
Parameter-Free Online Learning via Model Selection

Dylan J. Foster
Cornell University

Satyen Kale
Google Research

Mehryar Mohri
NYU and Google Research

Karthik Sridharan
Cornell University

Abstract

We introduce an efficient algorithmic framework for model selection in online learning, or parameter-free online learning. Our algorithms satisfy oracle inequalities in the adversarial online learning setting. Unlike previous work in this area that has focused on specific, highly structured function classes, such as nested balls in a Hilbert space, we propose a generic meta-algorithm framework that achieves model selection oracle inequalities under minimal structural assumptions: we give the first computationally efficient algorithms that work in arbitrary Banach spaces under mild smoothness assumptions — previous results only applied to Hilbert spaces. We further derive new oracle inequalities for various matrix classes, non-nested convex sets, and \mathbb{R}^d with generic regularizers. Finally, we generalize these results by providing oracle inequalities for arbitrary non-linear classes in the contextual learning model; in particular, we give new algorithms for learning with multiple kernels. These results are all derived through a unified meta-algorithm scheme using a novel “multi-scale” algorithm for prediction with expert advice based on random payout, which may be of independent interest.

1 Introduction

A key problem in the design of learning algorithms is the choice of the hypothesis set \mathcal{F} . This is known as the *model selection* problem. The choice of \mathcal{F} is driven by inherent trade-offs. In the statistical learning setting, this can be analyzed in terms of the *estimation* and *approximation errors*. A richer or more complex \mathcal{F} helps better approximate the Bayes predictor (smaller approximation error). On the other hand, a hypothesis set that is too complex may have too large a VC-dimension or an unfavorable Rademacher complexity, thereby resulting in looser guarantees on the difference of the loss of a hypothesis and that of the best-in class (large estimation error).

In the batch setting, this problem has been extensively studied with the main ideas originating in the seminal work of [41] and [40] and the principle of Structural Risk Minimization (SRM). It is typically formulated as follows: let $(\mathcal{F}_i)_{i \in \mathbb{N}}$ be an infinite sequence of hypothesis sets (or models); the problem consists of using the training sample to select a hypothesis set \mathcal{F}_i with a favorable trade-off and choose the best hypothesis f in \mathcal{F}_i .

If we had access to a hypothetical oracle informing us of the best choice of i for a given problem, then the problem would reduce to the standard one of learning with a fixed hypothesis set. Remarkably though, techniques such as SRM, or similar penalty-based model selection methods, return a hypothesis f^* that enjoys finite-sample learning guarantees that are almost as favorable as those that would be obtained had an oracle informed us of the index i^* of the best-in-class classifier’s hypothesis set [40; 13; 37; 22; 4; 25]. Such guarantees are sometimes referred to as *oracle inequalities*. They can be derived even for data-dependent penalties [22; 4; 3].

Such results naturally raise the following questions in the online setting, which we study in this paper: can we develop an analogous theory of model selection in online learning? Can we design online algorithms for model selection with solutions benefitting from strong guarantees, analogous to the batch ones? Unlike the statistical setting, in online learning, one cannot split samples to first learn the optimal predictor within each subclass and then later learn the optimal subclass choice.

A series of recent works on online learning provide some positive results along that direction. On the algorithmic side, [26; 28; 31; 32] present solutions that efficiently achieve oracle inequalities for the (important) special case where $\mathcal{F}_1, \mathcal{F}_2, \dots$ is a sequence of nested balls in a Hilbert space. On the theoretical side, a different line of work focusing on general hypothesis classes [14] uses martingale-based sequential complexity measures to show that, information-theoretically, one can obtain oracle inequalities in the online setting at a level of generality comparable to that of the batch statistical learning. However, this last result is not algorithmic.

The first approach that a familiar reader might think of for tackling the online model selection problem is to run, for each i , an online learning algorithm that minimizes regret against \mathcal{F}_i and aggregate over these algorithms using the multiplicative weights algorithm for prediction with expert advice. This would work if all the losses or “experts” considered were uniformly bounded by a reasonably small quantity. However, in most reasonable problems, the losses of predictors or experts for \mathcal{F}_i are bounded by some quantity that grows with i . Using simple aggregation would scale our regret with the magnitude of the largest \mathcal{F}_i and not the i^* we want to compare against. This is the main technical challenge faced in this context and that we fully address in this paper.

This work presents an efficient algorithmic framework for online model selection in the adversarial setting, an online analogue of SRM. Our results are based on a novel *multi-scale algorithm* for prediction with expert advice. This algorithm works in a situation where the different experts’ losses lie in different ranges, and guarantees that the regret to each individual expert is adapted to the range of its losses. The algorithm can also take advantage of a given prior over the experts reflecting their importance. This general, abstract setting of prediction with expert advice yields algorithms for a host of applications detailed below in a straightforward manner.

First, we give efficient algorithms for model selection for nested linear classes that provide oracle inequalities in terms of the norm of the benchmark to which the algorithm’s performance is compared. Our algorithm works for any norm, which considerably generalizes previous work [26; 28; 31; 32] from Hilbert spaces to arbitrary normed vector spaces. For most of the classes considered, we give the first polynomial-time algorithms. This includes online oracle inequalities for high-dimensional learning tasks such as online PCA and online matrix prediction. We then generalize these results even further by providing oracle inequalities for arbitrary non-linear classes in the contextual learning model. This yields applications for online penalized risk minimization and multiple kernel learning. Due to space limitations, all proofs in the paper have been moved to the appendices in the supplementary material, with proof sketches and intuitions in the main body of the paper.

1.1 Preliminaries

Notation. For a given norm $\|\cdot\|$, let $\|\cdot\|_*$ denote the dual norm. Likewise, for any function F , F^* will denote its Fenchel conjugate. For a Banach space $(\mathfrak{B}, \|\cdot\|)$, the dual is $(\mathfrak{B}^*, \|\cdot\|_*)$. We use $x_{1:n}$ as shorthand for a sequence of vectors (x_1, \dots, x_n) . For such sequences, we will use $x_t[i]$ to denote the t th vector’s i th coordinate. We let e_i denote the i th standard basis vector. $\|\cdot\|_p$ denotes the ℓ_p norm, $\|\cdot\|_\sigma$ denotes the spectral norm, and $\|\cdot\|_\Sigma$ denotes the trace norm.

Setup and Goals. We work in two closely related settings: online convex optimization (Protocol 1) and online supervised learning (Protocol 2). In online convex optimization, the learner selects decisions from a convex subset \mathcal{W} of some Banach space \mathfrak{B} . Regret to a comparator $w \in \mathcal{W}$ in this setting is defined as $\sum_{t=1}^n f_t(w_t) - \sum_{t=1}^n f_t(w)$. Suppose \mathcal{W} can be decomposed into sets $\mathcal{W}_1, \mathcal{W}_2, \dots$. For a fixed set \mathcal{W}_k , the optimal regret, if one tailors the algorithm to compete with \mathcal{W}_k , is typically characterized by some measure of intrinsic complexity of the class (such as Littlestone’s dimension [5] and sequential Rademacher complexity [34]), denoted $\mathbf{Comp}_n(\mathcal{W}_k)$. We would like to develop algorithms that predict a sequence (w_t) such that

$$\sum_{t=1}^n f_t(w_t) - \min_{w \in \mathcal{W}_k} \sum_{t=1}^n f_t(w) \leq \mathbf{Comp}_n(\mathcal{W}_k) + \mathbf{Pen}_n(k) \quad \forall k. \quad (1)$$

This equation is called an *oracle inequality* and states that the performance of the sequence (w_t) matches that of a comparator that minimizes the bias-variance tradeoff $\min_k \{ \min_{w \in \mathcal{W}_k} \sum_{t=1}^n f_t(w) + \mathbf{Comp}_n(\mathcal{W}_k) \}$, up to a penalty $\mathbf{Pen}_n(k)$ whose scale ideally matches that of $\mathbf{Comp}_n(\mathcal{W}_k)$. We shall see shortly that ensuring that the scale of $\mathbf{Pen}_n(k)$ does

Protocol 1 Online Convex Optimization

for $t = 1, \dots, n$ **do**
 Learner selects strategy $q_t \in \Delta(\mathcal{W})$ for convex decision set \mathcal{W} .
 Nature selects convex loss $f_t: \mathcal{W} \rightarrow \mathbb{R}$.
 Learner draws $w_t \sim q_t$ and incurs loss $f_t(w_t)$.
end for

indeed match is the core technical challenge in developing online oracle inequalities for commonly used classes.

In the supervised learning setting we measure regret against a benchmark class $\mathcal{F} = \bigcup_{k=1}^{\infty} \mathcal{F}_k$ of functions $f: \mathcal{X} \rightarrow \mathbb{R}$, where \mathcal{X} is some abstract context space, also called feature space. In this case, the desired oracle inequality has the form:

$$\sum_{t=1}^n \ell(\hat{y}_t, y_t) - \inf_{f \in \mathcal{F}_k} \sum_{t=1}^n \ell(f(x_t), y_t) \leq \mathbf{Comp}_n(\mathcal{F}_k) + \mathbf{Pen}_n(k) \quad \forall k. \quad (2)$$

Protocol 2 Online Supervised Learning

for $t = 1, \dots, n$ **do**
 Nature provides $x_t \in \mathcal{X}$.
 Learner selects randomized strategy $q_t \in \Delta(\mathbb{R})$.
 Nature provides outcome $y_t \in \mathcal{Y}$.
 Learner draws $\hat{y}_t \sim q_t$ and incurs loss $\ell(\hat{y}_t, y_t)$.
end for

2 Online Model Selection

2.1 The need for multi-scale aggregation

Let us briefly motivate the main technical challenge overcome by the model selection approach we consider. The most widely studied oracle inequality in online learning has the following form

$$\sum_{t=1}^n f_t(w_t) - \sum_{t=1}^n f_t(w) \leq O\left(\left(\|w\|_2 + 1\right)\sqrt{n \cdot \log\left(\left(\|w\|_2 + 1\right)n\right)}\right) \quad \forall w \in \mathbb{R}^d. \quad (3)$$

In light of (1), a *model selection approach* to obtaining this inequality would be to split the set $\mathcal{W} = \mathbb{R}^d$ into ℓ_2 norm balls of doubling radius, i.e. $\mathcal{W}_k = \{w \mid \|w\|_2 \leq 2^k\}$. A standard fact [16] is that such a set has $\mathbf{Comp}_n(\mathcal{W}_k) = 2^k \sqrt{n}$ if one optimizes over it using Mirror Descent, and so obtaining the oracle inequality (1) is sufficient to recover (3), so long as $\mathbf{Pen}_n(k)$ is not too large relative to $\mathbf{Comp}_n(\mathcal{W}_k)$.

Online model selection is fundamentally a problem of prediction with expert advice [8], where the experts correspond to the different model classes one is choosing from. Our basic meta-algorithm, MULTISCALEFTPL (Algorithm 3), operates in the following setup. The algorithm has access to a finite number, N , of experts. In each round, the algorithm is required to choose one of the N experts. Then the losses of all experts is revealed, and the algorithm incurs the loss of the chosen expert.

The twist from the standard setup is that the losses of all the experts are *not* uniformly bounded in the same range. Indeed, for the setup described for the oracle inequality (3), class \mathcal{W}_k will produce predictions with norm as large as 2^k . Therefore, here, we assume that expert i incurs losses in the range $[-c_i, c_i]$, for some known parameter $c_i \geq 0$. The goal is to design an online learning algorithm whose regret to expert i scales with c_i , rather than $\max_i c_i$, which is what previous algorithms for learning from expert advice (such as the standard multiplicative weights strategy or AdaHedge [12]) would achieve. Indeed, any regret bound scaling in $\max_i c_i$ will be far too large to achieve (3), as the term $\mathbf{Pen}_n(k)$ will dominate. This new type of scale-sensitive regret bound, achieved by our algorithm MULTISCALEFTPL, is stated below.

Algorithm 3

procedure MULTISCALEFTPL(c, π) ▷ Scale vector c with $c_i \geq 1$, Prior distribution π .
for time $t = 1, \dots, n$: **do**
 Draw sign vectors $\sigma_{t+1}, \dots, \sigma_n \in \{\pm 1\}^N$ each uniformly at random.
 Compute distribution
$$p_t(\sigma_{t+1:n}) = \arg \min_{p \in \Delta_N} \sup_{g_t: |g_t[i]| \leq c_i} \left[\langle p, g_t \rangle + \sup_i \left[- \sum_{s=1}^t \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right] \right],$$

 where $B(i) = 5c_i \sqrt{n \log(4c_i^2 n / \pi_i)}$.
 Play $i_t \sim p_t$.
 Observe loss vector g_t .
end for
end procedure

Theorem 1. *Suppose the loss sequence $(g_t)_{t \leq n}$ satisfies $|g_t[i]| \leq c_i$ for a sequence $(c_i)_{i \in [N]}$ with each $c_i \geq 1$. Let $\pi \in \Delta_N$ be a given prior distribution on the experts. Then, playing the strategy $(p_t)_{t \leq n}$ given by [Algorithm 3](#), MULTISCALEFTPL yields the following regret bound:¹*

$$\mathbb{E} \left[\sum_{t=1}^n \langle e_{i_t}, g_t \rangle - \sum_{t=1}^n \langle e_i, g_t \rangle \right] \leq O \left(c_i \sqrt{n \log(nc_i / \pi_i)} \right) \quad \forall i \in [N]. \quad (4)$$

The proof of the theorem is deferred to [Appendix A](#) in the supplementary material due to space constraints. Briefly, the proof follows the technique of adaptive relaxations from [\[14\]](#). It relies on showing that the following function of the first t loss vectors $g_{1:t}$ is an admissible relaxation (see [\[14\]](#) for definitions):

$$\mathbf{Rel}(g_{1:t}) \triangleq \mathbb{E}_{\sigma_{t+1}, \dots, \sigma_T \in \{\pm 1\}^N} \sup_i \left[- \sum_{s=1}^t \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^T \sigma_s[i] c_i - B(i) \right].$$

This implies that if we play the strategy $(p_t)_{t \leq n}$ given by [Algorithm 3](#), the regret to the i th expert is bounded by $B(i) + \mathbf{Rel}(\cdot)$, where $\mathbf{Rel}(\cdot)$ indicates the \mathbf{Rel} function applied to an empty sequence of loss vectors. Then, as a final step, we bound $\mathbf{Rel}(\cdot)$ using a probabilistic maximal inequality ([Lemma 2](#) in the supplementary material), yielding the given bound. Compared to related FTPL algorithms [\[35\]](#), the analysis is surprisingly delicate, as additive c_i factors can spoil the desired regret bound [\(4\)](#) if the c_i s differ by orders of magnitude.

The min-max optimization problem in MULTISCALEFTPL can be solved in polynomial-time using linear programming — see [Appendix A.1](#) in the supplementary material for a full discussion.

In related work, [\[7\]](#) simultaneously developed a multi-scale experts algorithm which could also be used in our framework. Their regret bound has sub-optimal dependence on the prior distribution over experts, but their algorithm is more efficient and is able to obtain multiplicative regret guarantees.

2.2 Online convex optimization

One can readily apply MULTISCALEFTPL for online optimization problems whenever it is possible to obtain good bounds on the losses of the different experts. One such application is to online convex optimization, where such a bound can be obtained via appropriate bounds on the relevant norms of the parameter vectors and the gradients of the loss functions. We detail this application below.

We now show how to apply MULTISCALEFTPL in the online convex optimization framework to derive algorithms for parameter-free online learning and more. All of the algorithms in this section are derived using a unified meta-algorithm strategy MULTISCALEOCO.

¹This regret bound holds under expectation over the player’s randomization. It is assumed that each g_t is selected before the randomized strategy p_t is revealed, but may adapt to the distribution over p_t . In fact, a slightly stronger version of this bound holds, namely $\mathbb{E} \left[\sum_{t=1}^n \langle e_{i_t}, g_t \rangle - \min_{i \in [N]} \left\{ \sum_{t=1}^n \langle e_i, g_t \rangle + O \left(c_i \sqrt{n \log(nc_i / \pi_i)} \right) \right\} \right] \leq 0$. A similar strengthening applies to all subsequent bounds.

The setup is as follows. We have access to N sub-algorithms, denoted ALG_i for $i \in [N]$. In round t , each sub-algorithm ALG_i produces a prediction $w_t^i \in \mathcal{W}_i$, where \mathcal{W}_i is a set in a vector space V over \mathbb{R} containing 0. Our desired meta-algorithm is then required to choose one of the predictions w_t^i . Then, a loss function $f_t : V \rightarrow \mathbb{R}$ is revealed, whereupon ALG_i incurs loss $f_t(w_t^i)$, and the meta-algorithm suffers the loss of the chosen prediction. We make the following assumption on the sub-algorithms:

Assumption 1. The sub-algorithms satisfy the following conditions:

- For each $i \in [N]$, there is an associated norm $\|\cdot\|_{(i)}$ such that $\sup_{w \in \mathcal{W}_i} \|w\|_{(i)} \leq R_i$.
- For each $i \in [N]$, the sequence of functions f_t are L_i -Lipschitz on \mathcal{W}_i with respect to $\|\cdot\|_{(i)}$.
- For each sub-algorithm ALG_i , the iterates $(w_t^i)_{t \leq n}$ enjoy a regret bound $\sum_{t=1}^n f_t(w_t^i) - \inf_{w \in \mathcal{W}_i} \sum_{t=1}^n f_t(w) \leq \mathbf{Reg}_n(i)$, where $\mathbf{Reg}_n(i)$ may be data- or algorithm-dependent.

Algorithm 4

```

1: procedure MULTISCALEOCO( $\{\text{ALG}_i, R_i, L_i\}_{i \in [N]}, \pi$ )           ▷ Collection of sub-algorithms, prior  $\pi$ .
2:    $c \leftarrow (R_i \cdot L_i)_{i \in [N]}$                                ▷ Sub-algorithm scale parameters.
3:   for  $t = 1, \dots, n$  do
4:      $w_t^i \leftarrow \text{ALG}_i(\tilde{f}_1, \dots, \tilde{f}_{t-1})$  for each  $i \in \mathcal{A}$ .
5:      $i_t \leftarrow \text{MULTISCALEFTPL}[c, \pi](g_1, \dots, g_{t-1})$ .
6:     Play  $w_t = w_t^{i_t}$ .
7:     Observe loss function  $f_t$  and let  $\tilde{f}_t(w) = f_t(w) - f_t(0)$ .
8:      $g_t \leftarrow (\tilde{f}_t(w_t^i))_{i \in [N]}$ .
9:   end for
10: end procedure

```

In most applications, \mathcal{W}_i will be a convex set and f_t a convex function; this convexity is not necessary to prove a regret bound for the meta-algorithm. We simply need boundedness of the set \mathcal{W}_i and Lipschitzness of the functions f_t , as specified in [Assumption 1](#). This assumption implies that for any i , we have $|f_t(w) - f_t(0)| \leq R_i L_i$ for any $w \in \mathcal{W}_i$. Thus, we can design a meta-algorithm for this setup by using MULTISCALEFTPL with $c_i = R_i L_i$, which is precisely the MULTISCALEOCO algorithm ([Algorithm 4](#)). The following theorem provides a bound on the regret of MULTISCALEOCO; a direct consequence of [Theorem 1](#).

Theorem 2. *Without loss of generality, assume that $R_i L_i \geq 1^2$. Suppose that the inputs to [Algorithm 4](#) satisfy [Assumption 1](#). Then, the iterates $(w_t)_{t \leq n}$ returned by [Algorithm 4](#) follow the regret bound*

$$\mathbb{E} \left[\sum_{t=1}^n f_t(w_t) - \inf_{w \in \mathcal{W}_i} \sum_{t=1}^n f_t(w) \right] \leq \mathbb{E}[\mathbf{Reg}_n(i)] + O\left(R_i L_i \sqrt{n \log(R_i L_i n / \pi_i)}\right) \quad \forall i \in [N]. \quad (5)$$

[Theorem 2](#) shows that if we use [Algorithm 4](#) to aggregate the iterates produced by a collection of sub-algorithms $(\text{ALG}_i)_{i \in [N]}$, the regret against any sub-algorithm i will only depend on that algorithm's scale, not the regret of the worst sub-algorithm.

Application 1: Parameter-free online learning in uniformly convex Banach spaces. Using our general framework, we can give a generalization of the parameter-free online learning bounds found in [26; 28; 31; 32; 10] from Hilbert spaces to arbitrary uniformly convex Banach spaces. Recall that a Banach space $(\mathfrak{B}, \|\cdot\|)$ is $(2, \lambda)$ -uniformly convex if $\frac{1}{2}\|\cdot\|^2$ is λ -strongly convex with respect to itself [33]. Our algorithm is efficient whenever mirror descent over $(\mathfrak{B}, \|\cdot\|)$ is efficient because it is an instantiation of MULTISCALEOCO with the following collection \mathcal{A} of $N = n + 1$ sub-algorithms: for each $i \in [n + 1]$, we set $R_i = 2^{i-1}$, $L_i = L$, $\mathcal{W}_i = \{w \in \mathfrak{B} \mid \|w\| \leq R_i\}$, $\eta_i = \sqrt{\lambda \frac{R_i}{L n}}$, and $\text{ALG}_i = \text{MIRRORDESCENT}(\eta_i, \mathcal{W}_i, \|\cdot\|^2)$. Finally, we set $\pi = \text{Uniform}([n + 1])$.

Mirror descent is a standard tool for online convex optimization and is described precisely in [Appendix A.2](#) in the supplementary material, but the only feature of its performance that will be important to us is that, when configured as described above, the iterates $(w_t^i)_{t \leq n}$ produced by ALG_i

²For notational convenience, all Lipschitz bounds are assumed to be at least 1 without loss of generality for the remainder of the paper.

specified above will satisfy $\sum_{t=1}^n f_t(w_t^i) - \inf_{w \in \mathcal{W}_i} \sum_{t=1}^n f_t(w) \leq O(R_i L \sqrt{\lambda n})$ on any sequence of losses that are L -Lipschitz with respect to $\|\cdot\|_*$. Using just this simple fact, combined with the regret bound for MULTISCALEOCO (plus a few technical details in [Appendix A.2](#)), we can deduce the following parameter-free regret bound:

Theorem 3 (Oracle Inequality for Uniformly Convex Banach Spaces). *The iterates $(w_t)_{t \leq n}$ produced by MULTISCALEOCO on any L -Lipschitz sequence of losses $(f_t)_{t \leq n}$ satisfy*

$$\mathbb{E} \left[\sum_{t=1}^n f_t(w_t) - \sum_{t=1}^n f_t(w) \right] \leq O \left(L \cdot (\|w\| + 1) \sqrt{n \cdot \log(L(\|w\| + 1)n)/\lambda} \right) \quad \forall w \in \mathfrak{B}. \quad (6)$$

Note that the above oracle inequality applies for **any uniformly convex norm** $\|\cdot\|$. Previous results only obtain bounds of this form efficiently when $\|\cdot\|$ is a Hilbert space norm or ℓ_1 . As is standard for such oracle inequality results, the bound is weaker than the optimal bound if $\|w\|$ were selected in advance, but only by a mild $\sqrt{\log(L(\|w\| + 1)n)}$ factor.

Proposition 1. The algorithm can be implemented in time $O(T_{\text{MD}} \cdot \text{poly}(n))$ per iteration, where T_{MD} is the time complexity of a single mirror descent update.

In the example above, the $(2, \lambda)$ -uniform convexity condition was mainly chosen because it is a familiar assumption. The result can be straightforwardly generalized to related notions such as q -uniform convexity (see [\[38\]](#) for discussion). More generally, the approach can be used to derive oracle inequalities with respect to general strongly convex regularizer \mathcal{R} defined over the space \mathcal{W} . Such a bound would have the form $O \left(L \cdot \sqrt{n(\mathcal{R}(w) + 1)} \cdot \log((\mathcal{R}(w) + 1)n) \right)$ for typical choices of \mathcal{R} . This example captures well-known *quantile bounds* [\[23\]](#) when one takes \mathcal{R} to be the KL-divergence and \mathcal{W} to be the simplex, or, in the matrix case, takes \mathcal{R} to be the quantum relative entropy and \mathcal{W} to be the set of density matrices, as in [\[19\]](#).

It is instructive to think of MULTISCALEOCO as executing a (scale-sensitive) online analogue of the structural risk minimization principle. We simply specify a set of subclasses and a prior π specifying the importance of each subclass, and we are guaranteed that the algorithm's performance matches that of each sub-class, plus a penalty depending on the prior weight placed on that subclass. The advantage of this approach is that the nested structure used in the [Theorem 3](#) is completely inessential. This leads to the exciting prospect of developing parameter-free algorithms over new and exotic set systems. One such example is given below.

Application 2: Oracle inequality for many ℓ_p norms. The MULTISCALEOCO framework easily allows us to obtain an oracle inequality with respect to *many ℓ_p norms in \mathbb{R}^d simultaneously*. To the best of our knowledge all previous works have only considered a single norm.

Theorem 4. *Suppose that the loss functions $(f_t)_{t \leq n}$ are all L_p -Lipschitz for each $p \in [1 + \delta, 2]$, for some $\delta > 0$, and that $L_p/L_{p+1/\log(d)} = O(1)$ for all p, p' in this range. Then, there is a computationally efficient algorithm that guarantees, for all $w \in \mathbb{R}^d$ and all $p \in [1 + \delta, 2]$,*

$$\mathbb{E} \left[\sum_{t=1}^n f_t(w_t) - \sum_{t=1}^n f_t(w) \right] \leq O \left((\|w\|_p + 1) L_p \sqrt{n \log((\|w\|_p + 1) L_p \log(d)n) / (p-1)} \right). \quad (7)$$

The configuration in the above theorem is described in full in [Appendix A.2](#) in the supplementary material. This strategy can be trivially extended to handle p in the range $(2, \infty)$. The inequality holds for $p \geq 1 + \delta$ rather than for $p \geq 1$ because the ℓ_1 norm is not uniformly convex, but this is easily rectified by changing the regularizer at $p = 1$; we omit this for simplicity of presentation.

The same strategy can also be applied to matrix optimization over $\mathbb{R}^{d \times d}$ by replacing the ℓ_p norm with the Schatten S_p norm. The Schatten S_p -norm has strong convexity parameter on the order of $p - 1$ (which matches the ℓ_p norm up to absolute constants [\[2\]](#)) so the only change to [Theorem 4](#) will be the running-time T_{MD} . Likewise, the approach applies to (p, q) -group norms for group-structured sparsity tasks [\[21\]](#).

Application 3: Adapting to rank for Online PCA For the online PCA task, the learner predicts from a class $\mathcal{W}_k = \{W \in \mathbb{R}^{d \times d} \mid W \geq 0, \|W\|_\sigma \leq 1, \langle W, I \rangle = k\}$. For a fixed value of k , such a class is a convex relaxation of the set of all rank k projection matrices. After producing a prediction W_t , we experience affine loss functions $f_t(W_t) = \langle I - W_t, Y_t \rangle$, where $Y_t \in \mathcal{Y}_\sigma = \mathbb{S}_+ \cap \mathbb{B}_\sigma$.

We leverage an analysis of online PCA due to [30] together with MULTISCALEOCO to derive an algorithm that competes with many values of the rank simultaneously. This gives the following result: **Theorem 5.** *There is an efficient algorithm for Online PCA with regret bound*

$$\mathbb{E} \left[\sum_{t=1}^n \langle I - W_t, Y_t \rangle - \min_{\substack{W \text{ projection} \\ \text{rank}(W)=k}} \sum_{t=1}^n \langle I - W, Y_t \rangle \right] \leq \tilde{O} \left(\sqrt{n \min\{k, d-k\}^2} \right) \quad \forall k \in [d/2].$$

For a fixed value of k , the above bound is already optimal up to log factors, but it holds for all k simultaneously.

Application 4: Adapting to norm for Matrix Multiplicative Weights In the matrix multiplicative weights setting [1] we consider hypothesis classes of the form $\mathcal{W}_r = \{W \in \mathbb{R}^{d \times d} \mid W \geq 0, \|W\|_\Sigma \leq r\}$. Losses are given by $f_t(W) = \langle W, Y_t \rangle$, where $\|Y_t\|_\sigma \leq 1$. For a fixed value of r , the matrix multiplicative weights strategy has regret against \mathcal{W}_r bounded by $O(r\sqrt{n \log d})$. Using this strategy for fixed r as a sub-algorithm for MULTISCALEOCO, we achieve the following oracle inequality efficiently:

Theorem 6. *There is an efficient matrix prediction algorithm with regret bound*

$$\mathbb{E} \left[\sum_{t=1}^n \langle W_t, Y_t \rangle - \sum_{t=1}^n \langle W, Y_t \rangle \right] \leq (\|W\|_\Sigma + 1) \sqrt{n \log d \log((\|W\|_\Sigma + 1)n)} \quad \forall W \geq 0. \quad (8)$$

A remark on efficiency All of our algorithms providing bounds for the form (6) instantiate $O(n)$ experts with MULTISCALEFTPL because, in general, the worst case w for achieving (6) can have norm as large as e^n . If one has an a priori bound — say B — on the range at which each f_t attains its minimum, then the number of experts be reduced to $O(\log(B))$.

2.3 Supervised learning

We now consider the online supervised learning problem, with the goal being to compete with a sequence of hypothesis class $(\mathcal{F}_k)_{k \in [N]}$ simultaneously. Working in this setting makes clear a key feature of the meta-algorithm approach: **We can efficiently obtain online oracle inequalities for arbitrary nonlinear function classes** — so long as we have an efficient algorithm for each \mathcal{F}_k . We obtain a supervised learning meta-algorithm by simply feeding the observed losses $\ell(\cdot, y_t)$ (which may even be non-convex) to the meta-algorithm MULTISCALEFTPL in the same fashion as MULTISCALEOCO. The resulting strategy, which is described in detail in [Appendix A.3](#) for completeness, is called MULTISCALELEARNING. We make the following assumptions analogous to [Assumption 1](#), which lead to the performance guarantee for MULTISCALELEARNING given in [Theorem 7](#) below.

Assumption 2. The sub-algorithms satisfy the following conditions:

- For each $i \in [N]$, the iterates $(\hat{y}_t^i)_{t \leq n}$ produced by sub-algorithm ALG_i satisfy $|\hat{y}_t^i| \leq R_i$.
- For each $i \in [N]$, the function $\ell(\cdot, y_t)$ is L_i -Lipschitz on $[-R_i, R_i]$.
- For each sub-algorithm ALG_i , the iterates $(\hat{y}_t^i)_{t \leq n}$ enjoy a regret bound $\mathbb{E} \left[\sum_{t=1}^n \ell(\hat{y}_t^i, y_t) - \inf_{f \in \mathcal{F}_i} \sum_{t=1}^n \ell(f(x_t), y_t) \right] \leq \mathbf{Reg}_n(i)$, where $\mathbf{Reg}_n(i)$ may be data- or algorithm-dependent.

Theorem 7. *Suppose that the inputs to [Algorithm 5](#) satisfy [Assumption 2](#). Then the iterates $(\hat{y}_t)_{t \leq n}$ produced by [Algorithm 5](#) enjoy the regret bound*

$$\mathbb{E} \left[\sum_{t=1}^n \ell(\hat{y}_t, y_t) - \inf_{f \in \mathcal{F}_i} \sum_{t=1}^n \ell(f(x_t), y_t) \right] \leq \mathbb{E}[\mathbf{Reg}_n(i)] + O \left(R_i L_i \sqrt{n \log(R_i L_i n / \pi_i)} \right) \quad \forall i \in [N]. \quad (9)$$

Online penalized risk minimization In the statistical learning setting, oracle inequalities for arbitrary sequence of hypothesis classes $\mathcal{F}_1, \dots, \mathcal{F}_K$ are readily available. Such inequalities are typically stated in terms of complexity parameters for the classes (\mathcal{F}_k) such as VC dimension or Rademacher complexity. For the online learning setting, it is well-known that *sequential Rademacher complexity* $\mathbf{Rad}_n(\mathcal{F})$ provides a sequential counterpart to these complexity measures [34], in that it generically characterizes the minimax optimal regret for a given class. We will obtain an oracle inequality in terms of this parameter.

Assumption 3. The sequence of hypothesis classes $\mathcal{F}_1, \dots, \mathcal{F}_K$ are such that

1. There is an efficient algorithm ALG_k producing iterates $(\hat{y}_t^k)_{t \leq n}$ satisfying $\sum_{t=1}^n \ell(\hat{y}_t^k, y_t) - \inf_{f \in \mathcal{F}_k} \sum_{t=1}^n \ell(f(x_t), y_t) \leq C \cdot \mathbf{Rad}_n(\mathcal{F}_k)$.
2. Each \mathcal{F}_k has output range $[-R_k, R_k]$, where $R_k \geq 1$ without loss of generality.
3. $\mathbf{Rad}_n(\mathcal{F}_k) = \Omega(R_k \sqrt{n})$ — this is obtained by all non-trivial classes.

Theorem 8 (Online penalized risk minimization). *Under Assumption 3 there is an efficient (in K) algorithm that achieves the following regret bound for any L -Lipschitz loss:*

$$\mathbb{E} \left[\sum_{t=1}^n \ell(\hat{y}_t, y_t) - \inf_{f \in \mathcal{F}_k} \sum_{t=1}^n \ell(f(x_t), y_t) \right] \leq O \left(L \cdot \mathbf{Rad}_n(\mathcal{F}_k) \cdot \sqrt{\log(kn)} \right) \quad \forall k. \quad (10)$$

As in the previous section, one can of course derive tighter regret bounds and more efficient (e.g. sublinear in K) algorithms if $\mathcal{F}_1, \mathcal{F}_2, \dots$ are nested.

Application: Multiple Kernel Learning

Theorem 9. *Let $\mathcal{H}_1, \dots, \mathcal{H}_N$ be reproducing kernel Hilbert spaces for which each \mathcal{H}_k has a kernel \mathbf{K} such that $\sup_{x \in \mathcal{X}} \sqrt{\mathbf{K}(x, x)} \leq B_k$. Then there is an efficient learning algorithm that guarantees*

$$\mathbb{E} \left[\sum_{t=1}^n \ell(\hat{y}_t, y_t) - \sum_{t=1}^n \ell(f(x_t), y_t) \right] \leq O \left(LB_k k (\|f\|_{\mathcal{H}_k} + 1) \sqrt{\log(LB_k n (\|f\|_{\mathcal{H}_k} + 1))} \right) \quad \forall k, \forall f \in \mathcal{H}_k$$

for any L -Lipschitz loss, whenever an efficient algorithm is available for the unit ball in each \mathcal{H}_k .

3 Discussion and Further Directions

Related work There are two directions in parameter-free online learning that have been explored extensively. The first explores bounds of the form (3); namely, the Hilbert space version of the more general setting explored in Section 2.2. Beginning with [27], which obtained a slightly looser rate than (3), research has focused on obtaining tighter dependence on $\|w\|_2$ and $\log(n)$ in this type of bound [26; 28; 31; 32]; all of these algorithms run in linear time per update step. Recent work [10; 11] has extended these results to the case where the Lipschitz constant is not known in advance. These works give lower bounds for general norms, but only give efficient algorithms for Hilbert spaces. Extending Algorithm 4 to reach this Pareto frontier of regret in the unknown Lipschitz setting as in [11] may be an interesting direction for future research.

The second direction concerns so-called “quantile bounds” [9; 23; 24; 32] for experts setting, where the learner’s decision set \mathcal{W} is the simplex Δ_d and losses are bounded in ℓ_∞ . The multi-scale machinery developed in this is not needed to obtain bounds for this setting because the losses are uniformly bounded across all model classes. Indeed, [14] recovered a basic form of quantile bound using the vanilla multiplicative weights strategy as a meta-algorithm. It is not known whether the more sophisticated data-dependent quantile bounds given in [23; 24] can be recovered similarly.

Losses with curvature. The $O(\sqrt{n})$ type regret bounds for the meta-algorithm derived in this paper are appropriate when the sub-algorithms themselves incur $O(\sqrt{n})$ regret bounds. However, assuming certain curvature properties (such as strong convexity, exp-concavity, stochastic mixability, etc [17; 39]) of the loss functions it is possible to construct sub-algorithms that admit significantly more favorable regret bounds ($O(\log n)$ or even $O(1)$). These are also referred to as “fast rates” in online learning. A natural direction for further study is to design a meta-algorithm that admits logarithmic or constant regret to each sub-algorithm, assuming that the loss functions of interest satisfy similar curvature properties, with the regret to each individual sub-algorithm adapted to the curvature parameters for that sub-algorithm. Perhaps surprisingly, for the special case of the logistic

loss, improper prediction and aggregation strategies similar to those proposed in this paper offer a way to circumvent known proper learning lower bounds [18]. This approach will be explored in detail in a forthcoming companion paper.

Computational efficiency. We suspect that a running-time of $O(n)$ may be unavoidable for inequalities like (6) through our approach, since we essentially do not make use of the relationship between sub-algorithms beyond using the nested class structure. In this sense, the high level of generality we work in is both a blessing and a curse. Whether the runtime of MULTISCALEFTPL can be brought down to match $O(n)$ is an open question. This question boils down to whether or not the min-max optimization the algorithm must solve can be done in 1) Linear time in the number of experts 2) strongly polynomial time in the scales c_i .

Acknowledgements

We thank Francesco Orabona and David Pal for inspiring initial discussions. Part of this work was done while DF was an intern at Google Research and while DF and KS were visiting the Simons Institute for the Theory of Computing. DF is supported by the NDSEG fellowship.

References

- [1] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.
- [2] Keith Ball, Eric A Carlen, and Elliott H Lieb. Sharp uniform convexity and smoothness inequalities for trace norms. *Inventiones mathematicae*, 115(1):463–482, 1994.
- [3] Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2003. ISSN 1532-4435.
- [4] Peter L. Bartlett, Stéphane Boucheron, and Gábor Lugosi. Model selection and error estimation. *Machine Learning*, 48(1-3):85–113, 2002.
- [5] Shai Ben-David, David Pal, and Shai Shalev-Shwartz. Agnostic online learning. In *Proceedings of the 22th Annual Conference on Learning Theory*, 2009.
- [6] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.
- [7] Sebastien Bubeck, Nikhil Devanur, Zhiyi Huang, and Rad Niazadeh. Online auctions and multi-scale online learning. *Accepted to The 18th ACM conference on Economics and Computation (EC 17)*, 2017.
- [8] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [9] Kamalika Chaudhuri, Yoav Freund, and Daniel J Hsu. A parameter-free hedging algorithm. In *Advances in neural information processing systems*, pages 297–305, 2009.
- [10] Ashok Cutkosky and Kwabena A Boahen. Online convex optimization with unconstrained domains and losses. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 748–756. 2016.
- [11] Ashok Cutkosky and Kwabena A. Boahen. Online learning without prior information. *The 30th Annual Conference on Learning Theory*, 2017.
- [12] Steven De Rooij, Tim Van Erven, Peter D Grünwald, and Wouter M Koolen. Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research*, 15(1):1281–1316, 2014.
- [13] Luc Devroye, Lázló Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

- [14] Dylan J Foster, Alexander Rakhlin, and Karthik Sridharan. Adaptive online learning. In *Advances in Neural Information Processing Systems*, pages 3375–3383, 2015.
- [15] S. Gerchinovitz and J. Yu. Adaptive and optimal online linear regression on ℓ_1 -balls. *Theoretical Computer Science*, 2013.
- [16] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.
- [17] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.
- [18] Elad Hazan, Tomer Koren, and Kfir Y Levy. Logistic regression: Tight bounds for stochastic and online optimization. In *Proceedings of The 27th Conference on Learning Theory*, pages 197–209, 2014.
- [19] Elad Hazan, Satyen Kale, and Shai Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. *SIAM J. Comput.*, 46(2):744–773, 2017. doi: 10.1137/120895731.
- [20] Sham M. Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in Neural Information Processing Systems 21*, pages 793–800. MIT Press, 2009.
- [21] Sham M Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Regularization techniques for learning with matrices. *Journal of Machine Learning Research*, 13(Jun):1865–1890, 2012.
- [22] Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Trans. Information Theory*, 47(5):1902–1914, 2001.
- [23] Wouter M Koolen and Tim Van Erven. Second-order quantile methods for experts and combinatorial games. In *Proceedings of The 28th Conference on Learning Theory*, pages 1155–1175, 2015.
- [24] Haipeng Luo and Robert E Schapire. Achieving all with no parameters: Adanormalhedge. In *Conference on Learning Theory*, pages 1286–1304, 2015.
- [25] Pascal Massart. Concentration inequalities and model selection. *Lecture Notes in Mathematics*, 1896, 2007.
- [26] Brendan McMahan and Jacob Abernethy. Minimax optimal algorithms for unconstrained linear optimization. In *Advances in Neural Information Processing Systems*, pages 2724–2732, 2013.
- [27] Brendan McMahan and Matthew Streeter. No-regret algorithms for unconstrained online convex optimization. In *Advances in neural information processing systems*, pages 2402–2410, 2012.
- [28] H. Brendan McMahan and Francesco Orabona. Unconstrained online linear learning in hilbert spaces: Minimax algorithms and normal approximations. In *Proceedings of The 27th Conference on Learning Theory*, pages 1020–1039, 2014.
- [29] Arkadi Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.
- [30] Jiazhong Nie, Wojciech Kotłowski, and Manfred K Warmuth. Online pca with optimal regrets. In *International Conference on Algorithmic Learning Theory*, pages 98–112. Springer, 2013.
- [31] Francesco Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems*, pages 1116–1124, 2014.
- [32] Francesco Orabona and Dávid Pál. From coin betting to parameter-free online learning. *arXiv preprint arXiv:1602.04128*, 2016.
- [33] Gilles Pisier. Martingales in banach spaces (in connection with type and cotype). course ihp, feb. 2–8, 2011. 2011.

- [34] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Random averages, combinatorial parameters, and learnability. *Advances in Neural Information Processing Systems 23*, pages 1984–1992, 2010.
- [35] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Relax and randomize: From value to algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2150–2158, 2012.
- [36] James Renegar. A polynomial-time algorithm, based on newton’s method, for linear programming. *Mathematical Programming*, 40(1):59–93, 1988.
- [37] John Shawe-Taylor, Peter L Bartlett, Robert C Williamson, and Martin Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE transactions on Information Theory*, 44(5):1926–1940, 1998.
- [38] Nati Srebro, Karthik Sridharan, and Ambuj Tewari. On the universality of online mirror descent. In *Advances in neural information processing systems*, pages 2645–2653, 2011.
- [39] Tim van Erven, Peter D. Grünwald, Nishant A. Mehta, Mark D. Reid, and Robert C. Williamson. Fast rates in statistical and online learning. *Journal of Machine Learning Research*, 16:1793–1861, 2015.
- [40] Vladimir Vapnik. *Estimation of dependences based on empirical data*, volume 40. Springer-Verlag New York, 1982.
- [41] Vladimir Vapnik and Alexey Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

A Proofs

A.1 Multi-scale FTPL algorithm

Proof of Theorem 1. Recall that $B(i) = 5c_i \sqrt{n(\log(1/\pi_i) + \log(4c_i^2 n))}$. Let \mathcal{C} denote the set $\{g \in \mathbb{R}^N \mid |g_i| \leq c_i \forall i \in [N]\}$. For a regret bound of the form $B(i) + K$ to be achievable by a randomized algorithm such as [Algorithm 3](#) we need

$$\mathcal{V}_n \triangleq \left\langle \left\langle \inf_{P_t \in \Delta(\Delta_N)} \sup_{g_t \in \mathcal{C}} \mathbb{E}_{P_t} \mathbb{E}_{i_t \sim P_t} \right\rangle \right\rangle_{t=1}^n \sup_{i \in [N]} \left[\sum_{t=1}^n \langle e_{i_t}, g_t \rangle - \sum_{t=1}^n \langle e_i, g_t \rangle - B(i) \right] \leq K,$$

where $\langle \star \rangle_{t=1}^n$ denotes interleaving of the operator \star from $t = 1$ to n . In the context of [Algorithm 3](#), the distributions p_t above refer to the strategy $p_t(\sigma_{t+1:n})$ selected by the algorithm, and P_t refers to the distribution over this strategy induced by sampling the random variables $\sigma_{t+1:n}$.

See [\[14\]](#) for a more extensive introduction to the minimax analysis of comparator-dependent regret bounds.

We will develop an algorithm to certify this bound (for $K = 1$ specifically) using the framework of adaptive relaxations proposed by [\[14\]](#). Define a relaxation

$$\mathbf{Rel}(g_{1:t}) \triangleq \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{i \in [N]} \left[- \sum_{s=1}^t \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right].$$

The proof structure is as follows: We simply verify that playing p_t as suggested by [Algorithm 3](#) with \mathbf{Rel} satisfies the initial condition and admissibility condition for adaptive relaxations from [\[14\]](#), which implies that if we play p_t we will have $\mathbf{Reg}_n(i) \leq B(i) + \mathbf{Rel}(\cdot)$. Then as a final step we bound $\mathbf{Rel}(\cdot)$ using a probabilistic maximal inequality, [Lemma 2](#).

Initial Condition This condition asks that the initial value of the relaxation \mathbf{Rel} upper bound the worst-case value of the negative benchmark minus the bound $B(i)$ (in other words, the inner part of \mathcal{V}_n with the learner's loss removed). This holds by definition and is trivial to verify:

$$\mathbf{Rel}(g_{1:n}) = \sup_{i \in [N]} \left[- \sum_{t=1}^n \langle e_i, g_t \rangle - B(i) \right].$$

Admissibility For this step we must show that the inequality

$$\inf_{P_t \in \Delta(\Delta_N)} \sup_{g_t \in \mathcal{C}} \mathbb{E}_{P_t} \mathbb{E}_{i_t \sim P_t} [\langle e_{i_t}, g_t \rangle + \mathbf{Rel}(g_{1:t})] \leq \mathbf{Rel}(g_{1:t-1})$$

holds for each timestep t , and further that the inequality is certified by the strategy of [Algorithm 3](#). We begin by expanding the definition of \mathbf{Rel} :

$$\begin{aligned} & \inf_{P_t \in \Delta(\Delta_N)} \sup_{g_t \in \mathcal{C}} \mathbb{E}_{P_t} \mathbb{E}_{i_t \sim P_t} [\langle e_{i_t}, g_t \rangle + \mathbf{Rel}(g_{1:t})] \\ &= \inf_{P_t \in \Delta(\Delta_N)} \sup_{g_t \in \mathcal{C}} \mathbb{E}_{P_t} \mathbb{E}_{i_t \sim P_t} \left[\langle e_{i_t}, g_t \rangle + \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{i \in [N]} \left[- \sum_{s=1}^t \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right] \right]. \end{aligned}$$

Now plug in the randomized strategy given by [Algorithm 3](#), with $\mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N}$ taking the place of $\mathbb{E}_{p_t \sim P_t}$:

$$\leq \sup_{g_t \in \mathcal{C}} \left[\mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \left[\mathbb{E}_{i_t \sim p_t(\sigma_{t+1:n})} \langle e_{i_t}, g_t \rangle \right] + \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{i \in [N]} \left[- \sum_{s=1}^t \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right] \right].$$

Grouping expectations and applying Jensen's inequality:

$$\leq \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{g_t \in \mathcal{C}} \left[\mathbb{E}_{i_t \sim p_t(\sigma_{t+1:n})} \langle e_{i_t}, g_t \rangle + \sup_{i \in [N]} \left[- \sum_{s=1}^t \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right] \right].$$

Expanding the definition of p_t (using its optimality in particular):

$$= \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \inf_{p_t \in \Delta_N} \sup_{g_t \in \mathcal{C}} \left[\langle p_t, g_t \rangle + \sup_{i \in [N]} \left[- \sum_{s=1}^t \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right] \right].$$

Now we go through the fairly standard sequential Rademacher comexplecity symmetrization procedure. Begin by using the minimax theorem to swap the order of \inf_{p_t} and \sup_{g_t} . To do so, we allow the g_t player to randomize, and denote their distribution by $Q_t \in \Delta(\mathcal{C})$.

$$= \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{Q_t \in \Delta(\mathcal{C})} \inf_{p_t \in \Delta_N} \mathbb{E}_{g_t \sim Q_t} \left[\langle p_t, g_t \rangle + \sup_{i \in [N]} \left[- \sum_{s=1}^t \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right] \right].$$

Since the supremum over i does not directly depend on p_t , we can rewrite this expression by introducing a (conditionally) IID copy of g_t , called g'_t :

$$= \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{Q_t \in \Delta(\mathcal{C})} \inf_{p_t \in \Delta_N} \mathbb{E}_{g_t \sim Q_t} \left[\sup_{i \in [N]} \left[\mathbb{E}_{g'_t \sim Q_t} [\langle p_t, g'_t \rangle] - \sum_{s=1}^t \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right] \right].$$

Choosing p_t to match e_i :

$$\leq \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{Q_t \in \Delta(\mathcal{C})} \mathbb{E}_{g_t \sim Q_t} \sup_{i \in [N]} \left[\mathbb{E}_{g'_t \sim Q_t} [\langle e_i, g'_t \rangle] - \langle e_i, g_t \rangle - \sum_{s=1}^{t-1} \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right].$$

Applying Jensen's inequality:

$$\leq \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{Q_t \in \Delta(\mathcal{C})} \mathbb{E}_{g_t, g'_t \sim Q_t} \sup_{i \in [N]} \left[\langle e_i, g'_t \rangle - \langle e_i, g_t \rangle - \sum_{s=1}^{t-1} \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right].$$

At this point, we may introduce a new Rademacher random variable ϵ_t without changing the distribution of $g'_t - g_t$, thereby not changing the value of the game:

$$\begin{aligned} &= \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{Q_t \in \Delta(\mathcal{C})} \mathbb{E}_{\epsilon_t \in \{\pm 1\}} \mathbb{E}_{g_t, g'_t \sim Q_t} \sup_{i \in [N]} \left[\epsilon_t \langle e_i, g'_t - g_t \rangle - \sum_{s=1}^{t-1} \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right] \\ &\leq \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{Q_t \in \Delta(\mathcal{C})} \mathbb{E}_{\epsilon_t \in \{\pm 1\}} \mathbb{E}_{g_t, g'_t \sim Q_t} \left\{ \begin{aligned} &\sup_{i \in [N]} \left[\epsilon_t \langle e_i, g'_t \rangle + \frac{1}{2} \left(- \sum_{s=1}^{t-1} \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right) \right] \\ &+ \sup_{i \in [N]} \left[\epsilon_t \langle e_i, -g_t \rangle + \frac{1}{2} \left(- \sum_{s=1}^{t-1} \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right) \right] \end{aligned} \right\} \\ &= \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{Q_t \in \Delta(\mathcal{C})} \mathbb{E}_{\epsilon_t \in \{\pm 1\}} \mathbb{E}_{g_t \sim Q_t} \sup_{i \in [N]} \left[2\epsilon_t \langle e_i, g_t \rangle - \sum_{s=1}^{t-1} \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right] \end{aligned}$$

The above expression is now linear in Q_t , so it may be replaced with a pure strategy:

$$= \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{g_t \in \mathcal{C}} \mathbb{E}_{\epsilon_t \in \{\pm 1\}} \sup_{i \in [N]} \left[2\epsilon_t \langle e_i, g_t \rangle - \sum_{s=1}^{t-1} \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right]$$

This expression is also convex in g_t , which means that the supremum will be obtained at a vertex of \mathcal{C} :

$$= \mathbb{E}_{\sigma_{t+1:n} \in \{\pm 1\}^N} \sup_{\sigma_t \in \{\pm 1\}^N} \mathbb{E}_{\epsilon_t \in \{\pm 1\}} \sup_{i \in [N]} \left[2\epsilon_t \sigma_t[i] c_i - \sum_{s=1}^{t-1} \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right]$$

Now apply [Theorem 10](#) conditioned on $\sigma_{t+1:n}$, with $w_i = - \sum_{s=1}^{t-1} \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i)$.

$$\begin{aligned} &\leq \mathbb{E}_{\sigma_{t:n} \in \{\pm 1\}^N} \sup_{i \in [N]} \left[- \sum_{s=1}^{t-1} \langle e_i, g_s \rangle + 4 \sum_{s=t}^n \sigma_s[i] c_i - B(i) \right] \\ &= \mathbf{Rel}(g_{1:t-1}). \end{aligned}$$

Final Value The final value of the relaxation is

$$\mathbf{Rel}(\cdot) = 2 \mathbb{E}_{\sigma_{1:n} \in \{\pm 1\}^N} \sup_{i \in [N]} \left[2 \sum_{t=1}^n \sigma_t[i] c_i - 5c_i \sqrt{n(\log(1/\pi_i) + \log(4c_i^2 n))} \right] \leq 2 \sum_{i \in [N]} \frac{\pi_i}{4c_i^2 n} \leq 1.$$

To get the final bound, we have applied a maximal inequality, [Lemma 2](#) by recognizing that the process $\mathbf{Rel}(\cdot)$ has the form of a supremum of a random process. Namely, we can write $\mathbf{Rel}(\cdot)$ in terms of $\mathbb{E} \sup_{i \in [N]} \{X_i - B(i)\}$ with $X_i = 2 \sum_{t=1}^n \sigma_t[i] c_i$. The standard mgf bound of $\mathbb{E} e^{\lambda X} \leq e^{\lambda^2 (b-a)^2 / 8}$ for mean-zero random variables X with $a \leq X \leq b$ [\[6\]](#), along with independence of the Rademacher random variables in X_i , implies an mgf bound of

$$\mathbb{E} e^{\lambda X_i} \leq e^{2c_i^2 \lambda^2 n}.$$

Thus, to prove the result it suffices to take $h_i = 4c_i^2 n$ and $p = 2$ in the statement [Lemma 2](#) and note that $B(i) \geq (2 + 1/p)h_i^{1/p}(\log(h_i) + \log(1/\pi_i))^{1-1/p}$, the quantity appearing in the statement of [Lemma 2](#). The only additional detail to verify is that, since it was assumed that $c_i \geq 1$ for all i and since $n \geq 1$ by definition, the condition $h_i/\pi_i \geq e$ required by [Lemma 2](#) is satisfied.

Computational Efficiency We briefly state how the min-max optimization problem in the learner's strategy can be computed efficiently. Recall that the optimization problem is

$$\begin{aligned} & \min_{p \in \Delta_N} \sup_{g_t: |g_t[i]| \leq c_i} \left[\langle p, g_t \rangle + \sup_{i \in [N]} \left[- \sum_{s=1}^t \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right] \right] \\ & = \min_{p \in \Delta_N} \sup_{i \in [N]} \sup_{g_t: |g_t[i]| \leq c_i} \left[\langle p, g_t \rangle - \sum_{s=1}^t \langle e_i, g_s \rangle + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right] \end{aligned}$$

Since the inner quantity is linear in g_t and there are no interactions between coordinates, we can verify that conditioned on i the max over g_t is obtained via (letting $G_{t-1}(i) = \sum_{s=1}^{t-1} g_s[i]$)

$$\begin{aligned} & = \min_{p \in \Delta_N} \sup_{i \in [N]} \left[\langle p, c \rangle + (1 - 2p[i])c_i - G_{t-1}(i) + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i) \right] \\ & = \min_{p \in \Delta_N} \sup_{i \in [N]} [\langle p, c \rangle + \langle a, e_i \rangle - 2\langle p, \text{diag}(c)e_i \rangle], \end{aligned}$$

where $a[i] = c_i - G_{t-1}(i) + 4 \sum_{s=t+1}^n \sigma_s[i] c_i - B(i)$. We can now employ a standard reduction from saddle point optimization to linear programming, i.e.

$$\begin{aligned} & \text{minimize} && \langle p, c \rangle + s \\ & \text{subject to} && s \geq \langle a, e_i \rangle - 2\langle p, \text{diag}(c)e_i \rangle \quad \forall i. \\ & && p \in \Delta_N. \end{aligned}$$

Assuming that $\min_i c_i \geq 1$, this linear program can be solved to accuracy ϵ by interior point methods (e.g. [\[36\]](#)) in time $O(N^{3.5} \log(\epsilon^{-1} \max_i c_i))$ or by Mirror-Prox [\[29\]](#) in time $O(N\epsilon^{-1} \max_i c_i)$. Since our rates scale as \sqrt{n} we can set $\epsilon = 1/(\sqrt{n} \max_i c_i)$ to conclude the result. \square

Theorem 10. For any $w \in \mathbb{R}^N$, any $c \in \mathbb{R}_+^N$,

$$\sup_{\sigma \in \{\pm 1\}^N} \mathbb{E} \max_{\epsilon \in \{\pm 1\}} \max_{i \in [N]} \{w_i + 2\epsilon \sigma_i c_i\} \leq \mathbb{E} \max_{\sigma \in \{\pm 1\}^N} \max_{i \in [N]} \{w_i + 4\sigma_i c_i\}. \quad (11)$$

Proof of Theorem 10. Fix any $\sigma \in \{\pm 1\}^N$. Let $i_1 = \arg \max_{i \in [N]} \{w_i + 2\sigma_i c_i\}$ and $i_{-1} = \arg \max_{i \in [N]} \{w_i - 2\sigma_i c_i\}$. Then it is easy to see that

$$\mathbb{E} \max_{i \in [N]} \{w_i + 2\epsilon \sigma_i c_i\} = \mathbb{E} \max_{\epsilon \in \{\pm 1\}} \max_{i \in \{i_1, i_{-1}\}} \{w_i + 2\epsilon \sigma_i c_i\} \leq \mathbb{E} \max_{\sigma' \in \{\pm 1\}^N} \max_{i \in \{i_1, i_{-1}\}} \{w_i + 4\sigma'_i c_i\} \leq \mathbb{E} \max_{\sigma' \in \{\pm 1\}^N} \max_{i \in [N]} \{w_i + 4\sigma'_i c_i\}.$$

The central inequality above follows by [Lemma 1](#) with the pair $(w, 2c)$. Since the above bound holds for any σ , we conclude that [\(11\)](#) holds. \square

Lemma 1. For any pair (w, c) , the inequality

$$\sup_{\sigma \in \{\pm 1\}^N} \mathbb{E} \max_{\epsilon \in \{\pm 1\}} \max_{i \in [N]} \{w_i + \epsilon \sigma_i c_i\} \leq \mathbb{E} \max_{\sigma \in \{\pm 1\}^N} \max_{i \in [N]} \{w_i + 2\sigma_i c_i\}. \quad (12)$$

holds when $N = 2$.

Proof of Lemma 1. In this proof we adopt the notation that for any element $j \in [2]$, $-j$ denote the other element. Say the pair (w, c) is *dominated* if there exists j for which $w_j - c_j \geq w_{-j} + c_{-j}$. Note that this of course implies $w_j + c_j \geq w_{-j} + c_{-j}$ as well, since c is non-negative.

Dominated case Suppose (w, c) is dominated by index j . Then (12) holds trivially for any $K \in \mathbb{R}$ by

$$\sup_{\sigma \in \{\pm 1\}^N} \mathbb{E} \max_{\epsilon \in \{\pm 1\}} \max_{i \in [N]} \{w_i + \epsilon \sigma_i c_i\} = w_j = \max_{i \in [N]} \{w_i + K \mathbb{E} \max_{\sigma \in \{\pm 1\}^N} \sigma_i c_i\} \leq \mathbb{E} \max_{\sigma \in \{\pm 1\}^N} \max_{i \in [N]} \{w_i + K \sigma_i c_i\}.$$

We now focus on the trickier “not dominated” case.

Rescaling doesn’t induce domination We first observe that if (w, c) does is not dominated, (w, Bc) is not dominated either for any $B \geq 1$. Let j be the index for which $w_j + c_j \geq w_{-j} + c_{-j}$ which implies $w_j - c_j \leq w_{-j} + c_{-j}$ because (w, c) is not dominated. Observe that if (w, Bc) is dominated we either have $w_j - Bc_j \geq w_{-j} + Bc_{-j}$ or $w_{-j} - Bc_{-j} \geq w_j + Bc_j$. The first case cannot hold because $B \geq 1$ and we already know that (w, c) is not dominated. The second case in particular implies $w_{-j} \geq w_j$, so we must have had $c_j \geq c_{-j}$ to begin with. But in that case we will still have $w_j + Bc_j \geq w_{-j} + Bc_{-j}$ which contradicts the domination.

Note: It is good to keep in mind that while rescaling does not induce domination, it may not be the case in general that $w_j + Bc_j \geq w_{-j} + Bc_{-j}$ even though $w_j + c_j \geq w_{-j} + c_{-j}$. That is, the “leader” may change after rescaling.

LHS of (12) for (w, c) not dominated When (w, c) is not dominated we have

$$\sup_{\sigma \in \{\pm 1\}^N} \mathbb{E} \max_{\epsilon \in \{\pm 1\}} \max_{i \in [N]} \{w_i + \epsilon \sigma_i c_i\} = \frac{1}{2}(w_1 + c_1) + \frac{1}{2}(w_2 + c_2).$$

RHS of (12) for (w, c) not dominated We will consider the RHS of (12) for $(w, c') \triangleq (w, Bc)$ for some $B \geq 1$ to be decided. By the argument above, the pair (w, c') is also not dominated. For the remainder of the proof, 1 will denote the index for which $w_1 + c'_1 \geq w_2 + c'_2$. Because the pair is not dominated, the value the RHS takes can be classified into two cases based on the relationship between c' and w .

- Case 1: $w_1 - c'_1 \leq w_2 - c'_2$:

In this case there is equal probability that the process takes on value $w_2 - c'_2$ or $w_2 + c'_2$ conditioned on the event that $\sigma_1 = -1$, so we have the equality:

$$\mathbb{E} \max_{\sigma \in \{\pm 1\}^N} \max_{i \in [N]} \{w_i + \sigma_i c'_i\} = \frac{1}{2}(w_1 + w_2) + \frac{1}{2}c'_1$$

Furthermore, Case 1 implies $c'_1 \geq c'_2$, which leads to an inequality:

$$\geq \frac{1}{2}(w_1 + w_2) + \frac{1}{4}(c'_1 + c'_2).$$

- Case 2: $w_1 - c'_1 \geq w_2 - c'_2$:

In this case, conditioned on the event that $\sigma_1 = -1$, there is equal probability that the process takes on value $w_2 + c'_2$ or $w_1 - c'_1$, so the equality becomes:

$$\mathbb{E} \max_{\sigma \in \{\pm 1\}^N} \max_{i \in [N]} \{w_i + \sigma_i c'_i\} = \frac{1}{2}(w_1 + c'_1) + \frac{1}{4}(w_2 + c'_2) + \frac{1}{4}(w_1 - c'_1)$$

Case 2 implies that $w_1 \geq w_2$, because we may add the inequalities $w_1 + c'_1 \geq w_2 + c'_2$ and $w_1 - c'_1 \geq w_2 - c'_2$. This gives an inequality:

$$\geq \frac{1}{2}(w_1 + w_2) + \frac{1}{4}(c'_1 + c'_2).$$

Combining our results for the two cases, we have that for any vector c' , so long as (w, c') is not dominated,

$$\mathbb{E}_{\sigma \in \{\pm 1\}^N} \max_{i \in [N]} \{w_i + \sigma_i c'_i\} \geq \frac{1}{2}(w_1 + w_2) + \frac{1}{4}(c'_1 + c'_2).$$

In particular, choosing $B = 2$ implies (12) in the non-dominated case:

$$\begin{aligned} \mathbb{E}_{\sigma \in \{\pm 1\}^N} \max_{i \in [N]} \{w_i + 2\sigma_i c_i\} &\geq \frac{1}{2}(w_1 + w_2) + \frac{1}{2}(c_1 + c_2) \\ &= \sup_{\sigma \in \{\pm 1\}^N} \mathbb{E} \max_{i \in [N]} \{w_i + \epsilon \sigma_i c_i\}. \end{aligned}$$

Final result Combining the dominated and non-dominated results we have that for any (w, c) .

$$\sup_{\sigma \in \{\pm 1\}^N} \mathbb{E} \max_{i \in [N]} \{w_i + \epsilon \sigma_i c_i\} \leq \mathbb{E} \max_{i \in [N]} \{w_i + 2\sigma_i c_i\}.$$

□

Lemma 2 (Multi-scale maximal inequality). Let $(X_i)_{i \in [N]}$ be a real-valued random process for which there exists a sequence $(h_i)_{i \in [N]}$ with $h_i > 0$ such that the moment generating function bound $\mathbb{E} e^{\lambda X_i} \leq e^{\lambda^p h_i}$ is satisfied for all $\lambda > 0$ and some choice of $p > 0$. Then for any distribution $\pi \in \Delta_N$ for which $h_i/\pi_i \geq e$ for all $i \in [N]$ it holds that

$$\mathbb{E} \sup_{i \in [N]} \left\{ X_i - (2 + 1/p) h_i^{1/p} (\log(h_i) + \log(1/\pi_i))^{1-1/p} \right\} \leq \sum_{i \in [N]} \frac{\pi_i}{h_i}. \quad (13)$$

Proof. Let $B(i) = C h_i^{1/p} (\log(h_i) + \log(1/\pi_i))^{1-1/p}$ for some constant C to be decided later. One should verify that $\log(h_i) + \log(1/\pi_i)$ is always non-negative by the assumption that $h_i/\pi_i \geq e$, which will be used repeatedly. To begin, observe that

$$\mathbb{E} \sup_{i \in [N]} \{X_i - B(i)\} \leq \mathbb{E} \sup_{i \in [N]} [X_i - B(i)]_+,$$

where $[x]_+ = \max\{x, 0\}$. By non-negativity of $[x]_+$ it further holds that

$$\leq \mathbb{E} \sum_{i \in [N]} [X_i - B(i)]_+.$$

Fixing an arbitrary sequence $(\lambda_i)_{i \in \mathbb{N}}$ with $\lambda_i > 0$, the basic inequality $\max\{a, b\} \leq \frac{1}{\lambda} \log(e^{\lambda a} + e^{\lambda b})$ implies the following upper bound:

$$\leq \mathbb{E} \sum_{i \in [N]} \frac{1}{\lambda_i} \log(1 + e^{\lambda_i (X_i - B(i))}).$$

Apply Jensen's inequality:

$$\leq \sum_{i \in [N]} \frac{1}{\lambda_i} \log(1 + \mathbb{E} e^{\lambda_i (X_i - B(i))}).$$

Now use the moment bound assumed in the lemma statement:

$$\leq \sum_{i \in [N]} \frac{1}{\lambda_i} \log(1 + e^{(\lambda_i^p h_i - \lambda_i B(i))}).$$

Lastly, apply the inequality $\log(1 + x) \leq x$ for $x \geq 0$:

$$\leq \sum_{i \in [N]} \exp(\lambda_i^p h_i - \lambda_i B(i) + \log(1/\lambda_i)).$$

We now take $\lambda_i = \left(\frac{\log(h_i) + \log(1/\pi_i)}{h_i} \right)^{1/p}$ and bound each exponent in the sum above. Using the definition of $B(i)$:

$$\lambda_i^p h_i - \lambda_i B(i) + \log(1/\lambda_i) = \log(1/\lambda_i) - (C - 1)(\log(1/\pi_i) + \log(h_i)).$$

Next observe that

$$\log(1/\lambda_i) = \frac{1}{p} \log\left(\frac{h_i}{\log(h_i/\pi_i)}\right) \leq \frac{1}{p} \log(h_i),$$

where we have used that $h_i/\pi_i \geq e$. With this, and using that $\log(1/\pi_i) \geq 0$, we have

$$\lambda_i^p h_i - \lambda_i B(i) + \log(1/\lambda_i) \leq -(C-1-1/p)(\log(1/\pi_i) + \log(h_i)).$$

Taking $C \geq 2 + 1/p$ and using this bound in the summation over i yields the result:

$$\mathbb{E} \sup_{i \in [N]} \{X_i - B(i)\} \leq \sum_{i \in [N]} \frac{\pi_i}{h_i}.$$

□

A.2 Proofs for Section 2.2

Mirror Descent Online mirror descent is the standard algorithm for online linear optimization over convex sets. It is parameterized by a convex set \mathcal{W} , learning rate η , and strongly convex regularizer $\mathcal{R} : \mathcal{W} \rightarrow \mathbb{R}$. We define the update $\text{MIRRORDESCENT}(\eta, \mathcal{W}, \mathcal{R})$ as follows.

First, set $w_1 = \arg \min_{w \in \mathcal{W}} \mathcal{R}(w)$. Then, for each time $t \in [n]$:

- Receive gradient g_t and let \tilde{w}_{t+1} satisfy $\nabla \mathcal{R}(\tilde{w}_{t+1}) = \nabla \mathcal{R}(w_t) - \eta g_t$.
- Set $w_{t+1} = \arg \min_{w \in \mathcal{W}} \mathcal{D}_{\mathcal{R}}(w | \tilde{w}_{t+1})$.

Fact 1 (Mirror Descent (e.g. [16])). Let (w_t) be the iterates produced by $\text{MIRRORDESCENT}(\eta, \mathcal{W}, \mathcal{R})$ on a sequence of vectors $(g_t)_{t \leq n}$. If \mathcal{R} is λ -strongly convex with respect to a norm $\|\cdot\|_{\mathcal{R}}$, the iterates satisfy

$$\sum_{t=1}^n \langle w_t - w, g_t \rangle \leq \frac{\eta}{2\lambda} \sum_{t=1}^n \|g_t\|_{\mathcal{R},*}^2 + \frac{1}{\eta} \mathcal{R}(w) \quad \forall w \in \mathcal{W}. \quad (14)$$

Proof of Theorem 2. First, we verify that the loss sequence $(g_t)_{t \leq n}$ is such that the regret bound derived for MULTISCALEFTPL applies. In particular, we need to verify that $|g_t[i]| \leq c_i$ for each i . To this end, fix an index $i \in [N]$, and note that since f_t is L_i -Lipschitz on \mathcal{W}_i with respect to the norm $\|\cdot\|_{(i)}$ we have

$$|g_t[i]| = |f_t(w_t^i) - f_t(0)| \leq L_i \|w_t^i - 0\|_{(i)} \leq L_i R_i \leq L_i R_i = c_i,$$

as required. Also, it was assumed that $c_i = L_i R_i \geq 1$, as required for [Theorem 1](#).

Now, recall that (p_t) is the sequence of distributions produced by the meta-algorithm. The algorithm's total loss with respect to the centered iterates (\tilde{f}_t) is given by

$$\sum_{t=1}^n \tilde{f}_t(w_t^{i_t}) = \sum_{t=1}^n \langle e_{i_t}, g_t \rangle,$$

where this equality is due to the construction of the losses $(g_t)_{t \leq n}$ given to MULTISCALEFTPL . The regret bound for MULTISCALEFTPL now implies that

$$\mathbb{E} \left[\sum_{t=1}^n \langle e_{i_t}, g_t \rangle - \min_{i \in [N]} \left\{ \sum_{t=1}^n g_t[i] + O\left(R_i L_i \sqrt{n \log(R_i L_i n / \pi_i)}\right) \right\} \right] \leq 0,$$

where we have obtained this inequality by substituting the value of the vector c constructed by MULTISCALEOCO into the regret bound (4) for MULTISCALEFTPL . Now, observe that for each i we have

$$\sum_{t=1}^n g_t[i] = \sum_{t=1}^n \tilde{f}_t(w_t^i) \leq \inf_{w \in \mathcal{W}_i} \sum_{t=1}^n \tilde{f}_t(w) + \mathbf{Reg}_n(i),$$

where we have used the definition of g_t and the regret bound assumed on the sub-algorithm. Combining these inequalities, we have

$$\mathbb{E} \left[\sum_{t=1}^n \tilde{f}_t(w_t^{i_t}) - \min_{i \in [N]} \left\{ \inf_{w \in \mathcal{W}_i} \sum_{t=1}^n \tilde{f}_t(w) + \mathbf{Reg}_n(i) + O\left(R_i L_i \sqrt{n \log(R_i L_i n / \pi_i)}\right) \right\} \right] \leq 0.$$

Finally, observe that since $\widetilde{f}_t(w) = f_t(w) - f_t(0)$, the above is equivalent to

$$\mathbb{E} \left[\sum_{t=1}^n f_t(w_t^{i_t}) - \min_{i \in [N]} \left\{ \inf_{w \in \mathcal{W}_i} \sum_{t=1}^n f_t(w) + \mathbf{Reg}_n(i) + O\left(R_i L_i \sqrt{n \log(R_i L_i n / \pi_i)}\right) \right\} \right] \leq 0.$$

□

Proof of Theorem 3. Recall that each sub-algorithm ALG_i runs Mirror Descent over a ball in $(\mathfrak{B}, \|\cdot\|)$ of radius R_i using the regularizer $\frac{1}{2} \|\cdot\|^2$. From the regret bound for Mirror Descent (Fact 1), the meta-algorithm's choice of Mirror Descent parameters for ALG_i guarantees that

$$\sum_{t=1}^n f_t(w_t^i) - \inf_{w \in \mathcal{W}_i} \sum_{t=1}^n f_t(w) \leq O(R_i \sqrt{n/\lambda}).$$

Combined with the regret bound for MULTISCALEOCO (Theorem 2, noting that $R_i L_i \geq 1$), this implies that the meta-algorithm's regret satisfies

$$\mathbb{E} \left[\sum_{t=1}^n f_t(w_t^{i_t}) - \min_{i \in [N]} \left\{ \inf_{w \in \mathcal{W}_i} \sum_{t=1}^n f_t(w) + O(R_i L \sqrt{n/\lambda}) + O\left(R_i L \sqrt{n \log(R_i L n / \pi_i)}\right) \right\} \right] \leq 0.$$

Which further implies

$$\mathbb{E} \left[\sum_{t=1}^n f_t(w_t^{i_t}) - \min_{i \in [N]} \left\{ \inf_{w \in \mathcal{W}_i} \sum_{t=1}^n f_t(w) + O\left(R_i L \sqrt{n \log(R_i L n / \lambda)}\right) \right\} \right] \leq 0.$$

Now, recall that $i \in [n+1]$, and that $R_i = 2^{i-1}$. Consider the algorithm's regret against a comparator w . For now, assume that w satisfies $1 \leq \|w\| \leq e^n$ — we will see shortly that this is without loss of generality. Let $i^*(w) = \min\{i \mid w \in \mathcal{W}_i\}$. Then the regret bound above implies

$$\mathbb{E} \left[\sum_{t=1}^n f_t(w_t^{i_t}) - \left\{ \sum_{t=1}^n f_t(w) + O\left(R_{i^*(w)} L \sqrt{n \log(R_{i^*(w)} L n / \lambda)}\right) \right\} \right] \leq 0.$$

Furthermore, since $R_i = 2^{i-1}$, we have that $R_{i^*(w)} \leq 2\|w\|$, and so

$$\mathbb{E} \left[\sum_{t=1}^n f_t(w_t^{i_t}) - \left\{ \sum_{t=1}^n f_t(w) + O\left(\|w\| L \sqrt{n \log(\|w\| L n / \lambda)}\right) \right\} \right] \leq 0.$$

This is exactly the regret bound we wanted. Now, the case where $\|w\| \leq 1$ is handled by simply noting $i^*(w) = 1$ and writing $R_1 = 1 \leq 1 + \|w\|$, which gives the $\|w\| + 1$ factor as follows:

$$\mathbb{E} \left[\sum_{t=1}^n f_t(w_t^{i_t}) - \left\{ \sum_{t=1}^n f_t(w) + O\left((\|w\| + 1) L \sqrt{n \log((\|w\| + 1) L n / \lambda)}\right) \right\} \right] \leq 0.$$

To handle the case where $\|w\| \geq e^n$, we appeal to Corollary 1 with $c = L\sqrt{n}$ and $\gamma = 1/2$, which shows that it suffices to consider $\|w\| \leq \exp\left(\left(\frac{Ln}{c}\right)^{1/\gamma}\right) = e^n$. Note that the constants appearing in the regret bound above, both inside the $O(\cdot)$ and inside the $\sqrt{\log(\cdot)}$ are worse than those with which we instantiate Corollary 1. This is not an issue because worse constants only reduce the radius that must be considered in the corollary.

□

Lemma 3. Let $F : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ be given. Suppose the loss sequence $(f_t)_{t \leq n}$ is L -Lipschitz with respect to $\|\cdot\|_*$. Then a regret bound of the form

$$\sum_{t=1}^n f_t(w_t) - \sum_{t=1}^n f_t(w) \leq F(\|w\|) \quad \forall w \in \mathfrak{B} \quad (15)$$

holds if the restricted regret bound

$$\sum_{t=1}^n f_t(w_t) - \sum_{t=1}^n f_t(w) \leq F(\|w\|) \quad \forall f : \|f\| \leq \alpha^*, \quad (16)$$

holds, where α^* is the greatest non-negative number for which $F(\alpha^*) - \alpha^* Ln \geq F(0)$.

Proof of Lemma 3. Assume wlog that $f_t(0) = 0$ for each t . This is possible because

$$\sum_{t=1}^n f_t(w_t) - \sum_{t=1}^n f_t(w) = \sum_{t=1}^n (f_t(w_t) - f_t(0)) - \sum_{t=1}^n (f_t(w) - f_t(0)).$$

To begin, observe that (15) is equivalent to

$$\sum_{t=1}^n f_t(w_t) \leq \inf_{w \in \mathfrak{B}} \left\{ \sum_{t=1}^n f_t(w) + F(\|w\|) \right\}.$$

By selecting $w = 0$, $f_t(0) = 0$ implies that the infimum on the right is always upper bounded in value by $F(0)$. In the other direction, Lipschitzness of the losses along with $f_t(0) = 0$ implies that the infimum is lower bounded as

$$\inf_{w \in \mathfrak{B}} \left\{ \sum_{t=1}^n f_t(w) + F(\|w\|) \right\} \geq \inf_{w \in \mathfrak{B}} \{-L\|w\|n + F(\|w\|)\} = \inf_{\alpha \geq 0} \{-\alpha Ln + F(\alpha)\}.$$

Therefore if $\alpha \geq \alpha^*$, the lower bound $-\alpha Ln + F(\alpha)$ will be sub-optimal compared to the upper bound of $F(0)$ obtained by choosing $\alpha = 0$. \square

Corollary 1. When $F(r) = c \cdot (r + 1) \log(r + 1)^\gamma$ for $\gamma > 0$, it is sufficient to consider

$$\sum_{t=1}^n f_t(w_t) - \sum_{t=1}^n f_t(w) \leq F(\|w\|) \quad \forall w : \|w\| \leq \exp\left(\left(\frac{Ln}{c}\right)^{1/\gamma}\right). \quad (17)$$

Proof of Corollary 1. Note that $F(0) = 0$. Let r denote the minimizer of $F(\alpha) - \alpha \cdot a$ (where $a = Ln$). Differentiating this expression yields

$$a = c(\log(r + 1)^\gamma + \gamma \log(r + 1)^{\gamma-1}),$$

which further implies

$$\log(r + 1)^\gamma = \frac{a}{c} \cdot \frac{1}{1 + \gamma/\log(r + 1)} \leq \frac{a}{c}.$$

Rearranging, we have $r \leq \exp((a/c)^{1/\gamma}) - 1$. Since $F(\alpha) - \alpha \cdot a$ is strictly convex, this function is increasing above r . To conclude, we guess the value of α^* to be $\alpha^* = \exp((a/c)^{1/\gamma}) - 1$. Substituting this value in, we have

$$F(\alpha) - \alpha \cdot a \geq a \exp((a/c)^{1/\gamma}) - a \cdot \exp((a/c)^{1/\gamma}) = 0 = F(0),$$

which yields the result. \square

Proof of Theorem 4. We only sketch the details of this proof as it follows Theorem 3 very closely.

We now describe sub-algorithm configuration for MULTISCALEOCO that achieves the claimed regret bound. Our strategy will be to take a discretization the range of p values $(1 + \delta, 2]$, and produce a set of sub-algorithms for each p in this discrete set. For a fixed p , the construction of the set of sub-algorithms will be exactly is in Theorem 3. The discrete set of p s will have the form $p_k = 1 + \delta + (k - 1) \cdot \epsilon$, for $\epsilon = 1/\log(d)$ and $k \in [1, \dots, K]$, where $K = \lceil (1 - \delta)/\epsilon \rceil$ (in particular $k \leq \log(d)$).

For a fixed k , the norm $\|\cdot\|_{p_k}$ has that $\frac{1}{2}\|\cdot\|_{p_k}^2$ is $2(p_k - 1)$ -strongly convex with respect to itself [20]. With this in mind, we create a set of $N = K(n + 1)$ sub-algorithms, which we will index by pairs $(k, j) \in [K] \times [n + 1]$ instead of $i \in [K(n + 1)]$ for notational convenience.

- For each $k \in [K]$:
 - For each $j \in \{1, \dots, n + 1\}$:
 - * Set $R_j = 2^{j-1}$, $L_j = L_{p_k}$.
 - * Take $\mathcal{W}_{(k,j)} = \{w \in \mathfrak{B} \mid \|w\|_{p_k} \leq R_j\}$, $\eta_{(k,j)} = \sqrt{\lambda_{p_k} \frac{R_j}{Ln}}$, where $\lambda_{p_k} = 2(p_k - 1)$.
 - * Let $\text{ALG}_j = \text{MIRRORDESCENT}(\eta_{(k,j)}, \mathcal{W}_{(k,j)}, \|\cdot\|_{p_k}^2)$.

- $\pi = \text{Uniform}([K] \times [n+1])$.

Clearly the total number of sub-algorithms and hence the running time scales as $n \cdot \log(d)$.

Referring back to the proof of [Theorem 3](#), and letting (k_t, j_t) denote the index pair chosen by MULTISCALEOCO in round t , it is clear that for a fixed k , the algorithm satisfies

$$\mathbb{E} \left[\sum_{t=1}^n f_t(w_t^{(k_t, j_t)}) - \left\{ \sum_{t=1}^n f_t(w) + O\left((\|w\|_{p_k} + 1) L_{p_k} \sqrt{n \log((\|w\|_{p_k} + 1) L_{p_k} n \log(d)) / (p_k - 1)} \right) \right\} \right] \leq 0.$$

In fact, the regret guarantee for MULTISCALEOCO implies that

$$\mathbb{E} \left[\sum_{t=1}^n f_t(w_t^{(k_t, j_t)}) - \min_{k \in [N]} \left\{ \sum_{t=1}^n f_t(w) + O\left((\|w\|_{p_k} + 1) L_{p_k} \sqrt{n \log((\|w\|_{p_k} + 1) L_{p_k} n \log(d)) / (p_k - 1)} \right) \right\} \right] \leq 0.$$

We now appeal to the choice of discretization to deduce that

$$\mathbb{E} \left[\sum_{t=1}^n f_t(w_t^{(k_t, j_t)}) - \min_{p \in (1+\delta, 2]} \left\{ \sum_{t=1}^n f_t(w) + O\left((\|w\|_{p_k} + 1) L_p \sqrt{n \log((\|w\|_{p_k} + 1) L_p \log(d) n) / (p - 1)} \right) \right\} \right] \leq 0.$$

Namely, for each k , $\|w\|_{p_k} / \|w\|_{p_{k+1}} \leq d^\epsilon = d^{1/\log(d)} = O(1)$. \square

Proof of [Theorem 5](#). For the online PCA task, we predict from a class $\mathcal{W}_k = \{W \in \mathbb{S}_+ \cap \mathbb{B}_\sigma \mid \langle W, I \rangle = k\}$, where \mathbb{B}_σ denotes the spectral norm ball. For a fixed value of k such a class is a convex relaxation of the set of all rank k projection matrices. We predict W_t and experience linear loss functions $f_t(W_t) = \langle I - W_t, Y_t \rangle$, where $Y_t \in \mathcal{Y}_\sigma = \mathbb{S}_+ \cap \mathbb{B}_\sigma$.

The regret for this game is given by

$$\sup_{W \in \mathcal{W}_k} \left[\sum_{t=1}^n \langle I - W_t, Y_t \rangle - \sum_{t=1}^n \langle I - W, Y_t \rangle \right]. \quad (18)$$

From [\[30\]](#), we have that for fixed k the strategy MATRIX EXPONENTIATED GRADIENT has regret bounded by

$$\min \left\{ \sqrt{nk^2 \log(n/k)}, \sqrt{n(d-k)^2 \log(n/(d-k))} \right\} = \tilde{O} \left(\sqrt{n \min\{k, d-k\}^2} \right).$$

Note: The variant of MATRIX EXPONENTIATED GRADIENT that obtains this strategy uses either losses or gains depending on the value of k .

The meta-algorithm strategy we follow will be

- For each $i \in [\lceil \log(d/2) \rceil + 1]$:
 - Set $R_i = 2^{i-1}$, $L_i = 1$.
 - $\mathcal{W}_i = \{W \in \mathbb{S}_+ \cap \mathbb{B}_\sigma \mid \langle W, I \rangle = R_i\}$
 - Take $\text{ALG}_i = \text{MATRIX EXPONENTIATED GRADIENT}(\mathcal{W}_i)$ as described in [\[30\]](#).
- $\pi = \text{Uniform}([\lceil \log(d/2) \rceil + 1])$.

All that remains is that the losses of the sub-algorithms satisfy the claimed upper bound R_i . Observe since MULTISCALEOCO works with centered loss $\tilde{f}_t(W) = -\langle W, Y_t \rangle$. For any $W \in \mathcal{W}_k$, we have

$$|\langle W, Y_t \rangle| \leq \|Y_t\|_\sigma \|W\|_\Sigma \leq 1 \cdot R_k,$$

so the condition is satisfied. \square

Proof of [Theorem 6](#). We will use a meta-algorithm strategy closely resembling that of the smooth Banach space setting. The only difference is that $\|\cdot\|_\Sigma$ is not smooth, so matrix multiplicative weights, which use the log-trace-exponential as a surrogate for $\|\cdot\|_\Sigma$, is used as the sub-algorithm instead of working with $\|\cdot\|_\Sigma$ directly.

Recall that matrix multiplicative weights uses classes of the form $\mathcal{W}_r = \{W \in \mathbb{R}^{d \times d} \mid W \geq 0, \|W\|_{\Sigma} \leq r\}$ and has regret against \mathcal{W}_r bounded by $O(r\sqrt{n \log d})$ [1]. Using this strategy for fixed r as a sub-algorithm for MULTISCALEOCO, we achieve the following oracle inequality efficiently:

For each $i \in [n+1]$:

- We set $R_i = 2^{i-1}$
- $L_i = L$
- $\mathcal{W}_i = \{W \in \mathbb{R}^{d \times d} \mid W \geq 0, \|W\|_{\Sigma} \leq R_i\}$
- $\text{ALG}_i = \text{MATRIX MULTIPLICATIVE WEIGHTS}(\mathcal{W}_i)$

Finally, we set $\pi = \text{Uniform}([n+1])$. That this configuration is sufficient follows from the doubling analysis given in the proof of [Theorem 3](#). \square

A.3 Proofs from [Section 2.3](#)

Algorithm 5

```

1: procedure MULTISCALELEARNING( $\{\text{ALG}_i, R_i, L_i\}_{i \in [N]}$ ,  $\pi$ )  $\triangleright$  Collection of sub-algorithms, prior  $\pi$ .
2:    $c \leftarrow (R_i \cdot L_i)_{i \in [N]}$   $\triangleright$  Sub-algorithm scale parameters.
3:   Define  $\tilde{\ell}(\hat{y}, y) = \ell(\hat{y}, y) - \ell(0, y)$ .  $\triangleright$  Center the loss function.
4:   for  $t = 1, \dots, n$  do
5:     Receive context  $x_t$ 
6:      $\hat{y}_t^i \leftarrow \text{ALG}_i((x_1, y_1), \dots, (x_{t-1}, y_{t-1}), x_t)$  for each  $i \in \mathcal{A}$ .
7:      $i_t \leftarrow \text{MULTISCALEFTPL}[c, \pi](g_1, \dots, g_{t-1})$ .
8:     Play  $\hat{y}_t = \hat{y}_t^{i_t}$ .
9:     Observe  $y_t$  and let  $g_t = (\tilde{\ell}_t(\hat{y}_t^i, y_t))_{i \in [N]}$ .
10:   end for
11: end procedure

```

Proof of [Theorem 7](#). This theorem is an immediate consequence of [Theorem 2](#), using the absolute value $|\cdot|$ as the norm. The only significant detail one must check is that the proof of [Theorem 2](#) uses the regret statement for each sub-algorithm as a black box, and so the nonlinearity of the comparator \mathcal{F} does not change the analysis. \square

Proof of [Theorem 8](#). This is a corollary of [Theorem 7](#). That theorem, configured with one sub-algorithm for each class \mathcal{F}_k , $L_k = L$, $R_k = R_k$, and $\pi_k = 1/k$, implies

$$\mathbb{E} \left[\sum_{t=1}^n \ell(\hat{y}_t^i, y_t) - \inf_{f \in \mathcal{F}_k} \sum_{t=1}^n \ell(f(x_t), y_t) \right] \leq \mathbb{E}[\mathbf{Rad}_n(\mathcal{F}_k)] + O\left(R_k L \sqrt{n \log(R_k L n k)}\right) \quad \forall i \in [N]. \quad (19)$$

The result follows from the assumed growth rate on $\mathbf{Rad}(\mathcal{F}_k)$. \square

Proof of [Theorem 9](#). We briefly sketch the construction as follows:

1. For each \mathcal{H}_k , construct a sequence of nested subclasses (norm balls) as precisely as in the proof of [Theorem 3](#). There will be $O(n)$ sub-algorithms for each such class.
2. For each sub-algorithm in class k , take prior weight π proportional to $1/nk$.

Using the analysis from [Theorem 3](#), one can see that the result follows from [Theorem 7](#). \square