{B}\|_2)}{D(\boldsymbol{C})^2}$$

Using the entrywise normalization $\|\boldsymbol{A}\|_{\max}, \|\boldsymbol{B}\|_{\max} \le 1$, we have $\|\boldsymbol{A}\|_2, \|\boldsymbol{B}\|_2 \le m$ for multi-class classification, and $\|\boldsymbol{A}\|_2, \|\boldsymbol{B}\|_2 \le 2\sqrt{m}$ for multi-label classification. This gives the claimed bounds. □

E.2 Characterization of the population optimum (Dinkelbach root)

Recall the definition of the linear surrogate (7),

$$\psi_\lambda(\boldsymbol{C}) = N(\boldsymbol{C}) - \lambda D(\boldsymbol{C}),$$

and its optimal population value (8),

$$\Phi(\lambda) = \max_{\boldsymbol{h}} \psi_\lambda(\boldsymbol{C}(\boldsymbol{h})).$$

Proposition 4.1 (Dinkelbach root)**.** *Let* $\lambda^\star := \psi(\boldsymbol{C}(\boldsymbol{h}^\star)) = \max_{\boldsymbol{h}} \psi(\boldsymbol{C}(\boldsymbol{h}))$. *Then* $\Phi(\lambda^\star) = 0$, $\Phi(\lambda) > 0$ *for* $\lambda < \lambda^\star$, *and* $\Phi(\lambda) < 0$ *for* $\lambda > \lambda^\star$. *Moreover,*

$$(\lambda - \lambda^\star)\Phi(\lambda) \le -\gamma(\lambda - \lambda^\star)^2. \tag{9}$$

Finally, any maximizer $\boldsymbol{h}^\star \in \operatorname{argmax}_{\boldsymbol{h}} \psi_{\lambda^\star}(\boldsymbol{C}(\boldsymbol{h}))$ *of* $\psi_{\lambda^\star}$ *is also a PU classifier for* ψ.

Proof. For every classifier $\boldsymbol{h}$, and every $\lambda \in [0, 1]$,

$$\begin{aligned}\psi_\lambda(\boldsymbol{C}(\boldsymbol{h})) &= N(\boldsymbol{C}(\boldsymbol{h})) - \lambda D(\boldsymbol{C}(\boldsymbol{h})) = \frac{N(\boldsymbol{C}(\boldsymbol{h}))}{D(\boldsymbol{C}(\boldsymbol{h}))} D(\boldsymbol{C}(\boldsymbol{h})) - \lambda D(\boldsymbol{C}(\boldsymbol{h})) \\ &= D(\boldsymbol{C}(\boldsymbol{h}))(\psi(\boldsymbol{C}(\boldsymbol{h})) - \lambda).\end{aligned}$$

Let $\boldsymbol{h}^\star \in \operatorname{argmax}_{\boldsymbol{h}} \psi(\boldsymbol{C}(\boldsymbol{h}))$ be the PU classifier, so that $\psi(\boldsymbol{C}(\boldsymbol{h}^\star)) = \lambda^\star$. Evaluating the last expression at $\lambda = \lambda^\star$ gives

$$\psi_{\lambda^\star}(\boldsymbol{C}(\boldsymbol{h}^\star)) = D(\boldsymbol{C}(\boldsymbol{h}^\star))(\lambda^\star - \lambda^\star) = 0.$$

On the other hand, by the definition of $\lambda^\star$, $\psi(\boldsymbol{C}(\boldsymbol{h})) \le \lambda^\star$ for every $\boldsymbol{h}$, and hence

$$\psi_{\lambda^\star}(\boldsymbol{C}(\boldsymbol{h})) = N(\boldsymbol{C}(\boldsymbol{h})) - \lambda^\star D(\boldsymbol{C}(\boldsymbol{h})) \le 0$$

for every $\boldsymbol{h}$. Taking the maximum over $\boldsymbol{h}$ proves $\Phi(\lambda^\star) = 0$.

If $\lambda < \lambda^\star$, evaluating the surrogate at $\boldsymbol{h}^\star$ gives

$$\Phi(\lambda) \ge \psi_\lambda(\boldsymbol{C}(\boldsymbol{h}^\star)) = D(\boldsymbol{C}(\boldsymbol{h}^\star))(\lambda^\star - \lambda) > 0.$$

If $\lambda > \lambda^\star$, then for every $\boldsymbol{h}$,

$$N(\boldsymbol{C}(\boldsymbol{h})) - \lambda D(\boldsymbol{C}(\boldsymbol{h})) = D(\boldsymbol{C}(\boldsymbol{h}))(\psi(\boldsymbol{C}(\boldsymbol{h})) - \lambda) \le D(\boldsymbol{C}(\boldsymbol{h}))(\lambda^\star - \lambda) < 0,$$

so $\Phi(\lambda) < 0$.

We next prove the strong-sign inequality. Consider first $\lambda \ge \lambda^\star$. Since $\psi(\boldsymbol{C}(\boldsymbol{h})) - \lambda \le \lambda^\star - \lambda \le 0$ and $D(\boldsymbol{C}(\boldsymbol{h})) \ge \gamma$, we have

$$\psi_\lambda(\boldsymbol{C}(\boldsymbol{h})) = D(\boldsymbol{C}(\boldsymbol{h}))(\psi(\boldsymbol{C}(\boldsymbol{h})) - \lambda) \le D(\boldsymbol{C}(\boldsymbol{h}))(\lambda^\star - \lambda) \le \gamma(\lambda^\star - \lambda)$$

(the last inequality is due to $\lambda^\star - \lambda \le 0$). Taking the maximum over $\boldsymbol{h}$,

$$\Phi(\lambda) \le \gamma(\lambda^\star - \lambda).$$

Multiplying by $\lambda - \lambda^\star \ge 0$ yields

$$(\lambda - \lambda^\star)\Phi(\lambda) \le -\gamma(\lambda - \lambda^\star)^2.$$

Now consider $\lambda \le \lambda^\star$. Evaluating at $\boldsymbol{h}^\star$ gives

$$\Phi(\lambda) \ge \psi_\lambda(\boldsymbol{C}(\boldsymbol{h}^\star)) = D(\boldsymbol{C}(\boldsymbol{h}^\star))(\lambda^\star - \lambda) \ge \gamma(\lambda^\star - \lambda).$$

Since $\lambda - \lambda^\star \le 0$, multiplying this inequality by $\lambda - \lambda^\star$ reverses the sign and gives the same strong-sign bound. This proves (9).

Finally, let $\boldsymbol{h}^\star$ be any maximizer of $\psi_{\lambda^\star}(\boldsymbol{C}(\boldsymbol{h}))$. Because $\Phi(\lambda^\star) = 0$, this maximizer satisfies

$$N(\boldsymbol{C}(\boldsymbol{h}^\star)) - \lambda^\star D(\boldsymbol{C}(\boldsymbol{h}^\star)) = 0.$$

Dividing by $D(\boldsymbol{C}(\boldsymbol{h}^\star)) > 0$ gives $\psi(\boldsymbol{C}(\boldsymbol{h}^\star)) = \lambda^\star$. Hence $\boldsymbol{h}^\star$ is an optimal PU classifier for the original LF utility. □

E.3 CPE perturbation bound

Lemma E.2 (Pointwise surrogate and CPE perturbation)**.** *For any* $\lambda \in [0,1]$, *let* $\boldsymbol{G}_\lambda := \boldsymbol{A} - \lambda \boldsymbol{B}$. *Then*

$$\Phi(\lambda) = \mathbb{E}_{\boldsymbol{x}} \left[\max_{\widehat{\boldsymbol{y}}} \{\boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}})\} \right]. \tag{18}$$

Moreover, if $\widehat{\boldsymbol{y}}_{\widehat{\eta}}(\boldsymbol{x}) \in \operatorname{argmax}_{\widehat{\boldsymbol{y}}} \boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}), \widehat{\boldsymbol{y}})$ *then, pointwise in* $\boldsymbol{x}$,

$$\max_{\widehat{\boldsymbol{y}}} \{\boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}})\} - \{\boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}}_{\widehat{\eta}})\} \leq 4\mathfrak{c} \|\widehat{\boldsymbol{\eta}}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|_1, \tag{19}$$

and

$$\left| \max_{\widehat{\boldsymbol{y}}} \{\boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}})\} - \max_{\widehat{\boldsymbol{y}}} \{\boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}), \widehat{\boldsymbol{y}})\} \right| \leq 2\mathfrak{c} \|\widehat{\boldsymbol{\eta}}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|_1, \tag{20}$$

Proof. The surrogate ψ_λ is linear in the confusion matrix. Therefore, for any classifier $\boldsymbol{h}$,

$$\begin{aligned} \psi_\lambda(\boldsymbol{C}(\boldsymbol{h})) &= \boldsymbol{G}_\lambda \cdot \mathbb{E}_{\boldsymbol{x}} \, \mathbb{E}_{\boldsymbol{y} \sim \boldsymbol{\eta}(\boldsymbol{x})} [\boldsymbol{C}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x}))] \\ &= \mathbb{E}_{\boldsymbol{x}} \left[\boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \boldsymbol{h}(\boldsymbol{x})) \right]. \end{aligned}$$

The maximization over $\boldsymbol{h}$ therefore decomposes pointwise in $\boldsymbol{x}$ into maximization over every $\boldsymbol{h}(\boldsymbol{x})$, which proves (18).

For the perturbation bound, note that from Lemma C.4, for every $\boldsymbol{x}$ and $\widehat{\boldsymbol{y}}$,

$$\|\boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}), \widehat{\boldsymbol{y}}) - \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}})\|_1 \leq \mathfrak{c} \|\widehat{\boldsymbol{\eta}}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|_1.$$

Since $\lambda \in [0,1]$, we have $\|\boldsymbol{G}_\lambda\|_{\max} \leq \|\boldsymbol{A}\|_{\max} + \|\boldsymbol{B}\|_{\max} \leq 2$, and hence by Hölder's inequality $|\boldsymbol{X} \cdot \boldsymbol{Y}| \leq \|\boldsymbol{X}\|_{\max} \|\boldsymbol{Y}\|_1$,

$$|\boldsymbol{G}_\lambda \cdot (\boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}), \widehat{\boldsymbol{y}}) - \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}}))| \leq 2\mathfrak{c} \|\widehat{\boldsymbol{\eta}}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|_1. \tag{21}$$

Let $\widehat{\boldsymbol{y}}_\eta \in \operatorname{argmax}_{\widehat{\boldsymbol{y}}} \boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}})$ maximize the objective with respect to the true label conditionals $\boldsymbol{\eta}$, and let $\widehat{\boldsymbol{y}}_{\widehat{\eta}} \in \operatorname{argmax}_{\widehat{\boldsymbol{y}}} \boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}), \widehat{\boldsymbol{y}})$ maximize the objective with respect to the CPE. Using (21) twice for $\widehat{\boldsymbol{y}}_\eta$ and $\widehat{\boldsymbol{y}}_{\widehat{\eta}}$,

$$\begin{aligned} &\boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}}_\eta) - \boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}}_{\widehat{\eta}}) \\ &\leq \boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}), \widehat{\boldsymbol{y}}_\eta) - \boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}), \widehat{\boldsymbol{y}}_{\widehat{\eta}}) + 4\mathfrak{c} \|\widehat{\boldsymbol{\eta}}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|_1 \\ &\leq 4\mathfrak{c} \|\widehat{\boldsymbol{\eta}}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|_1, \end{aligned}$$

where the last step uses the optimality of $\widehat{\boldsymbol{y}}_{\widehat{\eta}}$ under $\widehat{\boldsymbol{\eta}}(\boldsymbol{x})$, thus proving (19).

Finally, for (20),

$$\begin{aligned} &\boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}}_\eta) - \boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}), \widehat{\boldsymbol{y}}_{\widehat{\eta}}) \\ &\leq \boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}), \widehat{\boldsymbol{y}}_\eta) - \boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}), \widehat{\boldsymbol{y}}_{\widehat{\eta}}) + 2\mathfrak{c} \|\widehat{\boldsymbol{\eta}}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|_1 \\ &\leq 2\mathfrak{c} \|\widehat{\boldsymbol{\eta}}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|_1, \end{aligned}$$

where we also used the optimality of $\widehat{\boldsymbol{y}}_{\widehat{\eta}}$ in the last step. Similar argument by replacing $\boldsymbol{\eta}$ and $\widehat{\boldsymbol{\eta}}$ leads to

$$\boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}(\boldsymbol{x}), \widehat{\boldsymbol{y}}_{\widehat{\eta}}) - \boldsymbol{G}_\lambda \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}}_\eta) \leq 2\mathfrak{c} \|\widehat{\boldsymbol{\eta}}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|_1,$$

which together prove (20).

□

E.4 Online root tracking

Lemma E.3 (One-dimensional root tracking with approximate CPE)**.** *For both the label-adaptive and label-oblivious versions of SOLO,*

$$\mathbb{E}[(\lambda_t - \lambda^\star)^2] \leq \frac{1}{\gamma^2 (t+1)} + \frac{8\mathfrak{c}}{\gamma} \frac{1}{t} \sum_{s=1}^{t} \mathbb{E}[\|\widehat{\boldsymbol{\eta}}_s(\boldsymbol{x}_s) - \boldsymbol{\eta}(\boldsymbol{x}_s)\|_1].$$

Consequently,

$$\frac{1}{n} \sum_{t=1}^{n} \mathbb{E}\left[|\lambda_{t-1} - \lambda^\star|\right] \leq \frac{2}{\gamma \sqrt{n}} + 4 \sqrt{\frac{2\mathfrak{c}}{\gamma} \operatorname{Err}(\widehat{\boldsymbol{\eta}}^n)}.$$

Proof. Write the scalar update in Algorithm 2 as

$$\lambda_t = \mathrm{clip}_{[0,1]}\left\{\lambda_{t-1} + \frac{1}{\gamma(t+1)}\widehat{\Phi}_t\right\},$$

where

$$\widehat{\Phi}_t := (\boldsymbol{A} - \lambda_{t-1}\boldsymbol{B}) \cdot \boldsymbol{\Delta}_t.$$

Here $\boldsymbol{\Delta}_t$ is either the realized contribution $\boldsymbol{\Delta}_t^{\mathrm{ad}} = \boldsymbol{C}(\boldsymbol{y}_t, \widehat{\boldsymbol{y}}_t)$ in the label-adaptive update, or the estimated conditional contribution $\boldsymbol{\Delta}_t^{\mathrm{obl}} = \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t)$ in the label-oblivious update. Since $\lambda_{t-1} \in [0,1]$ and $|N(\boldsymbol{C}) - \lambda D(\boldsymbol{C})| \le 1$, we have $|\widehat{\Phi}_t| \le 1$.

Let us now condition on the past examples $(\boldsymbol{x}, \boldsymbol{y})^{t-1}$ that determine λ_{t-1} and $\widehat{\boldsymbol{\eta}}_t$, and denote by $\mathbb{E}_{(\boldsymbol{x}_t, \boldsymbol{y}_t)}[\cdot]$, $\mathbb{E}_{\boldsymbol{x}_t}[\cdot]$ and $\mathbb{E}_{\boldsymbol{y}_t|\boldsymbol{x}_t}[\cdot]$ expectations conditional on $(\boldsymbol{x}, \boldsymbol{y})^{t-1}$. Denoting $\boldsymbol{G}_{\lambda_{t-1}} = \boldsymbol{A} - \lambda_{t-1}\boldsymbol{B}$, define

$$\widetilde{\Phi}_t = \max_{\widehat{\boldsymbol{y}}} \left\{\boldsymbol{G}_{\lambda_{t-1}} \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}})\right\},$$

which would be the objective achieved by the algorithm if the CPE were exact ($\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) = \boldsymbol{\eta}(\boldsymbol{x}_t)$). By Lemma E.2, $\mathbb{E}_{\boldsymbol{x}_t}[\widetilde{\Phi}_t] = \Phi(\lambda_{t-1})$. The approximate CPE introduces bias, which, however, can be controlled. For the label-oblivious version, we have

$$\widehat{\Phi}_t = \max_{\widehat{\boldsymbol{y}}} \left\{\boldsymbol{G}_{\lambda_{t-1}} \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}})\right\},$$

and by Lemma E.2,

$$|\widehat{\Phi}_t - \widetilde{\Phi}_t| \le 2\mathfrak{c}\|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_1.$$

For the label-adaptive version, we have

$$\mathbb{E}_{\boldsymbol{y}_t|\boldsymbol{x}_t}\left[\widehat{\Phi}_t\right] = \boldsymbol{G}_{\lambda_{t-1}} \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_{\widehat{\boldsymbol{\eta}}_t}),$$

where $\widehat{\boldsymbol{y}}_{\widehat{\boldsymbol{\eta}}_t} = \mathrm{argmax}_{\widehat{\boldsymbol{y}}}\, \boldsymbol{G}_{\lambda_{t-1}} \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t), \widehat{\boldsymbol{y}})$, and similarly by Lemma E.2,

$$|\,\mathbb{E}_{\boldsymbol{y}_t|\boldsymbol{x}_t}\,\widehat{\Phi}_t - \widetilde{\Phi}_t| \le 4\mathfrak{c}\|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_1$$

Thus, in both cases we bound the bias introduced by the CPE:

$$\begin{aligned}
|\,\mathbb{E}_{(\boldsymbol{x}_t, \boldsymbol{y}_t)}[\widehat{\Phi}_t] - \Phi(\lambda_{t-1})| &= |\,\mathbb{E}_{(\boldsymbol{x}_t, \boldsymbol{y}_t)}[\widehat{\Phi}_t - \widetilde{\Phi}_t]| \\
&\le \mathbb{E}_{\boldsymbol{x}_t}\left[|\,\mathbb{E}_{\boldsymbol{y}_t|\boldsymbol{x}_t}[\widehat{\Phi}_t] - \widetilde{\Phi}_t|\right] \\
&\le 4\mathfrak{c}\,\mathbb{E}_{\boldsymbol{x}_t}[\|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_1].
\end{aligned}$$

We now proceed with bounding the mean squared error of λ_t. Projection onto $[0,1]$ is non-expansive and $\lambda^\star \in [0,1]$, hence

$$\begin{aligned}
(\lambda_t - \lambda^\star)^2 &\le \left(\lambda_{t-1} - \lambda^\star + \frac{\widehat{\Phi}_t}{\gamma(t+1)}\right)^2 \\
&= (\lambda_{t-1} - \lambda^\star)^2 + \frac{2(\lambda_{t-1} - \lambda^\star)\widehat{\Phi}_t}{\gamma(t+1)} + \frac{\widehat{\Phi}_t^2}{\gamma^2(t+1)^2}.
\end{aligned}$$

Taking expectation conditional on the past, using the strong-sign inequality (9), the bias bound above, $|\lambda_{t-1} - \lambda^\star| \le 1$, and $|\widehat{\Phi}_t| \le 1$, we get

$$\begin{aligned}
\mathbb{E}_{(\boldsymbol{x}_t, \boldsymbol{y}_t)}[(\lambda_t - \lambda^\star)^2] \le &\left(1 - \frac{2}{t+1}\right)(\lambda_{t-1} - \lambda^\star)^2 \\
&+ \frac{8\mathfrak{c}}{\gamma(t+1)}\,\mathbb{E}_{\boldsymbol{x}_t}\left[\|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_1\right] \\
&+ \frac{1}{\gamma^2(t+1)^2}.
\end{aligned}$$

Let $q_t := \mathbb{E}[(\lambda_t - \lambda^\star)^2]$ and $e_t = \mathbb{E}\,\|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_1$. Taking now full expectations gives

$$q_t \le \left(1 - \frac{2}{t+1}\right) q_{t-1} + \frac{8\mathfrak{c}}{\gamma(t+1)} e_t + \frac{1}{\gamma^2(t+1)^2}.$$

A direct induction yields

$$q_t \leq \frac{1}{\gamma^2(t+1)} + \frac{8\mathfrak{c}}{\gamma}\frac{1}{t}\sum_{s=1}^{t} e_s.$$

Indeed, the bound is immediate for $t = 1$, as:

$$q_1 \leq \frac{8\mathfrak{c}}{2\gamma}e_1 + \frac{1}{4\gamma^2} \leq \frac{1}{2\gamma^2} + \frac{8\mathfrak{c}}{\gamma}e_1.$$

If it holds at $t-1$, substitution into the recursion gives

$$\begin{aligned} q_t &\leq \frac{t-1}{t+1}\left\{\frac{1}{\gamma^2 t} + \frac{8\mathfrak{c}}{\gamma}\frac{1}{t-1}\sum_{s=1}^{t-1} e_s\right\} + \frac{8\mathfrak{c}}{\gamma(t+1)}e_t + \frac{1}{\gamma^2(t+1)^2} \\ &= \frac{1}{\gamma^2(t+1)}\left(\frac{t-1}{t} + \frac{1}{t+1}\right) + \frac{8\mathfrak{c}}{\gamma}\frac{1}{t+1}\sum_{s=1}^{t} e_s \\ &\leq \frac{1}{\gamma^2(t+1)} + \frac{8\mathfrak{c}}{\gamma}\frac{1}{t}\sum_{s=1}^{t} e_s. \end{aligned}$$

We now finally average the absolute errors of λ_t. Jensen's inequality gives for any $t > 1$,

$$\mathbb{E}\left[|\lambda_{t-1} - \lambda^\star|\right] \leq \sqrt{\mathbb{E}\left[(\lambda_{t-1} - \lambda^\star)^2\right]} \leq \frac{1}{\gamma\sqrt{t}} + 2\sqrt{\frac{2\mathfrak{c}}{\gamma}\frac{1}{t-1}\sum_{s=1}^{t-1} e_s}.$$

For $t = 1$, we use the bound $\mathbb{E}\left[|\lambda_0 - \lambda^\star|\right] \leq 1$, as both $\lambda_0, \lambda^\star \in [0,1]$. We now average over $t = 1, \ldots, n$. Using $n^{-1}\sum_{t=1}^{n} t^{-1/2} \leq 2n^{-1/2}$, the first term on the right-hand side is bounded by:

$$\frac{1}{n}\sum_{t=2}^{n}\frac{1}{\gamma\sqrt{t}} = \frac{1}{\gamma}\left(\frac{1}{n}\sum_{t=1}^{n}\frac{1}{\sqrt{t}} - \frac{1}{n}\right) \leq \frac{2}{\gamma\sqrt{n}} - \frac{1}{\gamma n}$$

For the second term, we use

$$\sum_{t=2}^{n}\sqrt{\frac{1}{t-1}\sum_{s=1}^{t-1} e_s} \leq \sqrt{\sum_{s=1}^{n-1} e_s}\sum_{t=2}^{n}\frac{1}{\sqrt{t-1}} \leq 2\sqrt{(n-1)\sum_{s=1}^{n-1} e_s}$$

Consequently,

$$\begin{aligned} \frac{1}{n}\sum_{t=1}^{n}\mathbb{E}\left[|\lambda_{t-1} - \lambda^\star|\right] &\leq \frac{1}{n}\left(1 - \frac{1}{\gamma}\right) + \frac{2}{\gamma\sqrt{n}} + 4\sqrt{\frac{2\mathfrak{c}}{\gamma}\frac{1}{n}\sum_{t=1}^{n-1} e_t} \\ &\leq \frac{2}{\gamma\sqrt{n}} + 4\sqrt{\frac{2\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)}, \end{aligned}$$

where we used $\gamma \leq 1$.

□

E.5 Comparing the ETU and PU benchmarks

Unlike for concave utilities, the expected ETU benchmark for LF utilities need not be bounded above by the optimal PU value: adapting predictions to the realized input sample can yield a higher expected utility. The following lemma shows, however, that this possible advantage vanishes at rate $O(n^{-1/2})$. The comparison proceeds through the *empirical Dinkelbach root* $\widehat{\lambda}_n$, which represents the optimal LF utility evaluated at the conditional mean confusion matrix for the observed inputs. Smoothness controls the difference between the ETU benchmark and this empirical root, while concentration controls the difference between the empirical root and the population optimum $\lambda^\star$.

Lemma E.4 (LF ETU–PU benchmark gap)**.** *Let $\lambda^\star = \psi(\boldsymbol{C}(\boldsymbol{h}^\star)) = \psi^\star_{\mathrm{PU}}$ be the optimal PU value and let $\boldsymbol{y}^{\star n} = \boldsymbol{y}^{\star n}(\boldsymbol{x}^n)$ be the optimal ETU prediction sequence. Then*

$$\underbrace{\mathbb{E}[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{y}^{\star n}))]}_{\mathbb{E}[\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n)]} - \underbrace{\psi(\boldsymbol{C}(\boldsymbol{h}^\star))}_{\psi^\star_{\mathrm{PU}}} \le \frac{1}{\gamma\sqrt{n}} + \frac{2\mathfrak{c}^2 m^2}{\gamma^2 n}.$$

Proof. For a realized input sequence $\boldsymbol{x}^n$, define the empirical Dinkelbach map

$$\widehat{\Phi}_n(\lambda) := \frac{1}{n}\sum_{t=1}^{n} \max_{\widehat{\boldsymbol{y}}} \{(\boldsymbol{A} - \lambda \boldsymbol{B}) \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}})\},$$

and let $\widehat{\lambda}_n$ be its root. We first compare the ETU benchmark with this root. Fix $\boldsymbol{x}^n$. For any deterministic prediction sequence $\boldsymbol{z}^n$, define its true-CPE conditional mean confusion matrix

$$\bar{\boldsymbol{C}}(\boldsymbol{z}^n; \boldsymbol{x}^n) := \frac{1}{n}\sum_{t=1}^{n} \mathbb{E}_{\boldsymbol{y}\sim\boldsymbol{\eta}(\boldsymbol{x}_t)}[\boldsymbol{C}(\boldsymbol{y}, \boldsymbol{z}_t)].$$

The finite-sample Dinkelbach argument applied to the empirical input distribution gives

$$\psi(\bar{\boldsymbol{C}}(\boldsymbol{z}^n; \boldsymbol{x}^n)) \le \widehat{\lambda}_n \qquad \text{for all } \boldsymbol{z}^n.$$

Indeed, if the ratio of some $\bar{\boldsymbol{C}}(\boldsymbol{z}^n; \boldsymbol{x}^n)$ were strictly larger than $\widehat{\lambda}_n$, then the empirical surrogate at $\widehat{\lambda}_n$ evaluated on that prediction sequence would be positive, contradicting $\widehat{\Phi}_n(\widehat{\lambda}_n) = 0$.

For fixed $\boldsymbol{x}^n$ and $\boldsymbol{z}^n$, the predictions are label-oblivious. Hence Lemma C.2, applied conditionally on $\boldsymbol{x}^n$, gives

$$\mathbb{E}_{\boldsymbol{y}^n|\boldsymbol{x}^n}[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{z}^n))] \le \psi(\bar{\boldsymbol{C}}(\boldsymbol{z}^n; \boldsymbol{x}^n)) + \frac{M\mathfrak{m}}{2n} \le \widehat{\lambda}_n + \frac{M\mathfrak{m}}{2n},$$

where M is the smoothness constant of ψ; by Lemma E.1, $M \le 4m^2/\gamma^2$ for multi-class case, which gives $\frac{M\mathfrak{m}}{2n} \le \frac{2m^2}{\gamma^2 n}$, and $M \le 16m/\gamma^2$ for multi-label case, which gives $\frac{M\mathfrak{m}}{2n} \le \frac{8m^2}{\gamma^2 n}$, which can be jointly rewritten as

$$\frac{M\mathfrak{m}}{2n} \le \frac{2\mathfrak{c}^2 m^2}{\gamma^2 n}.$$

Maximizing over $\boldsymbol{z}^n$ and then averaging over $\boldsymbol{x}^n$ yields

$$\mathbb{E}[\psi(\boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{y}^{\star n}))] \le \mathbb{E}[\widehat{\lambda}_n] + \frac{2\mathfrak{c}^2 m^2}{\gamma^2 n}, \tag{22}$$

It remains to compare the empirical and population roots. For any input distribution $\mathbb{Q}$, define the corresponding Dinkelbach map by

$$\Phi_{\mathbb{Q}}(\lambda) := \mathbb{E}_{\boldsymbol{x}\sim\mathbb{Q}}\left[\max_{\widehat{\boldsymbol{y}}} \{(\boldsymbol{A} - \lambda \boldsymbol{B}) \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}})\}\right].$$

Thus, if $\mathbb{Q}$ is the empirical distribution on $\boldsymbol{x}^n$, then $\Phi_{\mathbb{Q}} = \widehat{\Phi}_n$, while for the population input distribution, $\mathbb{Q} = \mathbb{P}_{\mathcal{X}}$, we have $\Phi_{\mathbb{Q}} = \Phi$. For a point mass at a single input $\boldsymbol{x}$, we write

$$\Phi_{\boldsymbol{x}}(\lambda) := \Phi_{\delta_{\boldsymbol{x}}}(\lambda) = \max_{\widehat{\boldsymbol{y}}} \{(\boldsymbol{A} - \lambda \boldsymbol{B}) \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}})\}.$$

Fix $\lambda_1 < \lambda_2$ and a prediction $\widehat{\boldsymbol{y}}$. For the single-example mean confusion matrix $\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}})$, the corresponding surrogate satisfies

$$\begin{aligned}
&(\boldsymbol{A} - \lambda_2 \boldsymbol{B}) \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}}) \\
&\quad = (\boldsymbol{A} - \lambda_1 \boldsymbol{B}) \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}}) - (\lambda_2 - \lambda_1) D(\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}})) \\
&\quad \le (\boldsymbol{A} - \lambda_1 \boldsymbol{B}) \cdot \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}}) - \gamma(\lambda_2 - \lambda_1).
\end{aligned}$$

The last inequality uses $D(\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \widehat{\boldsymbol{y}})) \ge \gamma$. Taking the maximum over $\widehat{\boldsymbol{y}}$ gives the pointwise inequality

$$\Phi_{\boldsymbol{x}}(\lambda_2) \le \Phi_{\boldsymbol{x}}(\lambda_1) - \gamma(\lambda_2 - \lambda_1).$$

Averaging over any input law $\mathbb{Q}$ yields

$$\Phi_{\mathbb{Q}}(\lambda_2) \le \Phi_{\mathbb{Q}}(\lambda_1) - \gamma(\lambda_2 - \lambda_1). \tag{23}$$

We apply (23) with $\mathbb{Q}$ equal to the empirical input distribution over $\boldsymbol{x}^n$. If $\widehat{\lambda}_n \ge \lambda^\star$, then

$$0 = \widehat{\Phi}_n(\widehat{\lambda}_n) \le \widehat{\Phi}_n(\lambda^\star) - \gamma(\widehat{\lambda}_n - \lambda^\star),$$

and hence

$$\gamma|\widehat{\lambda}_n - \lambda^\star| \le \widehat{\Phi}_n(\lambda^\star) = \widehat{\Phi}_n(\lambda^\star) - \Phi(\lambda^\star),$$

where the last equality uses $\Phi(\lambda^\star) = 0$. If instead $\widehat{\lambda}_n < \lambda^\star$, then

$$\widehat{\Phi}_n(\lambda^\star) \le \widehat{\Phi}_n(\widehat{\lambda}_n) - \gamma(\lambda^\star - \widehat{\lambda}_n) = -\gamma(\lambda^\star - \widehat{\lambda}_n),$$

and therefore

$$\gamma|\widehat{\lambda}_n - \lambda^\star| \le -\widehat{\Phi}_n(\lambda^\star) = \Phi(\lambda^\star) - \widehat{\Phi}_n(\lambda^\star).$$

Combining the two cases gives:

$$|\widehat{\lambda}_n - \lambda^\star| \le \frac{|\widehat{\Phi}_n(\lambda^\star) - \Phi(\lambda^\star)|}{\gamma} = \frac{1}{\gamma}\left|\frac{1}{n}\sum_{t=1}^{n}\Phi_{\boldsymbol{x}_t}(\lambda^\star) - \mathbb{E}[\Phi_{\boldsymbol{x}}(\lambda^\star)]\right|.$$

It remains only to bound the expectation of the empirical-process term. For $\lambda^\star \in [0,1]$, each candidate surrogate has the form $N(\boldsymbol{C}) - \lambda^\star D(\boldsymbol{C})$ for some feasible $\boldsymbol{C}$. Since $0 \le N(\boldsymbol{C}) \le 1$ and $0 < \gamma \le D(\boldsymbol{C}) \le 1$, every candidate belongs to $[-1,1]$. Its maximum $\Phi_{\boldsymbol{x}}(\lambda^\star)$ also belongs to this interval, and hence $|\Phi_{\boldsymbol{x}}(\lambda^\star)| \le 1$. Consequently, with $Z_t := \Phi_{\boldsymbol{x}_t}(\lambda^\star) - \mathbb{E}[\Phi_{\boldsymbol{x}}(\lambda^\star)]$, the random variables $Z_1, \dots, Z_n$ are independent, mean-zero, scalar random variables and $\mathbb{E}[Z_t^2] \le 1$. Therefore

$$\begin{aligned}
&\mathbb{E}\left|\frac{1}{n}\sum_{t=1}^{n}\Phi_{\boldsymbol{x}_t}(\lambda^\star) - \mathbb{E}[\Phi_{\boldsymbol{x}}(\lambda^\star)]\right| \\
&\le \sqrt{\mathbb{E}\left[\left(\frac{1}{n}\sum_{t=1}^{n}(\Phi_{\boldsymbol{x}_t}(\lambda^\star) - \mathbb{E}\,\Phi_{\boldsymbol{x}}(\lambda^\star))\right)^2\right]} = \frac{1}{n}\sqrt{\sum_{t=1}^{n}\mathbb{E}[Z_t^2]} \le \frac{1}{\sqrt{n}}.
\end{aligned}$$

Consequently,

$$\mathbb{E}\,|\widehat{\lambda}_n - \lambda^\star| \le \frac{1}{\gamma\sqrt{n}}. \tag{24}$$

Combining (22) and (24), and using $\lambda^\star = \psi(\boldsymbol{C}(\boldsymbol{h}^\star))$, proves the stated ETU-PU benchmark gap. □

E.6 ETU regret: comparison through the population root

Theorem 4.2 (ETU regret for LF utilities)**.** *Run SOLO in Algorithm 2. Then, for both label-adaptive and label-oblivious updates,*

$$R_n^{\mathrm{ETU}} \le \frac{4}{\gamma^2\sqrt{n}} + \frac{4}{\gamma}\sqrt{\frac{2\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)} + \frac{4\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n) + \frac{2\mathfrak{c}^2 m^2}{\gamma^2 n}.$$

Proof. We first bound the algorithm's performance relative to the *population optimum* $\lambda^\star$, then apply Lemma E.4. Write $\mathcal{F}_{t-1}$ for the history before round t, including any independent algorithmic randomness needed to determine the current prediction rule. Set

$$\epsilon_t := \|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_1, \qquad \boldsymbol{C}_t := \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t).$$

Let $\widehat{\boldsymbol{y}}_t^\star$ maximize the true-CPE surrogate at $\lambda^\star$, and put

$$\boldsymbol{C}_t^\star := \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t^\star), \qquad Z_t^\star := \psi_{\lambda^\star}(\boldsymbol{C}_t^\star).$$

By Lemma E.2 and the Dinkelbach root identity,

$$\mathbb{E}[Z_t^\star \mid \mathcal{F}_{t-1}] = \Phi(\lambda^\star) = 0. \tag{25}$$

There are two differences between the algorithm's prediction and $\widehat{\boldsymbol{y}}_t^\star$: it uses the estimated conditionals $\widehat{\boldsymbol{\eta}}_t$ and the current parameter λ_{t-1} in place of $\lambda^\star$. We bound these effects separately.

First keep the parameter fixed at λ_{t-1}. Since $\widehat{\boldsymbol{y}}_t$ maximizes the surrogate under $\widehat{\boldsymbol{\eta}}_t$, (19) gives

$$\max_{\widehat{\boldsymbol{y}}} \psi_{\lambda_{t-1}}(\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}})) - \psi_{\lambda_{t-1}}(\boldsymbol{C}_t) \le 4\mathfrak{c}\epsilon_t,$$

and since

$$\psi_{\lambda_{t-1}}(\boldsymbol{C}_t^\star) = \psi_{\lambda_{t-1}}(\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}}_t^\star)) \le \max_{\widehat{\boldsymbol{y}}} \psi_{\lambda_{t-1}}(\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \widehat{\boldsymbol{y}})),$$

we have

$$\psi_{\lambda_{t-1}}(\boldsymbol{C}_t^\star) - \psi_{\lambda_{t-1}}(\boldsymbol{C}_t) \le 4\mathfrak{c}\epsilon_t. \tag{26}$$

Next change the parameter to $\lambda^\star$. From $\psi_\lambda(\boldsymbol{C}) = N(\boldsymbol{C}) - \lambda D(\boldsymbol{C})$, for every $\boldsymbol{C}$,

$$\psi_{\lambda^\star}(\boldsymbol{C}) = \psi_{\lambda_{t-1}}(\boldsymbol{C}) + (\lambda_{t-1} - \lambda^\star) D(\boldsymbol{C}).$$

Apply this identity to $\boldsymbol{C}_t^\star$ and $\boldsymbol{C}_t$, and subtract:

$$\begin{aligned} Z_t^\star - \psi_{\lambda^\star}(\boldsymbol{C}_t) = {} & \psi_{\lambda_{t-1}}(\boldsymbol{C}_t^\star) - \psi_{\lambda_{t-1}}(\boldsymbol{C}_t) \\ & + (\lambda_{t-1} - \lambda^\star)\big(D(\boldsymbol{C}_t^\star) - D(\boldsymbol{C}_t)\big). \end{aligned}$$

The first difference is bounded by (26); the remaining term is at most $|\lambda_{t-1} - \lambda^\star|$, because $0 \le D(\boldsymbol{C}) \le 1$ for every feasible confusion matrix. Hence

$$Z_t^\star - \psi_{\lambda^\star}(\boldsymbol{C}_t) \le 4\mathfrak{c}\epsilon_t + |\lambda_{t-1} - \lambda^\star|. \tag{27}$$

This pointwise comparison will now be averaged over the fresh example and then over the online trajectory.

Define

$$r_t := \psi_{\lambda^\star}(\boldsymbol{C}(\boldsymbol{y}_t, \widehat{\boldsymbol{y}}_t)), \qquad \bar{r}_t := \mathbb{E}[r_t \mid \mathcal{F}_{t-1}], \qquad M_n := \frac{1}{n}\sum_{t=1}^n (r_t - \bar{r}_t).$$

Taking conditional expectations in (27) and using (25) yields

$$\bar{r}_t \ge -|\lambda_{t-1} - \lambda^\star| - 4\mathfrak{c}\,\mathbb{E}[\epsilon_t \mid \mathcal{F}_{t-1}]. \tag{28}$$

Because $|r_t| \le 1$, the martingale differences $r_t - \bar{r}_t$ have conditional variance at most one. Orthogonality and Cauchy–Schwarz give

$$\mathbb{E}\,|M_n| \le \sqrt{\mathbb{E}[M_n^2]} \le \frac{1}{\sqrt{n}}.$$

Writing $\boldsymbol{C}_n := \boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n)$ and $S_n := n^{-1}\sum_t r_t = N(\boldsymbol{C}_n) - \lambda^\star D(\boldsymbol{C}_n)$, from (28) we have

$$S_n = \frac{1}{n}\sum_{t=1}^n \bar{r}_t + M_n \ge -|\lambda_{t-1} - \lambda^\star| - 4\mathfrak{c}\,\mathbb{E}[\epsilon_t \mid \mathcal{F}_{t-1}] + M_n,$$

so that

$$(-S_n)_+ \le \frac{1}{n}\sum_{t=1}^n |\lambda_{t-1} - \lambda^\star| + \frac{4\mathfrak{c}}{n}\sum_{t=1}^n \mathbb{E}[\epsilon_t \mid \mathcal{F}_{t-1}] + |M_n|,$$

where $(a)_+ := \max\{a, 0\}$ (the positive part is needed here, because the realized utility can exceed the population optimum). Since $D(\boldsymbol{C}_n) \ge \gamma$,

$$\lambda^\star - \psi(\boldsymbol{C}_n) = \frac{-S_n}{D(\boldsymbol{C}_n)} \le \frac{(-S_n)_+}{\gamma}.$$

Taking expectations and using Lemma E.3 now gives

$$\begin{aligned} \lambda^\star - \mathbb{E}[\psi(\boldsymbol{C}_n)] &\le \frac{1}{\gamma n}\sum_{t=1}^n \mathbb{E}\,|\lambda_{t-1} - \lambda^\star| + \frac{4\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n) + \frac{1}{\gamma\sqrt{n}} \\ &\le \frac{2}{\gamma^2\sqrt{n}} + \frac{4}{\gamma}\sqrt{\frac{2\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)} + \frac{4\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n) + \frac{1}{\gamma\sqrt{n}}. \end{aligned} \tag{29}$$

This argument applies to both update versions (label-adaptive and label-oblivious): only the root-tracking bound uses the particular update, while Lemma E.3 covers both.

By Lemma E.4,

$$\mathbb{E}[\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n)] - \lambda^\star \le \frac{1}{\gamma\sqrt{n}} + \frac{2\mathfrak{c}^2 m^2}{\gamma^2 n}.$$

Adding this to (29) proves the claimed bound, since $2/(\gamma^2\sqrt{n}) + 2/(\gamma\sqrt{n}) \le 4/(\gamma^2\sqrt{n})$ for $\gamma \le 1$. □

E.7 PU regret of the final classifier

Theorem 4.3 (PU regret for LF utilities)**.** *Return the classifier induced by the final root estimate:*

$$\widehat{\boldsymbol{h}}_n(\boldsymbol{x}) \in \operatorname*{argmax}_{\widehat{\boldsymbol{y}}} (\boldsymbol{A} - \lambda_n \boldsymbol{B}) \cdot \mathbb{E}_{\boldsymbol{y}\sim\widehat{\boldsymbol{\eta}}_{n+1}(\boldsymbol{x})}[\boldsymbol{C}(\boldsymbol{y}, \widehat{\boldsymbol{y}})].$$

Let

$$\mathrm{Err}(\widehat{\boldsymbol{\eta}}_{n+1}) = \mathbb{E}\,\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}_{n+1}(\boldsymbol{x})\|_1$$

be the one-step analogue of (6)*. Then*

$$R_n^{\mathrm{PU}} \le \frac{1}{\gamma}\left[\frac{1}{\gamma\sqrt{n+1}} + 2\sqrt{\frac{2\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)}\right] + \frac{4\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}_{n+1}).$$

Proof. Let $\boldsymbol{C}^\star := \boldsymbol{C}(\boldsymbol{h}^\star)$ and $\widehat{\boldsymbol{C}}_n := \boldsymbol{C}(\widehat{\boldsymbol{h}}_n)$. Condition on the online trajectory, so that λ_n and $\widehat{\boldsymbol{\eta}}_{n+1}$ are fixed. Let $e_{n+1} := \mathbb{E}_{\boldsymbol{x}}\,\|\boldsymbol{\eta}(\boldsymbol{x}) - \widehat{\boldsymbol{\eta}}_{n+1}(\boldsymbol{x})\|_1$ be the CPE estimation error given the training trajectory; note that $\mathrm{Err}(\widehat{\boldsymbol{\eta}}_{n+1}) := \mathbb{E}[e_{n+1}]$ is its full expectation.

The returned classifier maximizes the λ_n-surrogate ψ_{λ_n} under the CPE $\widehat{\boldsymbol{\eta}}_{n+1}$. By Lemma E.2, its λ_n-surrogate under true conditional probability $\boldsymbol{\eta}$ is within $4\mathfrak{c}\,e_{n+1}$ of the best λ_n-surrogate:

$$\psi_{\lambda_n}(\boldsymbol{C}^\star) - \psi_{\lambda_n}(\widehat{\boldsymbol{C}}_n) \le 4\mathfrak{c}\,e_{n+1}.$$

Since $\psi_{\lambda^\star}(\boldsymbol{C}^\star) = 0$, we can write

$$\begin{aligned}
-\psi_{\lambda^\star}(\widehat{\boldsymbol{C}}_n) &= \psi_{\lambda^\star}(\boldsymbol{C}^\star) - \psi_{\lambda^\star}(\widehat{\boldsymbol{C}}_n) \\
&= \psi_{\lambda_n}(\boldsymbol{C}^\star) - \psi_{\lambda_n}(\widehat{\boldsymbol{C}}_n) + (\lambda_n - \lambda^\star)\big(D(\boldsymbol{C}^\star) - D(\widehat{\boldsymbol{C}}_n)\big) \\
&\le 4\mathfrak{c}\,e_{n+1} + |\lambda_n - \lambda^\star|.
\end{aligned}$$

Now

$$\psi(\boldsymbol{C}^\star) - \psi(\widehat{\boldsymbol{C}}_n) = \lambda^\star - \frac{N(\widehat{\boldsymbol{C}}_n)}{D(\widehat{\boldsymbol{C}}_n)} = \frac{-(N(\widehat{\boldsymbol{C}}_n) - \lambda^\star D(\widehat{\boldsymbol{C}}_n))}{D(\widehat{\boldsymbol{C}}_n)} = \frac{-\psi_{\lambda^\star}(\widehat{\boldsymbol{C}}_n)}{D(\widehat{\boldsymbol{C}}_n)}.$$

Using $D(\widehat{\boldsymbol{C}}_n) \ge \gamma$ gives

$$\psi(\boldsymbol{C}^\star) - \psi(\widehat{\boldsymbol{C}}_n) \le \frac{1}{\gamma}|\lambda_n - \lambda^\star| + \frac{4\mathfrak{c}}{\gamma}e_{n+1}.$$

Taking expectations, using $\mathbb{E}[e_{n+1}] = \mathrm{Err}(\widehat{\boldsymbol{\eta}}_{n+1})$, it remains only to bound the root error. From Lemma E.3,

$$\mathbb{E}\,|\lambda_n - \lambda^\star| \le \sqrt{\mathbb{E}[(\lambda_n - \lambda^\star)^2]} \le \frac{1}{\gamma\sqrt{n+1}} + 2\sqrt{\frac{2\mathfrak{c}}{\gamma}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)}.$$

Substituting this into the preceding inequality proves the theorem. □

E.8 Population lower bound

Theorem 4.4 (PU lower bound for returned classifier)**.** *For regularized precision $\psi(C) = \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}+1/2}$, there is $c > 0$, such that for every algorithm that observes n i.i.d. examples, knows the exact conditional probability $\eta(\boldsymbol{x})$, and returns a classifier $\widehat{\boldsymbol{h}}_n$, there exists a distribution on which*

$$R_n^{\mathrm{PU}} = \psi(\boldsymbol{C}(\boldsymbol{h}^\star)) - \mathbb{E}[\psi(\boldsymbol{C}(\widehat{\boldsymbol{h}}_n))] \ge \frac{c}{\sqrt{n}}.$$

Thus the LF population rate is unimprovable in the general case.

Proof. We use the standard two-point version of Le Cam's method [Le Cam and Yang, 2000, Tsybakov, 2009]. The hard part of the problem is not estimating the conditional label probabilities, as they are identical under the two hypotheses and revealed to the learner. The learner only has to infer the input marginal probability accurately enough to choose the correct population classifier. We allow randomized learning algorithms throughout, conditioning also on their independent random seed when describing the returned classifier.

Consider binary classification ($y \in \{0, 1\}$) on the two-point instance space $\mathcal{X} = \{x_1, x_2\}$ with the regularized precision utility:

$$\psi(\boldsymbol{C}) = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp} + \frac{1}{2}}.$$

For $\pi \in (0, 1)$, define the distribution $\mathbb{P}_\pi(x, y)$ by

$$\mathbb{P}_\pi(x_1) = \pi, \qquad \mathbb{P}_\pi(x_2) = 1 - \pi, \qquad \mathbb{P}_\pi(y = 1 | x_i) = \eta(x_i),$$

and let

$$u := \eta(x_1) = \tfrac{3}{4}, \qquad v := \eta(x_2) = \tfrac{1}{4}.$$

The conditional probability function $\eta(\cdot)$ is the same for every π, so knowing the exact probabilities does not reveal which input marginal is in play.

Take an arbitrary randomized algorithm that observes n i.i.d. examples $(x, y)^n$ and returns a randomized classifier $\widehat{\boldsymbol{h}}_n$. Conditional on the sample $(x, y)^n$ and the algorithm's random seed, the returned classifier is described by two probabilities $q_1, q_2 \in [0, 1]$, where q_i is the probability of predicting the positive label on x_i. Its population utility under $\mathbb{P}_\pi$, $\psi(\boldsymbol{C}(\widehat{\boldsymbol{h}}_n))$, is

$$\psi_\pi(q_1, q_2) := \frac{\pi u q_1 + (1 - \pi) v q_2}{\pi q_1 + (1 - \pi) q_2 + \frac{1}{2}}.$$

For fixed π and q_2,

$$\frac{\partial}{\partial q_1} \psi_\pi(q_1, q_2) = \frac{\pi\big((u - v)(1 - \pi) q_2 + \frac{1}{2} u\big)}{\big(\pi q_1 + (1 - \pi) q_2 + \frac{1}{2}\big)^2} > 0.$$

Thus predicting positive on x_1 more often can only improve utility. For a lower bound we may therefore assume without loss of generality that the returned classifier predicts with $q_1 = 1$, as this can only decrease its regret. Denoting the remaining parameter q_2 simply by q, the utility becomes:

$$\psi_\pi(q) := \psi_\pi(1, q) = \frac{\pi u + (1 - \pi) v q}{\pi + (1 - \pi) q + \frac{1}{2}}.$$

Differentiating with respect to q gives

$$\frac{\partial}{\partial q} \psi_\pi(q) = \frac{(1 - \pi)\big(\frac{1}{2} v - (u - v)\pi\big)}{\big(\pi + (1 - \pi) q + \frac{1}{2}\big)^2}.$$

The sign changes at

$$\pi_0 := \frac{\frac{1}{2} v}{u - v} = \frac{1}{4}.$$

Therefore, if $\pi > \pi_0$, the optimal classifier "rejects" x_2 and has $q = 0$; if $\pi < \pi_0$, the optimal classifier "accepts" x_2 and has $q = 1$.

Set

$$\varepsilon := \frac{1}{32\sqrt{n}}, \qquad \pi_+ := \pi_0 + \varepsilon, \qquad \pi_- := \pi_0 - \varepsilon.$$

For all $n \geq 1$, $\pi_\pm \in [1/8, 3/8]$. Under π_+, direct algebraic manipulations give for every $q \in [0, 1]$,

$$\begin{aligned} \psi_{\pi_+}(0) - \psi_{\pi_+}(q) &= \frac{q(1 - \pi_+)(u - v)(\pi_+ - \pi_0)}{(\pi_+ + \frac{1}{2})(\pi_+ + (1 - \pi_+) q + \frac{1}{2})} \\ &\geq \frac{5}{21} \varepsilon q, \end{aligned}$$

where in the above we used $1-\pi_+ \geq 5/8$, $u-v = 1/2$, $\pi_+ + \frac{1}{2} \leq 7/8$, and $\pi_+ + (1-\pi_+)q + \frac{1}{2} \leq 3/2$. Hence

$$\mathbb{E}_{\pi_+}[\psi(\boldsymbol{C}(\boldsymbol{h}^\star)) - \psi(\boldsymbol{C}(\widehat{\boldsymbol{h}}_n))] \geq \frac{5}{21}\varepsilon\, \mathbb{E}_{\pi_+}[Q],$$

where Q is a (sample-dependent) random variable, realization of which is q. Under π_-, similarly,

$$\begin{aligned}\psi_{\pi_-}(1) - \psi_{\pi_-}(q) &= \frac{(1-q)(1-\pi_-)(u-v)(\pi_0-\pi_-)}{(1+\frac{1}{2})(\pi_- + (1-\pi_-)q + \frac{1}{2})} \\ &\geq \frac{5}{36}\,\varepsilon(1-q),\end{aligned}$$

because $1-\pi_- \geq 5/8$ and both denominator factors are at most $1+\frac{1}{2} = 3/2$. Thus

$$\mathbb{E}_{\pi_-}[\psi(\boldsymbol{C}(\boldsymbol{h}^\star)) - \psi(\boldsymbol{C}(\widehat{\boldsymbol{h}}_n))] \geq \frac{5}{36}\varepsilon\, \mathbb{E}_{\pi_-}[1-Q].$$

From now on, we use standard lower bounding arguments. Let $\mathbb{P}^n_{\pi_+}$ and $\mathbb{P}^n_{\pi_-}$ be the product distributions of the sample $(x,y)^n$ for two environments. Since the conditional label distribution is identical for both environments, the chain rule for KL divergence and independence of examples gives

$$\mathrm{KL}(\mathbb{P}^n_{\pi_+} \| \mathbb{P}^n_{\pi_-}) = n\,\mathrm{kl}(\pi_+ \| \pi_-),$$

where kl is the Bernoulli (binary) KL divergence:

$$\mathrm{kl}(p\|q) = p\ln\frac{p}{q} + (1-p)\ln\frac{1-p}{1-q}.$$

Using the second-order Taylor expansion around $p=q$, and upper-bounding the quadratic term, $\mathrm{kl}(p\|q) \leq \frac{(p-q)^2}{\min\{p(1-p), q(1-q)\}}$, which in our case gives:

$$\mathrm{kl}(\pi_+\|\pi_-) \leq \frac{(\pi_+ - \pi_-)^2}{\min\{\pi_-(1-\pi_-), \pi_+(1-\pi_+)\}} \leq 64\varepsilon^2.$$

as $\pi_-, \pi_+ \in [1/8, 3/8]$. Consequently,

$$\mathrm{KL}(\mathbb{P}^n_{\pi_+} \| \mathbb{P}^n_{\pi_-}) \leq 64n\varepsilon^2 = \frac{1}{16}.$$

Using Pinsker's inequality

$$\mathrm{TV}(\mathbb{P}^n_{\pi_+}, \mathbb{P}^n_{\pi_-}) \leq \sqrt{\frac{1}{2}\mathrm{KL}(\mathbb{P}^n_{\pi_+} \| \mathbb{P}^n_{\pi_-})} \leq \sqrt{\frac{1}{32}}.$$

Appending the algorithm's independent random seed to the sample does not change the total variation distance. Since $Q \in [0,1]$, $|\,\mathbb{E}_{\pi_+}[Q] - \mathbb{E}_{\pi_-}[Q]| \leq \mathrm{TV}(\mathbb{P}^n_{\pi_+}, \mathbb{P}^n_{\pi_-})$. Therefore

$$\mathbb{E}_{\pi_+}[Q] + \mathbb{E}_{\pi_-}[1-Q] = 1 + \mathbb{E}_{\pi_+}[Q] - \mathbb{E}_{\pi_-}[Q] \geq 1 - \sqrt{\frac{1}{32}} =: c_0.$$

Combining the two regret lower bounds,

$$\begin{aligned}&\mathbb{E}_{\pi_+}[\psi(\boldsymbol{C}(\boldsymbol{h}^\star)) - \psi(\boldsymbol{C}(\widehat{\boldsymbol{h}}_n))] + \mathbb{E}_{\pi_-}[\psi(\boldsymbol{C}(\boldsymbol{h}^\star)) - \psi(\boldsymbol{C}(\widehat{\boldsymbol{h}}_n))] \\ &\quad\geq \frac{5}{36}\varepsilon\left(\mathbb{E}_{\pi_+}[Q] + \mathbb{E}_{\pi_-}[1-Q]\right) \geq \frac{5c_0}{36}\cdot\frac{1}{32\sqrt{n}}.\end{aligned}$$

At least one of $\mathbb{P}_{\pi_+}$ and $\mathbb{P}_{\pi_-}$ therefore has expected PU regret at least $c/\sqrt{n}$. □

F LF Utilities with Label-Frequency Dependent Denominator Bounds

The global assumption $D(\boldsymbol{C}) \geq \gamma$ for all $\boldsymbol{C} \in \mathcal{C}$ is stronger than what is needed for many unregularized LF metrics. For instance, the denominator of recall, F-measure, and Jaccard contains the positive-label mass, and is therefore bounded away from zero only on distributions whose positive-label frequency is bounded away from zero. This section covers the corresponding extension of

the LF analysis. For simplicity, we state it only for the label-adaptive version of SOLO. This is essentially the LF specialization of the $\boldsymbol{p}$-Lipschitzness and $\boldsymbol{p}$-smoothness framework [Dembczynski et al., 2017, Kotlowski et al., 2024].

We retain the normalization $0 \le N(\boldsymbol{C}) \le D(\boldsymbol{C}) \le 1$ and $\|\boldsymbol{A}\|_{\max}, \|\boldsymbol{B}\|_{\max} \le 1$, but replace global denominator positivity by Assumption F.1. If $D(\boldsymbol{C}) = 0$, assign the utility any fixed value in $[0, 1]$; the bounds below hold for every such convention.

The population and finite-sample arguments use this assumption differently. At the population level, positive label frequencies give a denominator bound for every classifier, so the root-tracking and PU proofs carry over directly. For ETU, empirical frequencies can be small; we isolate those rare samples and pay an exponentially small error. We establish the population result first, then the frequency and benchmark bounds, and finally the ETU guarantee.

F.1 Denominator bounds from label frequencies

For a confusion matrix or tensor $\boldsymbol{C}$, define the label-frequency vector $\boldsymbol{p}(\boldsymbol{C}) \in [0, 1]^m$ by

$$p_k(\boldsymbol{C}) = \begin{cases} \sum_{j=1}^{m} C_{kj}, & \text{multi-class,} \\ C_{k,11} + C_{k,10}, & \text{multi-label.} \end{cases}$$

Thus, if $\boldsymbol{C}(\boldsymbol{h})$ is calculated over data distribution $\mathbb{P}$, $p_k(\boldsymbol{C}) = \mathbb{P}(y_k = 1)$ is the prior probability of class k in the multi-class case, or the prior marginal probability of label k in the multi-label case. Likewise, if $\boldsymbol{C} = \boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n)$ is empirical, then $p_k(\boldsymbol{C}) = n^{-1} \sum_{t=1}^{n} y_{tk}$ and is independent of the predictions.

For $\rho > 0$, write

$$\mathcal{C}_\rho := \{\boldsymbol{C} \in \mathcal{C} : \min_k p_k(\boldsymbol{C}) \ge \rho\}.$$

For an input distribution $\mathbb{Q}$, let

$$\boldsymbol{p}_{\mathbb{Q}} := \mathbb{E}_{\boldsymbol{x} \sim \mathbb{Q}}[\boldsymbol{\eta}(\boldsymbol{x})],$$

The population vector is denoted by

$$\boldsymbol{p}^\star = \mathbb{E}_{\mathbb{P}}[\boldsymbol{y}] \qquad (\text{so that } p_k^\star = \mathbb{P}(y_k = 1)),$$

and we write $p_{\min}^\star := \min_k p_k^\star$.

Assumption F.1. Assume $p_{\min}^\star > 0$. For every $\rho \in (0, p_{\min}^\star)$ there is a constant $\gamma_\rho > 0$ such that

$$D(\boldsymbol{C}) \ge \gamma_\rho \qquad \text{for all } \boldsymbol{C} \in \mathcal{C}_\rho.$$

For binary or micro-averaged recall, F-measure, and Jaccard, one may take γ_ρ simply proportional to ρ. Precision is different: its denominator is the predicted-positive mass, not the positive-label mass, so it still requires explicit regularization or a restriction on the predicted-positive mass.

Lemma F.2 (Local Lipschitzness and smoothness)**.** *Under Assumption F.1, the LF utility is L_ρ-Lipschitz and M_ρ-smooth on $\mathcal{C}_\rho$, with*

$$L_\rho \le \frac{2m}{\gamma_\rho}, \qquad M_\rho \le \frac{4m^2}{\gamma_\rho^2} \quad \textit{(multi-class)},$$

and

$$L_\rho \le \frac{4\sqrt{m}}{\gamma_\rho}, \qquad M_\rho \le \frac{16m}{\gamma_\rho^2} \quad \textit{(multi-label)}.$$

Proof. The proof is identical to that of Lemma E.1, except that the lower bound $D(\boldsymbol{C}) \ge \gamma$ is used only for $\boldsymbol{C} \in \mathcal{C}_\rho$ and is replaced by $D(\boldsymbol{C}) \ge \gamma_\rho$. Since $\mathcal{C}_\rho$ is convex, the Hessian bound holds along every line segment contained in this region, giving smoothness on $\mathcal{C}_\rho$. □

F.2 Population root tracking and PU regret

Population confusion matrices always have the fixed label-frequency vector $\boldsymbol{p}^\star$, irrespective of the classifier. Therefore their denominators are at least γ_ρ for every $\rho < p^\star_{\min}$. This is enough for root tracking and the PU guarantee: neither requires concentration of empirical label frequencies.

Lemma F.3 (Dinkelbach root under a fixed input distribution)**.** *Fix an input distribution $\mathbb{Q}$ such that $\min_k p_{\mathbb{Q},k} \geq \rho$ and define*

$$\Phi_{\mathbb{Q}}(\lambda) := \sup_{\boldsymbol{h}} \{N(\boldsymbol{C}_{\mathbb{Q}}(\boldsymbol{h})) - \lambda D(\boldsymbol{C}_{\mathbb{Q}}(\boldsymbol{h}))\},$$

where $\boldsymbol{C}_{\mathbb{Q}}(\boldsymbol{h}) = \mathbb{E}_{\boldsymbol{x}\sim\mathbb{Q}}[\boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}), \boldsymbol{h}(\boldsymbol{x}))]$. Let $\lambda^\star_{\mathbb{Q}} := \sup_{\boldsymbol{h}} \psi(\boldsymbol{C}_{\mathbb{Q}}(\boldsymbol{h}))$. Then $\Phi_{\mathbb{Q}}(\lambda^\star_{\mathbb{Q}}) = 0$ and, for all $\lambda \in [0, 1]$,

$$(\lambda - \lambda^\star_{\mathbb{Q}})\Phi_{\mathbb{Q}}(\lambda) \leq -\gamma_\rho(\lambda - \lambda^\star_{\mathbb{Q}})^2.$$

Proof. For every classifier $\boldsymbol{h}$, the confusion matrix $\boldsymbol{C}_{\mathbb{Q}}(\boldsymbol{h})$ has label-frequency vector $\boldsymbol{p}_{\mathbb{Q}}$, because label frequencies do not depend on the prediction rule. Hence $D(\boldsymbol{C}_{\mathbb{Q}}(\boldsymbol{h})) \geq \gamma_\rho$. The identity $\psi_\lambda(\boldsymbol{C}_{\mathbb{Q}}(\boldsymbol{h})) = D(\boldsymbol{C}_{\mathbb{Q}}(\boldsymbol{h}))(\psi(\boldsymbol{C}_{\mathbb{Q}}(\boldsymbol{h})) - \lambda)$ therefore gives the root and strong-sign claims by the argument of Proposition 4.1, with γ_ρ in place of γ. □

Lemma F.4 (Label-adaptive root tracking under $\boldsymbol{p}$-dependent denominator bounds)**.** *Fix $\rho \in (0, p^\star_{\min})$ and run the label-adaptive version of SOLO with stepsize $\alpha_t = 1/(\gamma_\rho(t+1))$. Then*

$$\mathbb{E}[(\lambda_t - \lambda^\star)^2] \leq \frac{1}{\gamma_\rho^2(t+1)} + \frac{8\mathfrak{c}}{\gamma_\rho}\frac{1}{t}\sum_{s=1}^{t}\mathbb{E}\,\|\widehat{\boldsymbol{\eta}}_s(\boldsymbol{x}_s) - \boldsymbol{\eta}(\boldsymbol{x}_s)\|_1,$$

and consequently

$$\frac{1}{n}\sum_{t=1}^{n}\mathbb{E}\,|\lambda_{t-1} - \lambda^\star| \leq \frac{2}{\gamma_\rho\sqrt{n}} + 4\sqrt{\frac{2\mathfrak{c}}{\gamma_\rho}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)}.$$

Proof. The proof of Lemma E.3 uses the denominator lower bound only through the population strong-sign inequality for Φ. Since every population confusion matrix $\boldsymbol{C}(\boldsymbol{h})$ has label-frequency vector $\boldsymbol{p}^\star$ and $\boldsymbol{p}^\star \geq p^\star_{\min}\mathbf{1} \geq \rho\mathbf{1}$, Lemma F.3 gives this strong-sign inequality with γ_ρ. The CPE perturbation part is unchanged because Lemma E.2 uses only $\|\boldsymbol{A} - \lambda\boldsymbol{B}\|_{\max} \leq 2$. Replacing γ by γ_ρ in the recursion of Lemma E.3 therefore proves the result for the label-adaptive update. □

Theorem F.5 (PU regret for label-adaptive SOLO with $\boldsymbol{p}$-dependent denominator bounds)**.** *Fix $\rho \in (0, p^\star_{\min})$ and run the label-adaptive version of SOLO with stepsize $\alpha_t = 1/(\gamma_\rho(t+1))$. Return the final root-induced classifier*

$$\widehat{\boldsymbol{h}}_n(\boldsymbol{x}) \in \operatorname*{argmax}_{\widehat{\boldsymbol{y}}}(\boldsymbol{A} - \lambda_n\boldsymbol{B}) \cdot \boldsymbol{C}(\widehat{\boldsymbol{\eta}}_{n+1}(\boldsymbol{x}), \widehat{\boldsymbol{y}}).$$

Then

$$R^{\mathrm{PU}}_n \leq \frac{1}{\gamma_\rho}\left[\frac{1}{\gamma_\rho\sqrt{n+1}} + 2\sqrt{\frac{2\mathfrak{c}}{\gamma_\rho}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)}\right] + \frac{4\mathfrak{c}}{\gamma_\rho}\mathrm{Err}(\widehat{\boldsymbol{\eta}}_{n+1}).$$

Proof. The proof of Theorem 4.3 applies without change. The population confusion matrices of both $\boldsymbol{h}^\star$ and $\widehat{\boldsymbol{h}}_n$ have label-frequency vector $\boldsymbol{p}^\star$, hence their denominators are at least γ_ρ. The root-tracking term is controlled by Lemma F.4, and the CPE perturbation term is controlled by Lemma E.2. □

F.3 Label-frequency concentration and the ETU benchmark

For ETU regret we also need to control confusion matrices formed from a finite sample. There are two relevant frequency vectors: $n^{-1}\sum_t \boldsymbol{\eta}(\boldsymbol{x}_t)$ for conditional mean matrices, and $n^{-1}\sum_t \boldsymbol{y}_t$ for realized matrices. The event E below controls the first and ensures that the empirical Dinkelbach root exists with the required denominator bound. The event F controls the second and allows us to convert the algorithm's realized surrogate into utility. Both events depend only on the data, not on the predictions.

We use an intermediate threshold $\bar{\rho}$ between ρ and $p^\star_{\min}$. This leaves room for labels to fluctuate around their conditional means while still remaining above ρ. The midpoint choice makes the two Hoeffding bounds identical.

Lemma F.6 (Label-frequency concentration)**.** *Fix $\rho \in (0, p^\star_{\min})$ and set*

$$\bar{\rho} := \frac{p^\star_{\min} + \rho}{2}, \qquad \delta_n(\rho) := m \exp\left(-\frac{n(p^\star_{\min} - \rho)^2}{2}\right).$$

Let

$$E := \left\{ \min_k \frac{1}{n} \sum_{t=1}^{n} \eta_k(\boldsymbol{x}_t) \geq \bar{\rho} \right\},$$
$$F := \left\{ \min_k \frac{1}{n} \sum_{t=1}^{n} y_{tk} \geq \rho \right\}.$$

Then

$$\mathbb{P}(E^c) \leq \delta_n(\rho), \qquad \mathbb{P}(F^c \mid \boldsymbol{x}^n) \leq \delta_n(\rho) \quad \textit{for all } \boldsymbol{x}^n \in E.$$

In particular, $\mathbb{P}(F^c) \leq 2\delta_n(\rho)$.

Proof. For the first claim, the random variables $\eta_k(\boldsymbol{x}_t) \in [0,1]$ are i.i.d. with mean $p^\star_k \geq p^\star_{\min}$. Hoeffding's inequality and a union bound over $k = 1, \ldots, m$ give

$$\mathbb{P}(E^c) \leq m \exp\{-2n(p^\star_{\min} - \bar{\rho})^2\} = \delta_n(\rho).$$

For the second claim, condition on an input sequence in E. For every k, the conditionally independent variables $y_{1k}, \ldots, y_{nk}$ have average conditional mean at least $\bar{\rho}$. Hoeffding's inequality for independent bounded variables with nonidentical means gives

$$\mathbb{P}\left(\frac{1}{n} \sum_{t=1}^{n} y_{tk} < \rho \,\middle|\, \boldsymbol{x}^n \right) \leq \exp\{-2n(\bar{\rho} - \rho)^2\} = \exp\left(-\frac{n(p^\star_{\min} - \rho)^2}{2}\right).$$

A union bound over $k = 1, \ldots, m$ gives the conditional claim. Finally,

$$\mathbb{P}(F^c) \leq \mathbb{P}(E^c) + \mathbb{E}[\mathbf{1}_E \mathbb{P}(F^c \mid \boldsymbol{x}^n)] \leq 2\delta_n(\rho).$$

□

Lemma F.7 (ETU–PU benchmark gap under local denominator bounds)**.** *Fix $\rho \in (0, p^\star_{\min})$ and let L_ρ, M_ρ and $\delta_n(\rho)$ be as above. Then*

$$\mathbb{E}[\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n)] - \lambda^\star \leq \frac{1}{\gamma_\rho \sqrt{n}} + \frac{M_\rho \mathfrak{m}}{2n} + \big(2 + L_\rho \sqrt{2\mathfrak{m}}\big)\delta_n(\rho).$$

Proof. As in Lemma E.4, we compare the ETU benchmark with an empirical Dinkelbach root and then compare that root with $\lambda^\star$. Here the denominator bound and smoothness are available only when label frequencies are sufficiently large. We therefore carry out the comparison for input sequences in E, where the conditional label frequencies are large enough, and bound the contribution of E^c separately.

Fix $\boldsymbol{x}^n \in E$, where E is defined in Lemma F.6. For a deterministic prediction sequence $\boldsymbol{z}^n$, write

$$\bar{\boldsymbol{C}}(\boldsymbol{z}^n; \boldsymbol{x}^n) := \frac{1}{n} \sum_{t=1}^{n} \boldsymbol{C}(\boldsymbol{\eta}(\boldsymbol{x}_t), \boldsymbol{z}_t), \qquad \widehat{\Phi}_n(\lambda) := \max_{\boldsymbol{z}^n} \psi_\lambda(\bar{\boldsymbol{C}}(\boldsymbol{z}^n; \boldsymbol{x}^n)).$$

Because the surrogate is linear, the maximum over $\boldsymbol{z}^n$ can be taken separately for each example. Thus $\widehat{\Phi}_n$ is the same empirical Dinkelbach map as in the proof of Lemma E.4. Every matrix $\bar{\boldsymbol{C}}(\boldsymbol{z}^n; \boldsymbol{x}^n)$ has label frequencies at least $\bar{\rho}$ on E, so its denominator is at least γ_ρ. Thus Lemma F.3 gives a unique root $\widehat{\lambda}_n \in [0,1]$, and

$$\psi(\bar{\boldsymbol{C}}(\boldsymbol{z}^n; \boldsymbol{x}^n)) \leq \widehat{\lambda}_n \qquad \text{for every } \boldsymbol{z}^n.$$

For completeness, define $\widehat{\lambda}_n$ arbitrarily in $[0,1]$ on E^c; we will use its root property only on E.

To control this root, observe that, for $\lambda_1 < \lambda_2$,

$$\psi_{\lambda_2}(\bar{\boldsymbol{C}}) = \psi_{\lambda_1}(\bar{\boldsymbol{C}}) - (\lambda_2 - \lambda_1)D(\bar{\boldsymbol{C}}) \le \psi_{\lambda_1}(\bar{\boldsymbol{C}}) - \gamma_\rho(\lambda_2 - \lambda_1).$$

Maximizing over prediction sequences gives, on E,

$$\widehat{\Phi}_n(\lambda_2) \le \widehat{\Phi}_n(\lambda_1) - \gamma_\rho(\lambda_2 - \lambda_1). \tag{30}$$

The root comparison in the proof of Lemma E.4 therefore applies on E:

$$|\widehat{\lambda}_n - \lambda^\star| \le \frac{|\widehat{\Phi}_n(\lambda^\star) - \Phi(\lambda^\star)|}{\gamma_\rho}.$$

Recall that $\widehat{\Phi}_n(\lambda^\star) = n^{-1}\sum_t \Phi_{\boldsymbol{x}_t}(\lambda^\star)$ and $\Phi(\lambda^\star) = 0$. The terms in this average are independent, have mean zero, and belong to $[-1, 1]$. Hence

$$\mathbb{E}\,|\widehat{\Phi}_n(\lambda^\star) - \Phi(\lambda^\star)| \le \sqrt{\mathbb{E}[\widehat{\Phi}_n(\lambda^\star)^2]} \le \frac{1}{\sqrt{n}}.$$

This expectation is over all input samples. To use it with the root bound, which holds only on E, we multiply that bound by $\mathbf{1}_E$ and then use $\mathbf{1}_E \le 1$. This gives

$$\mathbb{E}\big[|\widehat{\lambda}_n - \lambda^\star|\mathbf{1}_E\big] \le \frac{1}{\gamma_\rho\sqrt{n}}. \tag{31}$$

We next compare the expected utility over labels with the utility of the conditional mean confusion matrix. This is where local smoothness is needed. Still fixing $\boldsymbol{x}^n \in E$ and $\boldsymbol{z}^n$, abbreviate

$$\boldsymbol{C}_n := \boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{z}^n), \qquad \bar{\boldsymbol{C}} := \bar{\boldsymbol{C}}(\boldsymbol{z}^n; \boldsymbol{x}^n), \qquad \boldsymbol{V} := \boldsymbol{C}_n - \bar{\boldsymbol{C}}.$$

Conditional on $\boldsymbol{x}^n$, $\mathbb{E}[\boldsymbol{V} \mid \boldsymbol{x}^n] = 0$ and $\mathbb{E}[\|\boldsymbol{V}\|_2^2 \mid \boldsymbol{x}^n] \le \mathfrak{m}/n$, by the same variance calculation as in Lemma C.2. The conditional mean $\bar{\boldsymbol{C}}$ belongs to $\mathcal{C}_\rho$ because $\boldsymbol{x}^n \in E$. The realized matrix $\boldsymbol{C}_n$ also belongs to this region whenever F holds. Local smoothness therefore gives, on F,

$$\psi(\boldsymbol{C}_n) \le \psi(\bar{\boldsymbol{C}}) + \nabla\psi(\bar{\boldsymbol{C}}) \cdot \boldsymbol{V} + \frac{M_\rho}{2}\|\boldsymbol{V}\|_2^2.$$

If this inequality held for all label realizations, the linear term would vanish after taking conditional expectations, since $\mathbb{E}[\boldsymbol{V} \mid \boldsymbol{x}^n] = 0$. Here we can apply it only on F, so we must retain the indicator $\mathbf{1}_F$. Using the zero expectation over all labels, we can express the remaining linear term through F^c:

$$\mathbb{E}[\nabla\psi(\bar{\boldsymbol{C}}) \cdot \boldsymbol{V}\mathbf{1}_F \mid \boldsymbol{x}^n] = -\,\mathbb{E}[\nabla\psi(\bar{\boldsymbol{C}}) \cdot \boldsymbol{V}\mathbf{1}_{F^c} \mid \boldsymbol{x}^n].$$

By Lemma F.2, $\|\nabla\psi(\bar{\boldsymbol{C}})\|_2 \le L_\rho$, and the diameter of the confusion-matrix space gives $\|\boldsymbol{V}\|_2 \le \sqrt{2\mathfrak{m}}$. Thus the absolute value of this conditional expectation is at most $L_\rho\sqrt{2\mathfrak{m}}\,\mathbb{P}(F^c \mid \boldsymbol{x}^n)$. The quadratic term, multiplied by $\mathbf{1}_F$, is nonnegative and has conditional expectation at most $M_\rho\mathfrak{m}/(2n)$ by the variance bound above. Finally, label realizations in F^c contribute at most $\mathbb{P}(F^c \mid \boldsymbol{x}^n)$ to the expected utility, since $0 \le \psi(\boldsymbol{C}_n) \le 1$. Splitting the expectation over F and F^c and using $\psi(\bar{\boldsymbol{C}}) \ge 0$, we obtain

$$\begin{aligned}\mathbb{E}[\psi(\boldsymbol{C}_n) \mid \boldsymbol{x}^n] &\le \psi(\bar{\boldsymbol{C}}) + \frac{M_\rho\mathfrak{m}}{2n} + (1 + L_\rho\sqrt{2\mathfrak{m}})\mathbb{P}(F^c \mid \boldsymbol{x}^n) \\ &\le \widehat{\lambda}_n + \frac{M_\rho\mathfrak{m}}{2n} + (1 + L_\rho\sqrt{2\mathfrak{m}})\delta_n(\rho).\end{aligned}$$

The right-hand side is independent of the chosen prediction sequence, so maximizing over $\boldsymbol{z}^n$ gives the same bound for $\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n)$ whenever $\boldsymbol{x}^n \in E$. We can now average over the input sample. On E we use this bound; on E^c we use $\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n) - \lambda^\star \le 1$, which follows from $\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n) \le 1$ and $\lambda^\star \ge 0$. Splitting the expectation over these two events gives

$$\begin{aligned}\mathbb{E}[\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n)] - \lambda^\star \le \mathbb{E}[(\widehat{\lambda}_n - \lambda^\star)\mathbf{1}_E] + \frac{M_\rho\mathfrak{m}}{2n} \\ + (1 + L_\rho\sqrt{2\mathfrak{m}})\delta_n(\rho) + \mathbb{P}(E^c),\end{aligned}$$

To finish, bound the first term by $\mathbb{E}[|\widehat{\lambda}_n - \lambda^\star|\mathbf{1}_E]$ and apply (31). The remaining exceptional-event probability is at most $\delta_n(\rho)$ by Lemma F.6, giving the stated inequality. □

F.4 ETU regret under local denominator bounds

The final argument now has the same form as the global-denominator proof: compare the online utility with $\lambda^\star$, then add the benchmark gap. Only the conversion from the realized surrogate to utility must be restricted to F. The following bound has the same leading terms as Theorem 4.2, with an additional exponentially small term.

Theorem F.8 (ETU regret for label-adaptive SOLO with $\boldsymbol{p}$-dependent denominator bounds)**.** *Fix $\rho \in (0, p^\star_{\min})$ and run the label-adaptive version of SOLO with stepsize $\alpha_t = 1/(\gamma_\rho(t+1))$. Let L_ρ, M_ρ be the local constants from Lemma F.2 and let $\delta_n(\rho)$ be defined in Lemma F.6. Then*

$$R_n^{\mathrm{ETU}} \le \frac{4}{\gamma_\rho^2\sqrt{n}} + \frac{4}{\gamma_\rho}\sqrt{\frac{2\mathfrak{c}}{\gamma_\rho}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)} + \frac{4\mathfrak{c}}{\gamma_\rho}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n) + \frac{M_\rho \mathfrak{m}}{2n} + \left(4 + L_\rho\sqrt{2\mathfrak{m}}\right)\delta_n(\rho).$$

In particular, if $p^\star_{\min}$ is fixed and $\rho < p^\star_{\min}$ is fixed, the extra term is exponentially small in n.

Proof. We first compare the expected utility of the online predictions with the population optimum $\lambda^\star$. We will then use Lemma F.7 to replace this population benchmark with the ETU benchmark. The argument follows the proof of Theorem 4.2, but we must account for samples on which the realized label frequencies are too small to guarantee a positive denominator bound.

Let $\mathcal{F}_{t-1}$ denote the history before round t, including the randomness that determines the current prediction rule, and write

$$\epsilon_t := \|\widehat{\boldsymbol{\eta}}_t(\boldsymbol{x}_t) - \boldsymbol{\eta}(\boldsymbol{x}_t)\|_1, \qquad r_t := \psi_{\lambda^\star}(\boldsymbol{C}(\boldsymbol{y}_t, \widehat{\boldsymbol{y}}_t)), \qquad S_n := \frac{1}{n}\sum_{t=1}^n r_t.$$

The quantity S_n is the surrogate value of the algorithm's final empirical confusion matrix. More precisely, with $\boldsymbol{C}_n := \boldsymbol{C}(\boldsymbol{y}^n, \widehat{\boldsymbol{y}}^n)$, linearity gives

$$S_n = N(\boldsymbol{C}_n) - \lambda^\star D(\boldsymbol{C}_n).$$

To bound its negative part, we use the prediction comparison (27). That comparison requires only $0 \le D \le 1$ and the CPE perturbation bound, so it remains valid under the present assumptions. Taking expectations over the fresh example gives

$$\mathbb{E}[r_t \mid \mathcal{F}_{t-1}] \ge -|\lambda_{t-1} - \lambda^\star| - 4\mathfrak{c}\,\mathbb{E}[\epsilon_t \mid \mathcal{F}_{t-1}].$$

Define the fluctuation around these conditional expectations by

$$M_n := \frac{1}{n}\sum_{t=1}^n \big(r_t - \mathbb{E}[r_t \mid \mathcal{F}_{t-1}]\big).$$

Since $|r_t| \le 1$, the martingale variance calculation from the proof of Theorem 4.2 gives $\mathbb{E}\,|M_n| \le n^{-1/2}$. Summing the preceding conditional lower bounds and using $S_n = n^{-1}\sum_t \mathbb{E}[r_t \mid \mathcal{F}_{t-1}] + M_n$, we obtain

$$(-S_n)_+ \le \frac{1}{n}\sum_{t=1}^n |\lambda_{t-1} - \lambda^\star| + \frac{4\mathfrak{c}}{n}\sum_{t=1}^n \mathbb{E}[\epsilon_t \mid \mathcal{F}_{t-1}] + |M_n|.$$

All these bounds hold without restricting the data to a particular event.

We now convert the surrogate bound into a utility bound. Recall the event

$$F = \left\{\min_k \frac{1}{n}\sum_{t=1}^n y_{tk} \ge \rho\right\}$$

from Lemma F.6. On F, the label frequencies of $\boldsymbol{C}_n$ are at least ρ, regardless of the algorithm's predictions. Consequently, $\boldsymbol{C}_n \in \mathcal{C}_\rho$ and $D(\boldsymbol{C}_n) \ge \gamma_\rho$, giving

$$\lambda^\star - \psi(\boldsymbol{C}_n) = \frac{-S_n}{D(\boldsymbol{C}_n)} \le \frac{(-S_n)_+}{\gamma_\rho} \qquad \text{on } F.$$

For samples in F^c, we cannot use this denominator bound. We can still bound the difference in utilities directly: since $\lambda^\star \le 1$ and $\psi(\boldsymbol{C}_n) \ge 0$, we always have $\lambda^\star - \psi(\boldsymbol{C}_n) \le 1$. Splitting the expectation over F and F^c therefore yields

$$\begin{aligned}\lambda^\star - \mathbb{E}[\psi(\boldsymbol{C}_n)] &= \mathbb{E}[(\lambda^\star - \psi(\boldsymbol{C}_n))\mathbf{1}_F] + \mathbb{E}[(\lambda^\star - \psi(\boldsymbol{C}_n))\mathbf{1}_{F^c}] \\ &\le \frac{1}{\gamma_\rho}\,\mathbb{E}[(-S_n)_+\mathbf{1}_F] + \mathbb{P}(F^c) \\ &\le \frac{1}{\gamma_\rho}\,\mathbb{E}[(-S_n)_+] + \mathbb{P}(F^c).\end{aligned}$$

The last inequality uses $(-S_n)_+ \ge 0$. In particular, we can use the unconditional root and martingale bounds above, without conditioning them on F. Substituting the bound for $(-S_n)_+$, taking expectations, and using the definition of $\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)$ gives

$$\begin{aligned}\lambda^\star - \mathbb{E}[\psi(\boldsymbol{C}_n)] \le \frac{1}{\gamma_\rho n}\sum_{t=1}^{n}\mathbb{E}\,|\lambda_{t-1} - \lambda^\star| + \frac{4\mathfrak{c}}{\gamma_\rho}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n) \\ + \frac{1}{\gamma_\rho\sqrt{n}} + \mathbb{P}(F^c).\end{aligned}$$

The root-tracking guarantee in Lemma F.4 bounds the first term, while Lemma F.6 gives $\mathbb{P}(F^c) \le 2\delta_n(\rho)$. Thus

$$\begin{aligned}\lambda^\star - \mathbb{E}[\psi(\boldsymbol{C}_n)] \le \frac{2}{\gamma_\rho^2\sqrt{n}} + \frac{4}{\gamma_\rho}\sqrt{\frac{2\mathfrak{c}}{\gamma_\rho}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n)} + \frac{4\mathfrak{c}}{\gamma_\rho}\mathrm{Err}(\widehat{\boldsymbol{\eta}}^n) \\ + \frac{1}{\gamma_\rho\sqrt{n}} + 2\delta_n(\rho).\end{aligned}$$

It remains to compare with the ETU benchmark. By definition,

$$R_n^{\mathrm{ETU}} = \big(\mathbb{E}[\psi^\star_{\mathrm{ETU}}(\boldsymbol{x}^n)] - \lambda^\star\big) + \big(\lambda^\star - \mathbb{E}[\psi(\boldsymbol{C}_n)]\big).$$

The first difference is bounded by Lemma F.7, and the second by the preceding display. Adding the bounds gives the claimed result: the terms of order $n^{-1/2}$ satisfy $2/(\gamma_\rho^2\sqrt{n}) + 2/(\gamma_\rho\sqrt{n}) \le 4/(\gamma_\rho^2\sqrt{n})$ because $\gamma_\rho \le 1$, and the terms involving $\delta_n(\rho)$ sum to $(4 + L_\rho\sqrt{2\mathfrak{m}})\delta_n(\rho)$. □

G Experimental Details

Table 4: Dataset split sizes and numbers of labels or classes m.

Dataset	n CPE train	n online test	n PU test	m
YouTube	22,192	4,755	4,756	47
Mediamill	30,734	6,586	6,587	101
Flickr	56,359	12,076	12,078	195
RCV1X	545,866	116,971	116,972	2,456
Caltech-256	20,846	4,467	4,467	256
LEDGAR	56,000	12,000	12,000	100

Data splits and repetitions. We pool the original training and test partitions, randomly split the examples into CPE-training, online, and PU sets in proportions $70\%/15\%/15\%$, and train a separate CPE for each split. The splits are unstratified; the CPE-training and online sizes are rounded down, with the remainder assigned to PU evaluation. Exact sizes and label counts are given in Table 4. All methods share the CPE, partitions, and online ordering within a repetition. We repeat each experiment 10 times and report the mean and standard deviation of each metric.

Multi-label CPE training. For multi-label experiments, we train logistic-regression CPEs with LIBLINEAR [Fan et al., 2008] and L_2 regularization. The cost C is selected from $\{1, 4, 8, 16, 32, 64\}$ by cross-validation on the CPE train set to minimize log loss. All downstream methods use the same fitted CPE within each split.

Multi-class CPE training. Caltech-256 uses VGG16 features, and LEDGAR uses TF-IDF features. For each dataset and split, we fit a multinomial logistic-regression CPE with L_2 regularization and LBFGS. The cost C is selected from $\{1, 4, 8, 16, 32, 64\}$ by cross-validation on the CPE train set. The class weights in Equation (11) are computed from the train-partition class counts, supplied before streaming, and shared by all methods.

Method settings. The online methods predict before observing the current label and freeze their final state for PU prediction. For OMMA, this final-state evaluation is a heuristic; the concave-utility guaranty in Theorem 3.3 uses a randomized online-to-batch classifier. OFO, Greedy, OMMA add a small $\rho = 1/t$ regularization to the confusion-matrix estimates to bound it away from zero. SOLO uses $\lambda_0 = 0.5$, projects root estimates onto $[0, 1]$, and uses $\alpha_t = 1/(t + 1)$, where t counts root updates. For macro metrics, each label has its own counter, updated when the label is true or predicted positive. No parameter tuning is performed on the online or PU sets; all hyperparameters are fixed across datasets and repetitions.

Computational resources. All experiments were conducted on a workstation with 64 GB of RAM and a 24-core CPU and can be reproduced on a single workstation in under 24 hours.

Sensitivity protocol. In Appendix I, we vary one factor at a time over 5 repetitions (not 10, like in the case of the main results) on YouTube, Flickr, Caltech-256, and LEDGAR datasets. References use $\lambda_0 = 0.5$, $\rho = 1$, and $\alpha_t = \alpha_0/(t + 1)^{\alpha_e}$, with $\alpha_e = 1$ and $\alpha_0 = 1$. To study sensitivity to the quality of the CPEs, we additionally retrain CPEs on nested unstratified 100%, 40%, and 20% training subsets, keeping evaluation partitions, orderings, method settings, and weights fixed.

H Additional results

Tables 5 and 6 give the results from Tables 1 and 2 with standard deviations. Figures 4 to 6 show the incremental online performance for all multi-label and multi-class datasets and metrics.

Table 5: Results for multi-label online and PU test sets (%), reported as mean ± sample standard deviation over 10 runs. The best mean in each column is in **bold underlined**, the second best in ***bold italic***. Ma and Mi denote macro and micro averaging; Ja. denotes the Jaccard index. ⋆ marks a batch baseline; × denotes an unsupported metric for that baseline.

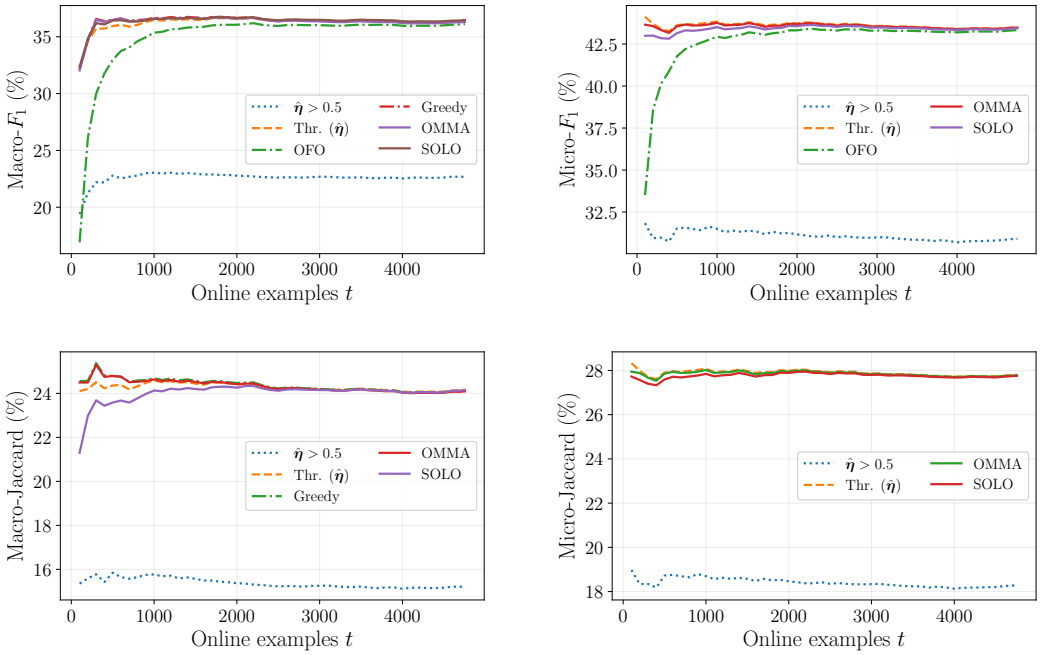
Method	Online				PU			
	Ma-F_1	Mi-F_1	Ma-Ja.	Mi-Ja.	Ma-F_1	Mi-F_1	Ma-Ja.	Mi-Ja.
	YouTube							
$\hat{\eta}>0.5$	22.79 ± 0.55	30.97 ± 0.47	15.27 ± 0.45	18.32 ± 0.33	22.90 ± 0.24	30.86 ± 0.84	15.42 ± 0.24	18.25 ± 0.59
OFO	36.10 ± 0.50	43.28 ± 0.62	×	×	36.35 ± 0.26	***43.29*** ± 0.14	×	×
Greedy	36.38 ± 0.37	×	24.14 ± 0.35	×	36.37 ± 0.26	×	24.20 ± 0.22	×
OMMA	36.34 ± 0.39	**43.44** ± 0.58	24.12 ± 0.37	**27.75** ± 0.47	***36.38*** ± 0.24	43.28 ± 0.15	***24.21*** ± 0.21	***27.62*** ± 0.12
SOLO	**36.56** ± 0.58	***43.42*** ± 0.58	**24.27** ± 0.45	***27.73*** ± 0.47	**36.54** ± 0.24	**43.29** ± 0.16	**24.31** ± 0.18	**27.63** ± 0.13
Thr. $(\boldsymbol{y})^\star$	×	×	×	×	36.01 ± 0.34	43.26 ± 0.07	23.94 ± 0.25	27.60 ± 0.06
Thr. $(\hat{\boldsymbol{\eta}})^\star$	***36.45*** ± 0.41	43.42 ± 0.60	***24.19*** ± 0.37	27.73 ± 0.49	×	×	×	×
	Mediamill							
$\hat{\eta}>0.5$	4.32 ± 0.06	53.64 ± 0.31	3.18 ± 0.03	36.65 ± 0.29	4.35 ± 0.05	53.65 ± 0.22	3.20 ± 0.03	36.66 ± 0.21
OFO	13.29 ± 0.16	58.01 ± 0.33	×	×	13.77 ± 0.28	57.83 ± 0.35	×	×
Greedy	13.48 ± 0.14	×	8.63 ± 0.07	×	13.90 ± 0.30	×	8.88 ± 0.19	×
OMMA	13.48 ± 0.15	**58.11** ± 0.30	8.64 ± 0.08	**40.95** ± 0.30	13.87 ± 0.30	57.84 ± 0.34	8.86 ± 0.18	40.69 ± 0.34
SOLO	***14.02*** ± 0.20	58.07 ± 0.31	***8.92*** ± 0.11	40.92 ± 0.31	***14.54*** ± 0.44	***57.84*** ± 0.34	***9.22*** ± 0.28	***40.69*** ± 0.33
Thr. $(\boldsymbol{y})^\star$	×	×	×	×	**16.16** ± 0.35	**57.85** ± 0.36	**10.14** ± 0.23	**40.70** ± 0.35
Thr. $(\hat{\boldsymbol{\eta}})^\star$	**16.31** ± 0.37	***58.10*** ± 0.34	**10.24** ± 0.23	***40.95*** ± 0.33	×	×	×	×
	Flickr							
$\hat{\eta}>0.5$	18.86 ± 0.75	29.54 ± 0.40	12.13 ± 0.56	17.33 ± 0.28	18.66 ± 0.28	29.75 ± 0.37	12.03 ± 0.16	17.48 ± 0.26
OFO	30.31 ± 0.53	40.65 ± 0.40	×	×	***30.68*** ± 0.26	***41.22*** ± 0.27	×	×
Greedy	30.93 ± 0.34	×	19.96 ± 0.31	×	30.67 ± 0.31	×	***19.83*** ± 0.22	×
OMMA	***30.94*** ± 0.36	40.91 ± 0.42	***19.97*** ± 0.33	25.71 ± 0.33	30.66 ± 0.32	41.15 ± 0.30	19.83 ± 0.23	25.91 ± 0.24
SOLO	30.90 ± 0.43	***40.94*** ± 0.43	19.96 ± 0.37	***25.74*** ± 0.34	**30.73** ± 0.34	**41.23** ± 0.31	**19.86** ± 0.26	**25.97** ± 0.25
Thr. $(\boldsymbol{y})^\star$	×	×	×	×	29.95 ± 0.39	41.20 ± 0.26	19.31 ± 0.27	***25.94*** ± 0.20
Thr. $(\hat{\boldsymbol{\eta}})^\star$	**31.14** ± 0.50	**40.95** ± 0.36	**20.15** ± 0.44	**25.75** ± 0.29	×	×	×	×
	RCV1X							
$\hat{\eta}>0.5$	15.67 ± 0.24	69.79 ± 0.20	10.70 ± 0.15	53.60 ± 0.23	15.43 ± 0.22	69.75 ± 0.12	10.53 ± 0.14	53.55 ± 0.14
OFO	22.21 ± 0.26	70.75 ± 0.20	×	×	23.87 ± 0.43	***70.72*** ± 0.08	×	×
Greedy	24.86 ± 0.49	×	16.69 ± 0.33	×	25.03 ± 0.54	×	***16.88*** ± 0.39	×
OMMA	24.74 ± 0.43	70.73 ± 0.18	16.61 ± 0.29	54.72 ± 0.22	24.99 ± 0.55	70.67 ± 0.07	16.85 ± 0.40	54.64 ± 0.09
SOLO	**25.77** ± 0.34	**70.79** ± 0.19	***17.21*** ± 0.23	**54.79** ± 0.23	**25.41** ± 0.59	**70.72** ± 0.08	**17.07** ± 0.43	**54.70** ± 0.10
Thr. $(\boldsymbol{y})^\star$	×	×	×	×	***25.10*** ± 0.47	70.72 ± 0.08	16.77 ± 0.30	***54.70*** ± 0.10
Thr. $(\hat{\boldsymbol{\eta}})^\star$	***25.66*** ± 0.23	***70.76*** ± 0.18	**17.27** ± 0.14	***54.75*** ± 0.22	×	×	×	×

Table 6: Results for multi-class online and PU test sets, measured by CW-$F_{0.25}$ (%) and reported as mean ± sample standard deviation over 10 runs. The best mean in each column is in **bold underlined**, the second best in ***bold italic***. ⋆ marks a batch baseline.

	Caltech-256		LEDGAR	
Method	Online	PU	Online	PU
Top-1	78.34 ± 0.75	80.47 ± 0.87	77.72 ± 4.61	79.59 ± 3.01
Greedy	89.49 ± 0.94	91.26 ± 0.64	***86.53*** ± 1.36	***87.79*** ± 1.60
OMMA	89.49 ± 0.94	91.26 ± 0.64	***86.53*** ± 1.36	**87.80** ± 1.60
SOLO	***91.47*** ± 0.68	***92.00*** ± 0.53	86.32 ± 1.39	87.11 ± 1.73
Batch plug-in⋆	**92.27** ± 0.54	**92.34** ± 0.47	**87.48** ± 1.23	87.63 ± 1.42

Figure 4: Incremental online performance (%) on YouTube and Mediamill, averaged over 10 runs. The first point is at $t = 100$.

YouTube



Mediamill

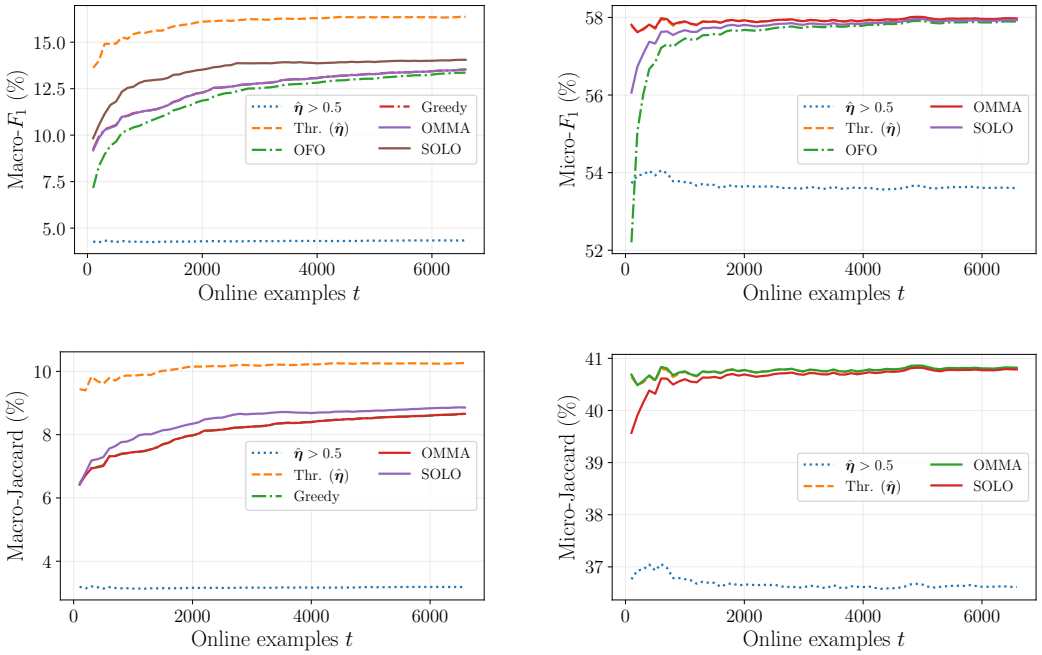
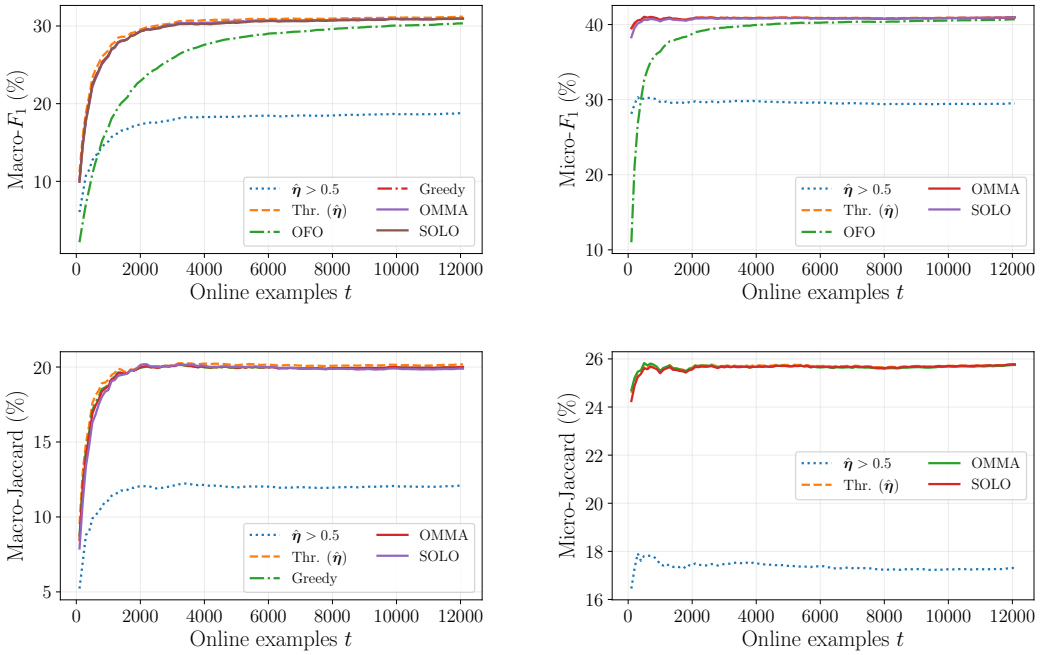
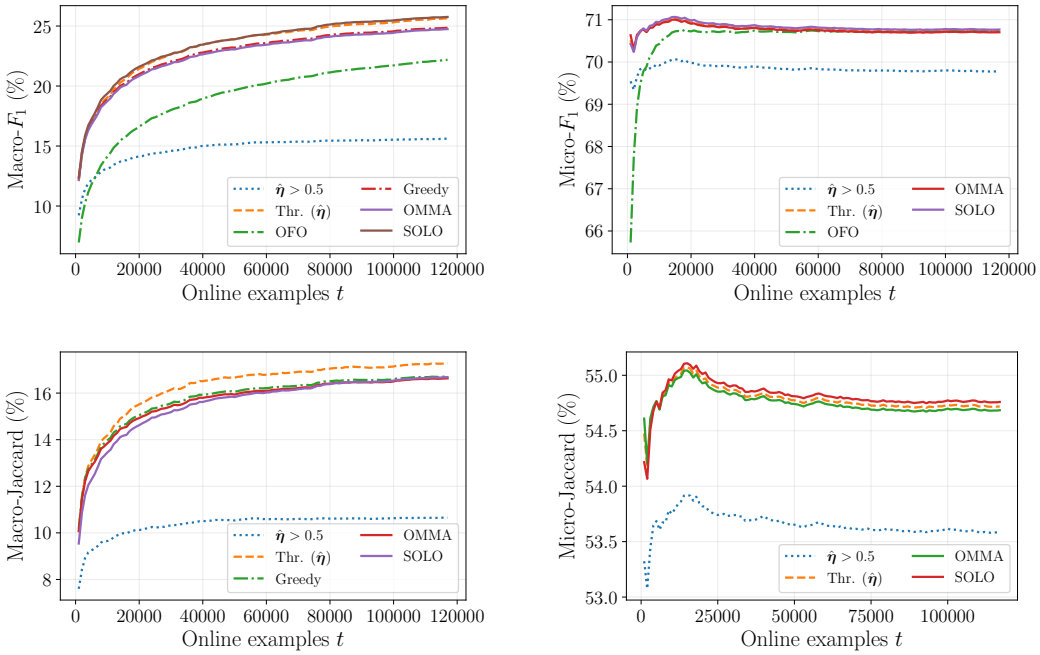


Figure 5: Incremental online performance (%) on Flickr and RCV1X, averaged over 10 runs. The first point is at $t = 100$.

FLICKR



RCV1X



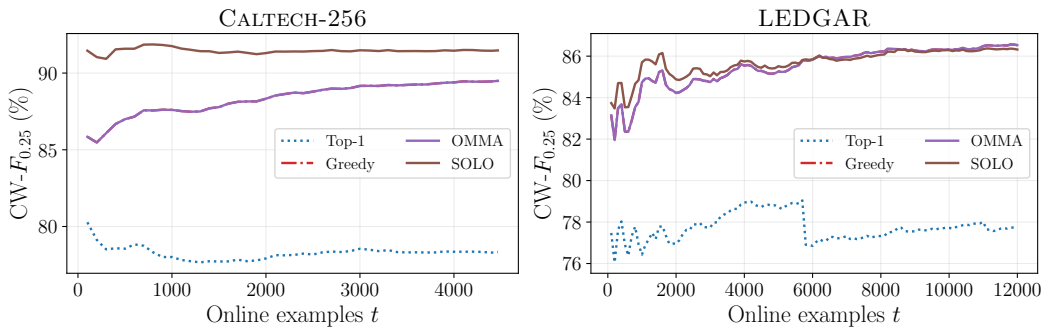


Figure 6: Incremental online CW-$F_{0.25}$ (%) on Caltech-256 and LEDGAR, averaged over 10 runs. The first point is at $t = 100$.

I Sensitivity experiments

We vary the confusion-matrix regularization, the step-size schedule, the initial root, and the CPE-training fraction, holding the other settings fixed. The protocol and reference configurations are given in Appendix G. Results are averaged over 5 matched repetitions on YouTube, Flickr, Caltech-256, and the LEDGAR datasets.

I.1 Confusion-matrix stabilization

Tables 7 and 8 report results for regularization values ρ ranging from $10^{-9}/t$ to $10/t$. The regularization magnitude has a substantial effect on OFO, Greedy, and OMMA. YouTube and Flickr contain rare labels for which the corresponding binary classification problems are highly imbalanced, and moderate regularization prevents nearly empty early confusion matrices from producing unstable operating points on macro metrics. Micro metrics and multi-class experiments are considerably less sensitive because the corresponding confusion statistics are pooled across labels. By contrast, SOLO is essentially invariant over the entire tested range: its results change by at most 0.01 percentage points. This robustness is consistent with the structural difference between the methods. OFO, Greedy, and OMMA use confusion-matrix statistics directly as their decision state, whereas SOLO maintains only the scalar root estimate.

I.2 Step-size schedule and root initialization

Tables 9 and 10 report results obtained by varying the coefficient α_0 and exponent α_e in $\alpha_t = \alpha_0/(t+1)^{\alpha_e}$. We also vary $\lambda_0 \in \{0, 0.25, 0.5, 0.75, 1\}$. Each sweep changes one parameter from the reference configuration. The results mostly show that the reference schedule and root selection are competitive across metrics, although nearby choices can improve individual scores. However, unsurprisingly, more suitable schedules and initial roots can be found depending on the dataset.

I.3 CPE-training fraction

To study sensitivity to the quality of the CPE, we retrained the same logistic-regression CPE using 100%, 40%, and 20% of each CPE-training partition. All downstream methods and their hyperparameters were kept fixed. Table 11 reports log loss and Brier score for the CPE probabilities for each partition, and Table 12 reports the resulting utilities. As the CPE training fraction decreases, both log loss and Brier score increase monotonically, and the achievable utilities decrease for all methods. This confirms that CPE quality is a shared upstream performance bottleneck rather than an issue specific to OMMA or SOLO. Importantly, the relative gap between SOLO and the corresponding threshold reference remains small across all levels of CPE quality. Weakening the CPE shifts the attainable operating point for every method, but it does not cause a disproportionate additional loss for SOLO. This behavior is consistent with our theoretical decomposition, in which CPE estimation error and the additional metric-optimization error appear as separate terms.

Table 7: Effect of confusion-matrix stabilization on YouTube and Flickr: online and PU utilities (%), reported as mean ± sample standard deviation over 5 runs. The best mean is in **bold underlined**, the second best in ***bold italic*** within each method, dataset, and column. Ma and Mi denote macro and micro averaging; Ja. denotes the Jaccard index. × denotes an unsupported metric.

Method	ρ	Online				PU			
		Ma-F_1	Mi-F_1	Ma-Ja.	Mi-Ja.	Ma-F_1	Mi-F_1	Ma-Ja.	Mi-Ja.
		YouTube							
OFO	$10^{-9}/t$	32.29 ± 0.95	43.16 ± 0.58	×	×	35.59 ± 0.42	43.28 ± 0.15	×	×
	$10^{-6}/t$	32.29 ± 0.95	43.15 ± 0.58	×	×	35.59 ± 0.42	43.28 ± 0.15	×	×
	$0.001/t$	32.47 ± 0.93	***43.44*** ± 0.58	×	×	35.61 ± 0.42	***43.30*** ± 0.16	×	×
	$0.1/t$	***35.29*** ± 0.64	**43.45** ± 0.62	×	×	36.14 ± 0.37	**43.30** ± 0.15	×	×
	$1/t$	**36.17** ± 0.51	43.35 ± 0.56	×	×	**36.37** ± 0.24	43.30 ± 0.14	×	×
	$10/t$	33.63 ± 0.53	41.63 ± 0.40	×	×	***36.35*** ± 0.17	43.19 ± 0.21	×	×
Greedy	$10^{-9}/t$	31.85 ± 0.80	×	20.46 ± 0.69	×	35.72 ± 0.34	×	23.72 ± 0.18	×
	$10^{-6}/t$	31.90 ± 0.82	×	20.50 ± 0.70	×	35.72 ± 0.35	×	23.72 ± 0.19	×
	$0.001/t$	32.66 ± 0.90	×	21.18 ± 0.73	×	35.78 ± 0.47	×	23.79 ± 0.29	×
	$0.1/t$	35.47 ± 0.68	×	23.51 ± 0.49	×	36.13 ± 0.27	×	24.05 ± 0.19	×
	$1/t$	**36.29** ± 0.33	×	**24.08** ± 0.30	×	***36.35*** ± 0.28	×	***24.19*** ± 0.24	×
	$10/t$	***36.18*** ± 0.39	×	***24.04*** ± 0.34	×	**36.48** ± 0.33	×	**24.27** ± 0.28	×
OMMA	$10^{-9}/t$	32.89 ± 0.87	43.02 ± 0.39	21.39 ± 0.74	27.41 ± 0.32	35.83 ± 0.46	***43.31*** ± 0.12	23.83 ± 0.28	***27.64*** ± 0.10
	$10^{-6}/t$	33.21 ± 1.16	43.21 ± 0.50	21.71 ± 0.95	27.56 ± 0.40	35.84 ± 0.38	43.30 ± 0.15	23.83 ± 0.22	27.63 ± 0.12
	$0.001/t$	33.31 ± 1.18	43.45 ± 0.60	21.79 ± 0.97	27.75 ± 0.49	35.84 ± 0.37	43.30 ± 0.16	23.83 ± 0.21	27.63 ± 0.13
	$0.1/t$	35.51 ± 0.67	***43.46*** ± 0.62	23.53 ± 0.51	***27.76*** ± 0.51	36.16 ± 0.26	43.30 ± 0.14	24.06 ± 0.20	27.63 ± 0.12
	$1/t$	**36.33** ± 0.37	**43.46** ± 0.62	**24.11** ± 0.33	**27.76** ± 0.51	***36.34*** ± 0.26	43.29 ± 0.16	***24.18*** ± 0.21	27.63 ± 0.13
	$10/t$	***36.19*** ± 0.38	43.42 ± 0.53	***24.05*** ± 0.33	27.73 ± 0.43	**36.48** ± 0.33	**43.32** ± 0.12	**24.27** ± 0.28	**27.65** ± 0.09
SOLO	$10^{-9}/t$	***36.49*** ± 0.41	**43.43** ± 0.54	24.20 ± 0.31	**27.76** ± 0.48	***36.42*** ± 0.31	**43.27** ± 0.16	***24.24*** ± 0.30	**27.62** ± 0.13
	$10^{-6}/t$	***36.49*** ± 0.41	**43.43** ± 0.54	24.20 ± 0.31	**27.76** ± 0.48	***36.42*** ± 0.31	**43.27** ± 0.16	***24.24*** ± 0.30	**27.62** ± 0.13
	$0.001/t$	**36.50** ± 0.40	**43.43** ± 0.54	24.20 ± 0.31	**27.76** ± 0.48	**36.42** ± 0.30	**43.27** ± 0.16	24.24 ± 0.30	**27.62** ± 0.13
	$0.1/t$	36.48 ± 0.41	**43.43** ± 0.54	***24.20*** ± 0.31	**27.76** ± 0.48	36.42 ± 0.30	**43.27** ± 0.16	**24.24** ± 0.29	**27.62** ± 0.13
	$1/t$	36.45 ± 0.36	**43.43** ± 0.54	24.19 ± 0.30	**27.76** ± 0.48	36.39 ± 0.30	**43.27** ± 0.16	24.23 ± 0.30	**27.62** ± 0.13
	$10/t$	36.44 ± 0.47	**43.43** ± 0.54	**24.23** ± 0.25	**27.76** ± 0.48	36.36 ± 0.33	**43.27** ± 0.16	24.19 ± 0.29	**27.62** ± 0.13
		Flickr							
OFO	$10^{-9}/t$	18.85 ± 0.62	40.68 ± 0.40	×	×	26.62 ± 0.32	41.21 ± 0.32	×	×
	$10^{-6}/t$	18.85 ± 0.62	40.68 ± 0.40	×	×	26.62 ± 0.32	41.21 ± 0.32	×	×
	$0.001/t$	19.07 ± 0.67	**40.93** ± 0.42	×	×	26.63 ± 0.36	41.22 ± 0.31	×	×
	$0.1/t$	***28.37*** ± 0.62	***40.90*** ± 0.42	×	×	***29.53*** ± 0.44	**41.23** ± 0.30	×	×
	$1/t$	**30.22** ± 0.64	40.55 ± 0.40	×	×	**30.65** ± 0.23	***41.22*** ± 0.29	×	×
	$10/t$	22.54 ± 0.94	37.01 ± 0.41	×	×	27.74 ± 0.21	40.84 ± 0.32	×	×
Greedy	$10^{-9}/t$	19.27 ± 0.30	×	11.83 ± 0.22	×	26.94 ± 0.19	×	17.24 ± 0.17	×
	$10^{-6}/t$	19.40 ± 0.31	×	11.93 ± 0.22	×	26.97 ± 0.20	×	17.26 ± 0.18	×
	$0.001/t$	21.07 ± 0.81	×	13.16 ± 0.53	×	27.41 ± 0.33	×	17.58 ± 0.24	×
	$0.1/t$	29.03 ± 0.53	×	18.73 ± 0.43	×	29.69 ± 0.25	×	19.19 ± 0.18	×
	$1/t$	**30.86** ± 0.44	×	**19.92** ± 0.38	×	**30.69** ± 0.27	×	**19.85** ± 0.21	×
	$10/t$	***29.83*** ± 0.54	×	***19.37*** ± 0.43	×	***30.36*** ± 0.25	×	***19.71*** ± 0.20	×
OMMA	$10^{-9}/t$	20.64 ± 0.44	40.59 ± 0.61	12.88 ± 0.33	25.46 ± 0.48	27.32 ± 0.20	41.17 ± 0.30	17.51 ± 0.17	25.92 ± 0.24
	$10^{-6}/t$	21.92 ± 0.85	40.59 ± 0.61	13.90 ± 0.66	25.46 ± 0.48	27.40 ± 0.38	41.17 ± 0.30	17.63 ± 0.27	25.92 ± 0.24
	$0.001/t$	22.10 ± 0.90	40.85 ± 0.42	14.03 ± 0.69	25.67 ± 0.33	27.40 ± 0.41	***41.18*** ± 0.31	17.63 ± 0.29	***25.93*** ± 0.24
	$0.1/t$	29.02 ± 0.47	**40.90** ± 0.44	18.73 ± 0.40	**25.71** ± 0.35	29.70 ± 0.29	41.17 ± 0.30	19.20 ± 0.21	25.92 ± 0.24
	$1/t$	**30.89** ± 0.45	***40.88*** ± 0.42	**19.94** ± 0.39	***25.69*** ± 0.33	**30.69** ± 0.26	41.16 ± 0.30	**19.85** ± 0.19	25.91 ± 0.24
	$10/t$	***29.82*** ± 0.52	40.86 ± 0.42	***19.37*** ± 0.41	25.67 ± 0.33	***30.36*** ± 0.26	**41.18** ± 0.26	***19.71*** ± 0.21	**25.93** ± 0.21
SOLO	$10^{-9}/t$	30.86 ± 0.42	**40.92** ± 0.42	19.60 ± 0.51	**25.74** ± 0.33	30.71 ± 0.20	**41.21** ± 0.32	19.79 ± 0.19	**25.96** ± 0.25
	$10^{-6}/t$	30.86 ± 0.42	**40.92** ± 0.42	19.60 ± 0.51	**25.74** ± 0.33	30.71 ± 0.20	**41.21** ± 0.32	19.79 ± 0.19	**25.96** ± 0.25
	$0.001/t$	30.86 ± 0.42	**40.92** ± 0.42	19.60 ± 0.51	**25.74** ± 0.33	**30.71** ± 0.20	**41.21** ± 0.32	19.79 ± 0.19	**25.96** ± 0.25
	$0.1/t$	30.86 ± 0.42	**40.92** ± 0.42	19.59 ± 0.51	**25.74** ± 0.33	30.71 ± 0.21	**41.21** ± 0.32	19.80 ± 0.18	**25.96** ± 0.25
	$1/t$	***30.87*** ± 0.41	**40.92** ± 0.42	***19.60*** ± 0.51	**25.74** ± 0.33	30.70 ± 0.21	**41.21** ± 0.32	***19.80*** ± 0.16	**25.96** ± 0.25
	$10/t$	**30.87** ± 0.46	**40.92** ± 0.42	**19.63** ± 0.52	**25.74** ± 0.33	***30.71*** ± 0.24	**41.21** ± 0.32	**19.83** ± 0.17	**25.96** ± 0.25

Table 8: Effect of confusion-matrix stabilization on Caltech-256 and LEDGAR: online and PU CW-$F_{0.25}$ (%), reported as mean $\pm$ sample standard deviation over 5 runs. The best mean is in **bold underlined**, the second best in ***bold italic*** within each method and column. † marks the unsmoothed OMMA diagnostic with an undefined initial gradient.

		CALTECH-256		LEDGAR	
Method	ρ	Online	PU	Online	PU
Greedy	$10^{-9}/t$	**91.52** $\pm$ 0.79	***92.08*** $\pm$ 0.65	***87.51*** $\pm$ 1.23	88.07 $\pm$ 1.58
	$10^{-6}/t$	**91.52** $\pm$ 0.79	***92.08*** $\pm$ 0.65	***87.51*** $\pm$ 1.23	88.07 $\pm$ 1.58
	$0.001/t$	***91.51*** $\pm$ 0.78	**92.08** $\pm$ 0.65	**87.52** $\pm$ 1.23	88.07 $\pm$ 1.58
	$0.1/t$	91.24 $\pm$ 0.81	92.06 $\pm$ 0.70	87.36 $\pm$ 1.27	***88.07*** $\pm$ 1.68
	$1/t$	89.68 $\pm$ 1.13	91.45 $\pm$ 0.79	87.05 $\pm$ 0.91	**88.27** $\pm$ 1.47
	$10/t$	86.40 $\pm$ 1.21	88.69 $\pm$ 1.27	84.57 $\pm$ 1.65	85.45 $\pm$ 2.92
OMMA	$10^{-9}/t$	***91.51*** $\pm$ 0.79	***92.08*** $\pm$ 0.65	86.79 $\pm$ 2.28	88.05 $\pm$ 1.59
	$10^{-6}/t$	***91.51*** $\pm$ 0.79	***92.08*** $\pm$ 0.65	**87.37** $\pm$ 1.10	88.06 $\pm$ 1.60
	$0.001/t$	**91.52** $\pm$ 0.79	***92.08*** $\pm$ 0.65	87.35 $\pm$ 1.06	88.05 $\pm$ 1.61
	$0.1/t$	91.24 $\pm$ 0.81	92.07 $\pm$ 0.70	***87.36*** $\pm$ 1.27	88.07 $\pm$ 1.68
	$1/t$	89.68 $\pm$ 1.13	91.45 $\pm$ 0.79	87.05 $\pm$ 0.91	***88.27*** $\pm$ 1.47
	$10/t$	86.40 $\pm$ 1.21	88.69 $\pm$ 1.27	84.57 $\pm$ 1.65	85.45 $\pm$ 2.92
SOLO	$10^{-9}/t$	**91.44** $\pm$ 0.82	**92.09** $\pm$ 0.65	**86.79** $\pm$ 0.62	**87.08** $\pm$ 1.50
	$10^{-6}/t$	**91.44** $\pm$ 0.82	**92.09** $\pm$ 0.65	**86.79** $\pm$ 0.62	**87.08** $\pm$ 1.50
	$0.001/t$	**91.44** $\pm$ 0.82	**92.09** $\pm$ 0.65	**86.79** $\pm$ 0.62	**87.08** $\pm$ 1.50
	$0.1/t$	**91.44** $\pm$ 0.82	**92.09** $\pm$ 0.65	**86.79** $\pm$ 0.62	**87.08** $\pm$ 1.50
	$1/t$	**91.44** $\pm$ 0.82	**92.09** $\pm$ 0.65	**86.79** $\pm$ 0.62	**87.08** $\pm$ 1.50
	$10/t$	**91.44** $\pm$ 0.82	**92.09** $\pm$ 0.65	**86.79** $\pm$ 0.62	**87.08** $\pm$ 1.50

Table 9: Effect of SOLO step size and root initialization on YouTube and Flickr: online and PU utilities (%), reported as mean ± sample standard deviation over 5 runs. The best mean is in **bold underlined**, the second best in ***bold italic*** within each parameter block, dataset, and column. Ma and Mi denote macro and micro averaging; Ja. denotes the Jaccard index.

Parameter	Online				PU			
	Ma-F_1	Mi-F_1	Ma-Ja.	Mi-Ja.	Ma-F_1	Mi-F_1	Ma-Ja.	Mi-Ja.
	YouTube							
$\alpha_0 = 0.0005$	33.77 ± 0.39	43.32 ± 0.47	20.74 ± 0.45	25.89 ± 0.33	33.60 ± 0.32	43.14 ± 0.23	20.70 ± 0.26	25.77 ± 0.39
$\alpha_0 = 0.005$	33.86 ± 0.45	43.38 ± 0.54	20.80 ± 0.48	26.11 ± 0.36	33.66 ± 0.34	43.11 ± 0.18	20.76 ± 0.26	26.06 ± 0.38
$\alpha_0 = 0.05$	34.55 ± 0.53	***43.46*** ± 0.55	21.46 ± 0.45	27.44 ± 0.42	34.50 ± 0.24	43.28 ± 0.18	21.51 ± 0.36	27.40 ± 0.28
$\alpha_0 = 0.25$	***36.25*** ± 0.25	**43.47** ± 0.63	23.41 ± 0.29	**27.78** ± 0.53	***36.38*** ± 0.15	**43.30** ± 0.15	23.77 ± 0.09	**27.63** ± 0.13
$\alpha_0 = 0.5$	35.93 ± 0.50	43.35 ± 0.56	***24.00*** ± 0.34	27.61 ± 0.50	36.24 ± 0.24	***43.28*** ± 0.14	24.12 ± 0.22	27.62 ± 0.13
$\alpha_0 = 1$	**36.45** ± 0.36	43.43 ± 0.54	**24.19** ± 0.30	***27.76*** ± 0.48	**36.39** ± 0.30	43.27 ± 0.16	**24.23** ± 0.30	***27.62*** ± 0.13
$\alpha_0 = 5$	32.00 ± 0.88	42.75 ± 0.43	20.84 ± 0.74	27.20 ± 0.28	36.24 ± 0.29	43.27 ± 0.08	***24.13*** ± 0.13	27.61 ± 0.07
$\alpha_0 = 50$	23.41 ± 0.66	37.80 ± 0.65	14.68 ± 0.45	23.36 ± 0.49	34.52 ± 0.43	42.95 ± 0.34	23.06 ± 0.27	27.47 ± 0.18
$\alpha_0 = 500$	10.79 ± 0.23	17.88 ± 0.40	6.26 ± 0.14	9.96 ± 0.25	29.17 ± 2.61	41.15 ± 4.30	20.18 ± 1.52	25.33 ± 4.72
$\alpha_0 = 5 \times 10^8$	7.57 ± 0.16	7.61 ± 0.10	4.05 ± 0.09	3.95 ± 0.05	16.43 ± 1.56	20.98 ± 13.27	10.76 ± 1.17	12.20 ± 8.10
$\alpha_e = 0.5$	35.21 ± 0.47	43.05 ± 0.33	23.59 ± 0.29	27.40 ± 0.29	35.95 ± 0.10	43.12 ± 0.19	23.98 ± 0.09	27.51 ± 0.09
$\alpha_e = 0.75$	***36.11*** ± 0.42	43.34 ± 0.55	***24.07*** ± 0.38	27.64 ± 0.47	***36.35*** ± 0.17	***43.26*** ± 0.09	***24.15*** ± 0.15	***27.61*** ± 0.08
$\alpha_e = 1$	**36.45** ± 0.36	**43.43** ± 0.54	**24.19** ± 0.30	**27.76** ± 0.48	**36.39** ± 0.30	**43.27** ± 0.16	**24.23** ± 0.30	**27.62** ± 0.13
$\alpha_e = 1.5$	35.76 ± 0.23	***43.40*** ± 0.64	22.89 ± 0.30	***27.71*** ± 0.51	35.64 ± 0.28	43.18 ± 0.18	22.97 ± 0.36	27.60 ± 0.15
$\alpha_e = 2$	35.05 ± 0.47	43.16 ± 0.59	21.96 ± 0.36	27.58 ± 0.49	34.77 ± 0.28	42.96 ± 0.28	21.87 ± 0.34	27.34 ± 0.26
$\lambda_0 = 0$	30.42 ± 0.80	43.35 ± 0.58	19.28 ± 0.69	27.67 ± 0.48	34.40 ± 0.22	43.27 ± 0.16	22.55 ± 0.16	***27.62*** ± 0.13
$\lambda_0 = 0.25$	36.03 ± 0.65	43.38 ± 0.55	***24.15*** ± 0.34	27.74 ± 0.45	36.22 ± 0.36	43.27 ± 0.16	24.21 ± 0.27	27.62 ± 0.13
$\lambda_0 = 0.5$	***36.45*** ± 0.36	43.43 ± 0.54	**24.19** ± 0.30	27.76 ± 0.48	36.39 ± 0.30	***43.27*** ± 0.16	***24.23*** ± 0.30	27.62 ± 0.13
$\lambda_0 = 0.75$	**36.48** ± 0.38	***43.45*** ± 0.58	23.94 ± 0.31	**27.76** ± 0.49	***36.44*** ± 0.33	**43.28** ± 0.16	**24.26** ± 0.24	***27.62*** ± 0.13
$\lambda_0 = 1$	36.42 ± 0.45	**43.46** ± 0.58	23.62 ± 0.39	***27.76*** ± 0.49	**36.49** ± 0.36	***43.27*** ± 0.16	24.14 ± 0.13	**27.62** ± 0.13
	Flickr							
$\alpha_0 = 0.0005$	28.25 ± 0.58	40.45 ± 0.37	16.63 ± 0.63	23.49 ± 0.28	28.14 ± 0.32	40.71 ± 0.26	16.52 ± 0.25	23.53 ± 0.21
$\alpha_0 = 0.005$	28.30 ± 0.60	40.49 ± 0.40	16.68 ± 0.64	23.69 ± 0.28	28.21 ± 0.33	40.76 ± 0.24	16.57 ± 0.27	23.76 ± 0.22
$\alpha_0 = 0.05$	28.87 ± 0.57	40.80 ± 0.44	17.14 ± 0.67	24.86 ± 0.31	28.93 ± 0.34	41.04 ± 0.25	17.18 ± 0.25	25.21 ± 0.19
$\alpha_0 = 0.25$	***30.29*** ± 0.54	**40.95** ± 0.39	18.67 ± 0.59	**25.75** ± 0.31	***30.43*** ± 0.24	**41.22** ± 0.31	18.94 ± 0.22	25.95 ± 0.22
$\alpha_0 = 0.5$	30.04 ± 0.53	***40.93*** ± 0.38	**19.71** ± 0.42	25.74 ± 0.34	30.10 ± 0.29	41.20 ± 0.31	***19.68*** ± 0.12	***25.95*** ± 0.24
$\alpha_0 = 1$	**30.87** ± 0.41	40.92 ± 0.42	***19.60*** ± 0.51	***25.74*** ± 0.33	**30.70** ± 0.21	***41.21*** ± 0.32	**19.80** ± 0.16	**25.96** ± 0.25
$\alpha_0 = 5$	19.05 ± 0.81	40.52 ± 0.47	11.86 ± 0.47	25.39 ± 0.46	29.78 ± 0.17	41.18 ± 0.30	19.22 ± 0.20	25.94 ± 0.23
$\alpha_0 = 50$	12.05 ± 0.16	35.61 ± 0.49	7.32 ± 0.06	21.66 ± 0.51	29.07 ± 0.45	41.20 ± 0.36	18.83 ± 0.26	25.94 ± 0.27
$\alpha_0 = 500$	5.15 ± 0.09	17.07 ± 0.42	2.92 ± 0.06	9.41 ± 0.22	22.86 ± 0.78	41.00 ± 0.36	15.13 ± 0.53	25.82 ± 0.26
$\alpha_0 = 5 \times 10^8$	3.53 ± 0.07	3.11 ± 0.03	1.88 ± 0.04	1.58 ± 0.02	11.69 ± 0.36	13.46 ± 14.89	7.44 ± 0.23	7.78 ± 8.86
$\alpha_e = 0.5$	28.80 ± 0.72	40.84 ± 0.40	19.05 ± 0.46	25.66 ± 0.31	30.08 ± 0.16	***41.21*** ± 0.35	19.53 ± 0.16	***25.94*** ± 0.27
$\alpha_e = 0.75$	***30.67*** ± 0.49	**40.96** ± 0.38	**19.88** ± 0.44	**25.76** ± 0.31	***30.55*** ± 0.23	41.18 ± 0.30	**19.80** ± 0.16	25.94 ± 0.23
$\alpha_e = 1$	**30.87** ± 0.41	***40.92*** ± 0.42	***19.60*** ± 0.51	***25.74*** ± 0.33	**30.70** ± 0.21	**41.21** ± 0.32	***19.80*** ± 0.16	**25.96** ± 0.25
$\alpha_e = 1.5$	30.05 ± 0.52	40.64 ± 0.53	18.47 ± 0.62	25.19 ± 0.31	30.01 ± 0.21	40.87 ± 0.28	18.56 ± 0.25	25.42 ± 0.21
$\alpha_e = 2$	29.28 ± 0.55	40.28 ± 0.68	17.72 ± 0.67	24.41 ± 0.31	29.25 ± 0.31	40.51 ± 0.54	17.65 ± 0.32	24.57 ± 0.51
$\lambda_0 = 0$	17.92 ± 0.27	40.80 ± 0.41	10.96 ± 0.15	25.65 ± 0.33	24.01 ± 0.35	41.21 ± 0.32	15.11 ± 0.26	25.96 ± 0.24
$\lambda_0 = 0.25$	***30.72*** ± 0.51	40.92 ± 0.41	**19.97** ± 0.44	**25.75** ± 0.33	30.43 ± 0.10	41.21 ± 0.32	**19.83** ± 0.14	25.96 ± 0.25
$\lambda_0 = 0.5$	**30.87** ± 0.41	***40.92*** ± 0.42	***19.60*** ± 0.51	25.74 ± 0.33	**30.70** ± 0.21	41.21 ± 0.32	***19.80*** ± 0.16	**25.96** ± 0.25
$\lambda_0 = 0.75$	30.45 ± 0.58	**40.92** ± 0.42	18.95 ± 0.57	25.74 ± 0.33	***30.63*** ± 0.27	**41.21** ± 0.32	19.51 ± 0.14	25.96 ± 0.24
$\lambda_0 = 1$	29.71 ± 0.59	40.92 ± 0.42	18.27 ± 0.63	***25.74*** ± 0.33	30.44 ± 0.18	***41.21*** ± 0.32	19.12 ± 0.31	***25.96*** ± 0.24

Table 10: Effect of SOLO step size and root initialization on Caltech-256 and LEDGAR: online and PU CW-$F_{0.25}$ (%), reported as mean $\pm$ sample standard deviation over 5 runs. The best mean is in **bold underlined**, the second best in ***bold italic*** within each parameter block and column.

	CALTECH-256		LEDGAR	
Parameter	Online	PU	Online	PU
$\alpha_0 = 0.001$	85.04 ± 1.38	86.36 ± 1.71	83.41 ± 1.59	83.12 ± 2.97
$\alpha_0 = 0.01$	85.29 ± 1.45	86.57 ± 1.70	83.44 ± 1.59	83.14 ± 3.00
$\alpha_0 = 0.1$	87.70 ± 1.49	89.13 ± 1.28	84.19 ± 1.24	83.65 ± 3.00
$\alpha_0 = 0.5$	90.93 ± 1.06	91.80 ± 0.70	85.24 ± 1.19	85.23 ± 3.12
$\alpha_0 = 1$	91.44 ± 0.82	92.09 ± 0.65	86.79 ± 0.62	87.08 ± 1.50
$\alpha_0 = 2$	***91.51*** ± 0.79	92.09 ± 0.66	***87.15*** ± 0.88	**88.22** ± 1.35
$\alpha_0 = 10$	**91.55** ± 0.82	***92.10*** ± 0.72	**87.53** ± 1.28	87.99 ± 1.61
$\alpha_0 = 100$	91.41 ± 0.82	92.10 ± 0.66	87.14 ± 0.83	***88.06*** ± 1.69
$\alpha_0 = 1000$	90.83 ± 0.75	**92.29** ± 0.57	83.84 ± 2.20	87.90 ± 1.03
$\alpha_e = 0.5$	**91.57** ± 0.88	92.05 ± 0.66	***87.20*** ± 1.04	***88.02*** ± 1.69
$\alpha_e = 0.75$	***91.52*** ± 0.80	**92.09** ± 0.68	**87.37** ± 1.17	**88.09** ± 1.61
$\alpha_e = 1$	91.44 ± 0.82	***92.09*** ± 0.65	86.79 ± 0.62	87.08 ± 1.50
$\alpha_e = 1.5$	89.61 ± 1.46	90.46 ± 0.93	83.99 ± 1.57	83.57 ± 3.11
$\alpha_e = 2$	87.31 ± 1.81	88.50 ± 1.43	83.56 ± 1.60	83.24 ± 3.03
$\lambda_0 = 0$	91.31 ± 0.90	92.09 ± 0.67	80.75 ± 3.00	82.63 ± 3.65
$\lambda_0 = 0.25$	91.40 ± 0.83	***92.11*** ± 0.65	84.33 ± 0.82	84.72 ± 3.63
$\lambda_0 = 0.5$	91.44 ± 0.82	92.09 ± 0.65	***86.79*** ± 0.62	***87.08*** ± 1.50
$\lambda_0 = 0.75$	***91.49*** ± 0.84	92.10 ± 0.66	**87.47** ± 1.23	**88.28** ± 1.49
$\lambda_0 = 1$	**91.52** ± 0.85	**92.12** ± 0.67	85.96 ± 1.01	87.07 ± 1.96

Table 11: Log loss and Brier score on the online/PU union for different CPE-training fractions, reported as mean $\pm$ sample standard deviation over 5 runs. Scores are averaged over label–example pairs; lower is better.

Dataset	CPE train size	Log loss	Brier score
YouTube	100%	0.09651 ± 0.00085	0.02447 ± 0.00022
	40%	0.10024 ± 0.00094	0.02507 ± 0.00024
	20%	0.10545 ± 0.00113	0.02589 ± 0.00028
Flickr	100%	0.02382 ± 0.00019	0.00520 ± 0.00002
	40%	0.02491 ± 0.00017	0.00529 ± 0.00002
	20%	0.02601 ± 0.00017	0.00539 ± 0.00002

Table 12: Effect of CPE-training fraction on YouTube and Flickr: online and PU F_1 (%), reported as mean $\pm$ sample standard deviation over 5 runs. The best mean is in **bold underlined**, the second best in ***bold italic*** within each fraction, dataset, and column. Ma and Mi denote macro and micro averaging. $\star$ marks a batch baseline; $\times$ denotes an unsupported metric for that baseline.

	YouTube				Flickr			
	Online		PU		Online		PU	
Method	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1
	CPE train size: 100%							
$\hat{\boldsymbol{\eta}} > 0.5$	22.79 ± 0.55	30.97 ± 0.47	22.90 ± 0.24	30.86 ± 0.84	18.86 ± 0.75	29.54 ± 0.40	18.66 ± 0.28	29.75 ± 0.37
OFO	36.17 ± 0.51	43.35 ± 0.56	***36.37*** ± 0.24	**43.30** ± 0.14	30.22 ± 0.64	40.55 ± 0.40	30.65 ± 0.23	**41.22** ± 0.29
Greedy	36.29 ± 0.33	$\times$	36.35 ± 0.28	$\times$	30.86 ± 0.44	$\times$	***30.69*** ± 0.27	$\times$
OMMA	36.33 ± 0.37	**43.46** ± 0.62	36.34 ± 0.26	***43.29*** ± 0.16	***30.89*** ± 0.45	40.88 ± 0.42	30.69 ± 0.26	41.16 ± 0.30
SOLO	**36.45** ± 0.36	***43.43*** ± 0.54	**36.39** ± 0.30	43.27 ± 0.16	30.87 ± 0.41	***40.92*** ± 0.42	**30.70** ± 0.21	***41.21*** ± 0.32
Thr. $(\boldsymbol{y})^\star$	$\times$	$\times$	36.01 ± 0.34	43.26 ± 0.07	$\times$	$\times$	29.95 ± 0.39	41.20 ± 0.26
Thr. $(\hat{\boldsymbol{\eta}})^\star$	***36.45*** ± 0.41	43.42 ± 0.60	$\times$	$\times$	**31.14** ± 0.50	**40.95** ± 0.36	$\times$	$\times$
	CPE train size: 40%							
$\hat{\boldsymbol{\eta}} > 0.5$	23.28 ± 0.80	31.26 ± 0.55	23.30 ± 0.57	31.11 ± 0.61	17.45 ± 0.48	28.73 ± 0.36	17.47 ± 0.47	29.01 ± 0.39
OFO	34.31 ± 0.45	41.55 ± 0.31	***34.45*** ± 0.57	***41.61*** ± 0.22	28.21 ± 0.39	39.25 ± 0.36	***28.68*** ± 0.24	***39.79*** ± 0.38
Greedy	34.48 ± 0.54	$\times$	34.43 ± 0.56	$\times$	***28.86*** ± 0.37	$\times$	28.66 ± 0.17	$\times$
OMMA	34.50 ± 0.52	***41.61*** ± 0.39	34.43 ± 0.57	41.61 ± 0.20	28.83 ± 0.38	39.41 ± 0.37	28.66 ± 0.18	39.64 ± 0.39
SOLO	***34.63*** ± 0.51	41.57 ± 0.39	**34.51** ± 0.46	41.58 ± 0.19	28.85 ± 0.35	***39.50*** ± 0.35	**28.78** ± 0.23	39.76 ± 0.37
Thr. $(\boldsymbol{y})^\star$	$\times$	$\times$	34.17 ± 0.43	**41.75** ± 0.32	$\times$	$\times$	27.99 ± 0.48	**39.79** ± 0.36
Thr. $(\hat{\boldsymbol{\eta}})^\star$	**34.63** ± 0.49	**41.82** ± 0.38	$\times$	$\times$	**29.02** ± 0.52	**39.55** ± 0.35	$\times$	$\times$
	CPE train size: 20%							
$\hat{\boldsymbol{\eta}} > 0.5$	23.78 ± 0.89	31.55 ± 0.51	23.36 ± 0.31	31.18 ± 0.62	15.84 ± 0.30	28.04 ± 0.42	15.80 ± 0.22	28.23 ± 0.24
OFO	32.46 ± 0.46	***39.48*** ± 0.24	***32.58*** ± 0.45	***39.21*** ± 0.41	26.60 ± 0.32	38.14 ± 0.41	26.77 ± 0.26	***38.51*** ± 0.33
Greedy	32.58 ± 0.41	$\times$	32.50 ± 0.42	$\times$	***27.32*** ± 0.34	$\times$	***26.82*** ± 0.15	$\times$
OMMA	32.58 ± 0.42	39.44 ± 0.25	32.50 ± 0.41	39.18 ± 0.40	27.26 ± 0.28	38.26 ± 0.35	26.79 ± 0.18	38.30 ± 0.32
SOLO	***32.66*** ± 0.44	39.35 ± 0.25	**32.67** ± 0.42	39.18 ± 0.38	27.31 ± 0.36	***38.38*** ± 0.35	**26.92** ± 0.21	38.43 ± 0.32
Thr. $(\boldsymbol{y})^\star$	$\times$	$\times$	36.01 ± 0.34	43.26 ± 0.07	$\times$	$\times$	29.95 ± 0.39	41.20 ± 0.26
Thr. $(\hat{\boldsymbol{\eta}})^\star$	***36.45*** ± 0.41	43.42 ± 0.60	$\times$	$\times$	**31.14** ± 0.50	**40.95** ± 0.36	$\times$	$\times$
Thr. $(\boldsymbol{y})^\star$	$\times$	$\times$	32.55 ± 0.36	**39.53** ± 0.37	$\times$	$\times$	26.64 ± 0.11	**38.55** ± 0.29
Thr. $(\hat{\boldsymbol{\eta}})^\star$	**32.78** ± 0.63	**39.92** ± 0.24	$\times$	$\times$	**27.45** ± 0.34	**38.43** ± 0.39	$\times$	$\times$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the sequential ETU/PU setting, the concave-utility results, the LF Dinkelbach-root method, and the stated regret rates. The scope and assumptions are then formalized in Sections 2 to 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The theorem statements make the main restrictions explicit, including i.i.d. data, access to a CPE, smoothness/concavity assumptions for one class, and denominator positivity for LF utilities. The discussion also notes that a matching unrestricted lower bound for adaptive online ETU regret remains open.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The main theoretical statements in Sections 3 and 4 state their assumptions and are cross-referenced. Complete formal proofs are provided in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5 and Appendix G describe the datasets, random splits and shuffles, CPE construction, evaluated methods, metrics, and hyperparameter choices used for the main table. The algorithms themselves are specified in Algorithm 1 and Algorithm 2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The experiments use publicly available benchmark datasets, and the submission is accompanied by code and scripts for the proposed online methods and baselines. Appendix G describes the data preparation, CPE generation, method configuration, and result aggregation used to reproduce the reported results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix G specifies the pooled train/online/PU split protocol, split and shuffle seeds, CPE training configuration, threshold baselines, online methods.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The results with standard errors are reported in Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provide general description of resources used for experiments, we believe this is enough, since experiments require only modest compute.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is theoretical and empirical research on classification-metric optimization using standard public benchmark datasets, with no human-subject experiments or release of high-risk models. The submission is prepared in anonymized form.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The work is foundational and not tied to a specific deployment.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the public benchmark datasets.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, user studies, or other research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, user studies, or other research with human subjects, so IRB or equivalent review is not applicable.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The paper do not use LLMs as an component in any way.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.