Adaptive Gradient-Based Meta-Learning Methods

Mikhail Khodak
Carnegie Mellon University
khodak@cmu.edu

Maria-Florina Balcan
Carnegie Mellon University
ninamf@cs.cmu.edu

Ameet Talwalkar
Carnegie Mellon University
& Determined AI
talwalkar@cmu.edu

Abstract

We build a theoretical framework for designing and understanding practical meta-learning methods that integrates sophisticated formalizations of task-similarity with the extensive literature on online convex optimization and sequential prediction algorithms. Our approach enables the task-similarity to be learned adaptively, provides sharper transfer-risk bounds in the setting of statistical learning-to-learn, and leads to straightforward derivations of average-case regret bounds for efficient algorithms in settings where the task-environment changes dynamically or the tasks share a certain geometric structure. We use our theory to modify several popular meta-learning algorithms and improve their meta-test-time performance on standard problems in few-shot learning and federated learning.

1 Introduction

Meta-learning, or *learning-to-learn* (LTL) [52], has recently re-emerged as an important direction for developing algorithms for multi-task learning, dynamic environments, and federated settings. By using the data of numerous training tasks, meta-learning methods seek to perform well on new, potentially related test tasks without using many samples. Successful modern approaches have also focused on exploiting the capabilities of deep neural networks, whether by learning multi-task embeddings passed to simple classifiers [51] or by neural control of optimization algorithms [46].

Because of its simplicity and flexibility, a common approach is *parameter-transfer*, where all tasks use the same class of Θ-parameterized functions $f_\theta : \mathcal{X} \mapsto \mathcal{Y}$; often a shared model $\phi \in \Theta$ is learned that is used to train within-task models. In *gradient-based meta-learning* (GBML) [23], ϕ is a meta-initialization for a gradient descent method over samples from a new task. GBML is used in a variety of LTL domains such as vision [38, 44, 35], federated learning [16], and robotics [20, 1]. Its simplicity also raises many practical and theoretical questions about the task-relations it can exploit and the settings in which it can succeed. Addressing these issues has naturally led several authors to online convex optimization (OCO) [55], either directly [24, 34] or from online-to-batch conversion [34, 19]. These efforts study how to find a meta-initialization, either by proving algorithmic learnability [24] or giving meta-test-time performance guarantees [34, 19].

However, this recent line of work has so far considered a very restricted, if natural, notion of task-similarity – closeness to a single fixed point in the parameter space. We introduce a new theoretical framework, **Average Regret-Upper-Bound Analysis (ARUBA)**, that enables the derivation of meta-learning algorithms that can provably take advantage of much more sophisticated structure. ARUBA treats meta-learning as the online learning of a sequence of losses that each upper bounds the regret on a single task. These bounds often have convenient functional forms that are (a) sufficiently nice, so that we can draw upon the existing OCO literature, and (b) strongly dependent on both the task-data and the meta-initialization, thus encoding task-similarity in a mathematically accessible way. Using ARUBA we introduce or dramatically improve upon GBML results in the following settings:

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

- **Adapting to the Task-Similarity:** A major drawback of previous work is a reliance on knowing the task-similarity beforehand to set the learning rate [24] or regularization [19], or the use of a sub-optimal guess-and-tune approach using the doubling trick [34]. ARUBA yields a simple gradient-based algorithm that eliminates the need to guess the similarity by learning it on-the-fly.
- **Adapting to Dynamic Environments:** While previous theoretical work has largely considered a fixed initialization [24, 34], in many practical applications of GBML the optimal initialization varies over time due to a changing environment [1]. We show how ARUBA reduces the problem of meta-learning in dynamic environments to a dynamic regret-minimization problem, for which there exists a vast array of online algorithms with provable guarantees that can be directly applied.
- **Adapting to the Inter-Task Geometry:** A recurring notion in LTL is that certain model weights, such as feature extractors, are shared, whereas others, such as classification layers, vary between tasks. By only learning a fixed initialization we must re-learn this structure on every task. Using ARUBA we provide a method that adapts to this structure and determines which directions in Θ need to be updated by learning a Mahalanobis-norm regularizer for online mirror descent (OMD). We show how a variant of this can be used to meta-learn a per-coordinate learning-rate for certain GBML methods, such as MAML [23] and Reptile [44], as well as for FedAvg, a popular federated learning algorithm [41]. This leads to improved meta-test-time performance on few-shot learning and a simple, tuning-free approach to effectively add user-personalization to FedAvg.
- **Statistical Learning-to-Learn:** ARUBA allows us to leverage powerful results in online-to-batch conversion [54, 33] to derive new bounds on the transfer risk when using GBML for statistical LTL [8], including fast rates in the number of tasks when the task-similarity is known and high-probability guarantees for a class of losses that includes linear regression. This improves upon the guarantees of Khodak et al. [34] and Denevi et al. [19] for similar or identical GBML methods.

1.1 Related Work

Theoretical LTL: The statistical analysis of LTL was formalized by Baxter [8]. Several works have built upon this theory for modern LTL, such as via a PAC-Bayesian perspective [3] or by learning the kernel for the ridge regression [18]. However, much effort has also been devoted to the online setting, often through the framework of lifelong learning [45, 5, 2]. Alquier et al. [2] consider a many-task notion of regret similar to the one we study in order to learn a shared data representation, although our algorithms are much more practical. Recently, Bullins et al. [11] developed an efficient online approach to learning a linear data embedding, but such a setting is distinct from GBML and more closely related to popular shared-representation methods such as ProtoNets [51]. Nevertheless, our approach does strongly rely on online learning through the study of data-dependent regret-upper-bounds, which has a long history of use in deriving adaptive single-task methods [40, 21]; however, in meta-learning there is typically not enough data to adapt to without considering multi-task data. Analyzing regret-upper-bounds was done implicitly by Khodak et al. [34], but their approach is largely restricted to using Follow-the-Leader (FTL) as the meta-algorithm. Similarly, Finn et al. [24] use FTL to show learnability of the MAML meta-initialization. In contrast, the ARUBA framework can handle general classes of meta-algorithms, which leads not only to new and improved results in static, dynamic, and statistical settings but also to significantly more practical LTL methods.

GBML: GBML stems from the Model-Agnostic Meta-Learning (MAML) algorithm [23] and has been widely used in practice [1, 44, 31]. An expressivity result was shown for MAML by Finn and Levine [22], proving that the meta-learner can approximate any permutation-invariant learner given enough data and a specific neural architecture. Under strong-convexity and smoothness assumptions and using a fixed learning rate, Finn et al. [24] show that the MAML meta-initialization is learnable, albeit via an impractical FTL method. In contrast to these efforts, Khodak et al. [34] and Denevi et al. [19] focus on providing finite-sample meta-test-time performance guarantees in the convex setting, the former for the SGD-based Reptile algorithm of Nichol et al. [44] and the latter for a regularized variant. Our work improves upon these analyses by considering the case when the learning rate, a proxy for the task-similarity, is not known beforehand as in Finn et al. [24] and Denevi et al. [19] but must be learned online; Khodak et al. [34] do consider an unknown task-similarity but use a doubling-trick-based approach that considers the absolute deviation of the task-parameters from the meta-initialization and is thus average-case suboptimal and sensitive to outliers. Furthermore, ARUBA can handle more sophisticated and dynamic notions of task-similarity and in certain settings can provide better statistical guarantees than those of Khodak et al. [34] and Denevi et al. [19].

2 Average Regret-Upper-Bound Analysis

Our main contribution is ARUBA, a framework for analyzing the learning of $\mathcal{X}$-parameterized learning algorithms via reduction to the online learning of a sequence of functions $\mathbf{U}_t : \mathcal{X} \mapsto \mathbb{R}$ upper-bounding their regret on task t. We consider a meta-learner facing a sequence of online learning tasks $t = 1, \dots, T$, each with m_t loss functions $\ell_{t,i} : \Theta \mapsto \mathbb{R}$ over action-space $\Theta \subset \mathbb{R}^d$. The learner has access to a set of learning algorithms parameterized by $x \in \mathcal{X}$ that can be used to determine the action $\theta_{t,i} \in \Theta$ on each round $i \in [m_t]$ of task t. Thus on each task t the meta-learner chooses $x_t \in \mathcal{X}$, runs the corresponding algorithm, and suffers regret $\mathbf{R}_t(x_t) = \sum_{i=1}^{m_t} \ell_{t,i}(\theta_{t,i}) - \min_\theta \sum_{i=1}^{m_t} \ell_{t,i}(\theta)$. We propose to analyze the meta-learner's performance by studying the online learning of a sequence of regret-upper-bounds $\mathbf{U}_t(x_t) \geq \mathbf{R}_t(x_t)$, specifically by bounding the **average regret-upper-bound** $\bar{\mathbf{U}}_T = \frac{1}{T}\sum_{t=1}^T \mathbf{U}_t(x_t)$. The following two observations highlight why we care about this quantity:

1. **Generality:** Many algorithms of interest in meta-learning have regret guarantees $\mathbf{U}_t(x)$ with nice, e.g. smooth and convex, functional forms that depend strongly on both their parameterizations $x \in \mathcal{X}$ and the task-data. This data-dependence lets us adaptively set the parameterization $x_t \in \mathcal{X}$.
2. **Consequences:** By definition of $\mathbf{U}_t$ we have that $\bar{\mathbf{U}}_T$ bounds the **task-averaged regret (TAR)** $\bar{\mathbf{R}}_T = \frac{1}{T}\sum_{t=1}^T \mathbf{R}_t(x_t)$ [34]. Thus if the average regret-upper-bound is small then the meta-learner will perform well on-average across tasks. In Section 5 we further show that a low average regret-upper-bound will also lead to strong statistical guarantees in the batch setting.

ARUBA's applicability depends only on finding a low-regret algorithm over the functions $\mathbf{U}_t$; then by observation 2 we get a task-averaged regret bound where the first term vanishes as $T \to \infty$ while by observation 1 the second term can be made small due to the data-dependent task-similarity:

$$\bar{\mathbf{R}}_T \leq \bar{\mathbf{U}}_T \leq o_T(1) + \min_x \frac{1}{T}\sum_{t=1}^T \mathbf{U}_t(x)$$

The Case of Online Gradient Descent: Suppose the meta-learner uses online gradient descent (OGD) as the within-task learning algorithm, as is done by Reptile [44]. OGD can be parameterized by an initialization $\phi \in \Theta$ and a learning rate $\eta > 0$, so that $\mathcal{X} = \{(\phi, \eta) : \phi \in \Theta, \eta > 0\}$. Using the notation $v_{a:b} = \sum_{i=a}^b v_i$ and $\nabla_{t,j} = \nabla \ell_{t,j}(\theta_{t,j})$, at each round i of task t OGD plays $\theta_{t,i} = \arg\min_{\theta \in \Theta} \frac{1}{2}\|\theta - \phi\|_2^2 + \eta\langle \nabla_{t,1:i-1}, \theta\rangle$. The regret of this procedure when run on m convex G-Lipschitz losses has a well-known upper-bound [48, Theorem 2.11]

$$\mathbf{U}_t(x) = \mathbf{U}_t(\phi, \eta) = \frac{1}{2\eta}\|\theta_t^* - \phi\|_2^2 + \eta G^2 m \geq \sum_{i=1}^m \ell_{t,i}(\theta_t) - \ell_{t,i}(\theta_t^*) = \mathbf{R}_t(x) \tag{1}$$

which is convex in the learning rate η and the initialization ϕ. Note the strong data dependence via $\theta_t^* \in \arg\min_\theta \sum_{i=1}^{m_t} \ell_{t,i}(\theta)$, the optimal action in hindsight. To apply ARUBA, first note that if $\bar{\theta}^* = \frac{1}{T}\theta_{1:T}^*$ is the mean of the optimal actions θ_t^* on each task and $V^2 = \frac{1}{T}\sum_{t=1}^T \|\theta_t^* - \bar{\theta}^*\|_2^2$ is their empirical variance, then $\min_{\phi,\eta} \frac{1}{T}\sum_{t=1}^T \mathbf{U}_t(\phi, \eta) = \mathcal{O}(GV\sqrt{m})$. Thus by running a low-regret algorithm on the regret-upper-bounds $\mathbf{U}_t$ the meta-learner will suffer task-averaged regret at most $o_T(1) + \mathcal{O}(GV\sqrt{m})$, which can be much better than the single-task regret $\mathcal{O}(GD\sqrt{m})$, where D is the ℓ_2-radius of Θ, if $V \ll D$, i.e. if the optimal actions θ_t^* are close together. See Theorem 3.2 for the result yielded by ARUBA in this simple setting.

3 Adapting to Similar Tasks and Dynamic Environments

We now demonstrate the effectiveness of ARUBA for analyzing GBML by using it to prove a general bound for a class of algorithms that can adapt to both *task-similarity*, i.e. when the optimal actions θ_t^* for each task are close to some good initialization, and to *changing environments*, i.e. when this initialization changes over time. The task-similarity will be measured using the **Bregman divergence** $\mathcal{B}_R(\theta||\phi) = R(\theta) - R(\phi) - \langle \nabla R(\phi), \theta - \phi\rangle$ of a 1-strongly-convex function $R : \Theta \mapsto \mathbb{R}$ [10], a generalized notion of distance. Note that for $R(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ we have $\mathcal{B}_R(\theta||\phi) = \frac{1}{2}\|\theta - \phi\|_2^2$. A changing environment will be studied by analyzing **dynamic regret**, which for a sequence of actions $\{\phi_t\}_t \subset \Theta$ taken by some online algorithm over a sequence of loss functions $\{f_t : \Theta \mapsto \mathbb{R}\}_t$ is defined w.r.t. a reference sequence $\Psi = \{\psi_t\}_t \subset \Theta$ as $\mathbf{R}_T(\Psi) = \sum_{t=1}^T f_t(\phi_t) - f_t(\psi_t)$. Dynamic regret measures the performance of an online algorithm taking actions ϕ_t relative to a potentially time-varying comparator taking actions ψ_t. Note that when we fix $\psi_t = \psi^* \in \arg\min_{\psi \in \Theta} \sum_{t=1}^T f_t(\psi)$ we recover the standard **static regret**, in which the comparator always uses the same action.

Algorithm 1: Generic online algorithm for gradient-based parameter-transfer meta-learning. To run OGD within-task set $R(\cdot) = \frac{1}{2}\|\cdot\|_2^2$. To run FTRL within-task substitute $\ell_{t,j}(\theta)$ for $\langle \nabla_{t,j}, \theta \rangle$.

Set meta-initialization $\phi_1 \in \Theta$ and learning rate $\eta_1 > 0$.
for task $t \in [T]$ **do**
  for round $i \in [m_t]$ **do**
    $\theta_{t,i} \leftarrow \arg\min_{\theta \in \Theta} \mathcal{B}_R(\theta||\phi_t) + \eta_t \langle \nabla_{t,1:i-1}, \theta \rangle$ `// online mirror descent step`
    Suffer loss $\ell_{t,i}(\theta_{t,i})$
  Update ϕ_{t+1}, η_{t+1} `// meta-update of OMD initialization and learning rate`

Putting these together, we seek to define variants of Algorithm 1 for which as $T \to \infty$ the average regret scales with V_Ψ, where $V_\Psi^2 = \frac{1}{T}\sum_{t=1}^T \mathcal{B}_R(\theta_t^*||\psi_t)$, without knowing this quantity in advance. Note for fixed $\psi_t = \bar{\theta}^* = \frac{1}{T}\theta_{1:T}^*$ this measures the empirical standard deviation of the optimal task-actions θ_t^*. Thus achieving our goal implies that average performance improves with task-similarity.

On each task t Algorithm 1 runs online mirror descent with regularizer $\frac{1}{\eta_t}\mathcal{B}_R(\cdot||\phi_t)$ for initialization $\phi_t \in \Theta$ and learning rate $\eta_t > 0$. It is well-known that OMD and the related Follow-the-Regularized-Leader (FTRL), for which our results also hold, generalize many important online methods, e.g. OGD and multiplicative weights [26]. For m_t convex losses with mean squared Lipschitz constant G_t^2 they also share a convenient, data-dependent regret-upper-bound for any $\theta_t^* \in \Theta$ [48, Theorem 2.15]:

$$\mathbf{R}_t \leq \mathbf{U}_t(\phi_t, \eta_t) = \frac{1}{\eta_t}\mathcal{B}_R(\theta_t^*||\phi_t) + \eta_t G_t^2 m_t \tag{2}$$

All that remains is to come up with update rules for the meta-initialization $\phi_t \in \Theta$ and the learning rate $\eta_t > 0$ in Algorithm 1 so that the average over T of these upper-bounds $\mathbf{U}_t(\phi_t, \eta_t)$ is small. While this can be viewed as a single online learning problem to determine actions $x_t = (\phi_t, \eta_t) \in \Theta \times (0, \infty)$, it is easier to decouple ϕ and η by first defining two function sequences $\{f_t^{\text{init}}\}_t$ and $\{f_t^{\text{sim}}\}_t$:

$$f_t^{\text{init}}(\phi) = \mathcal{B}_R(\theta_t^*||\phi) G_t \sqrt{m_t} \qquad\qquad f_t^{\text{sim}}(v) = \left(\frac{\mathcal{B}_R(\theta_t^*||\phi_t)}{v} + v\right) G_t \sqrt{m_t} \tag{3}$$

We show in Theorem 3.1 that to get an adaptive algorithm it suffices to specify two OCO algorithms, INIT and SIM, such that the actions $\phi_t = \text{INIT}(t)$ achieve good (dynamic) regret over f_t^{init} and the actions $v_t = \text{SIM}(t)$ achieve low (static) regret over f_t^{sim}; these actions then determine the update rules of ϕ_t and $\eta_t = v_t/(G_t\sqrt{m_t})$. We will specialize Theorem 3.1 to derive algorithms that provably adapt to task similarity (Theorem 3.2) and to dynamic environments (Theorem 3.3).

To understand the formulation of f_t^{init} and f_t^{sim}, first note that $f_t^{\text{sim}}(v) = \mathbf{U}_t(\phi_t, v/(G_t\sqrt{m_t}))$, so the online algorithm SIM over f_t^{sim} corresponds to an online algorithm over the regret-upper-bounds $\mathbf{U}_t$ when the sequence of initializations ϕ_t is chosen adversarially. Once we have shown that SIM is low-regret we can compare its losses $f_t^{\text{sim}}(v_t)$ to those of an arbitrary fixed $v > 0$; this is the first line in the proof of Theorem 3.1 (below). For fixed v, each $f_t^{\text{init}}(\phi_t)$ is an affine transformation of $f_t^{\text{sim}}(v)$, so the algorithm INIT with low dynamic regret over f_t^{init} corresponds to an algorithm with low dynamic regret over the regret-upper-bounds $\mathbf{U}_t$ when $\eta_t = v/(G_t\sqrt{m_t})\ \forall\ t$. Thus once we have shown a dynamic regret guarantee for INIT we can compare its losses $f_t^{\text{init}}(\phi_t)$ to those of an arbitrary comparator sequence $\{\psi_t\}_t \subset \Theta$; this is the second line in the proof of Theorem 3.1.

Theorem 3.1. *Assume $\Theta \subset \mathbb{R}^d$ is convex, each task $t \in [T]$ is a sequence of m_t convex losses $\ell_{t,i} : \Theta \mapsto \mathbb{R}$ with mean squared Lipschitz constant G_t^2, and $R : \Theta \mapsto \mathbb{R}$ is 1-strongly-convex.*

- *Let* INIT *be an algorithm whose dynamic regret over functions $\{f_t^{\mathit{init}}\}_t$ w.r.t. any reference sequence $\Psi = \{\psi_t\}_{t=1}^T \subset \Theta$ is upper-bounded by $\mathbf{U}_T^{\mathit{init}}(\Psi)$.*
- *Let* SIM *be an algorithm whose static regret over functions $\{f_t^{\mathit{sim}}\}_t$ w.r.t. any $v > 0$ is upper-bounded by a non-increasing function $\mathbf{U}_T^{\mathit{sim}}(v)$ of v.*

If Algorithm 1 sets $\phi_t = \text{INIT}(t)$ and $\eta_t = \frac{\text{SIM}(t)}{G_t\sqrt{m_t}}$ then for $V_\Psi^2 = \frac{\sum_{t=1}^T \mathcal{B}_R(\theta_t^*||\psi_t) G_t\sqrt{m_t}}{\sum_{t=1}^T G_t\sqrt{m_t}}$ it will achieve average regret

$$\bar{\mathbf{R}}_T \leq \bar{\mathbf{U}}_T \leq \frac{\mathbf{U}_T^{\mathit{sim}}(V_\Psi)}{T} + \frac{1}{T}\min\left\{\frac{\mathbf{U}_T^{\mathit{init}}(\Psi)}{V_\Psi}, 2\sqrt{\mathbf{U}_T^{\mathit{init}}(\Psi)\sum_{t=1}^T G_t\sqrt{m_t}}\right\} + \frac{2V_\Psi}{T}\sum_{t=1}^T G_t\sqrt{m_t}$$

Proof. For $\sigma_t = G_t\sqrt{m_t}$ we have by the regret bound on OMD/FTRL (2) that

$$\begin{aligned}\bar{\mathbf{U}}_T T = \sum_{t=1}^T \left(\frac{\mathcal{B}_R(\theta_t^*||\phi_t)}{v_t} + v_t\right)\sigma_t &\leq \min_{v>0} \mathbf{U}_T^{\text{sim}}(v) + \sum_{t=1}^T \left(\frac{\mathcal{B}_R(\theta_t^*||\phi_t)}{v} + v\right)\sigma_t \\ &\leq \min_{v>0} \mathbf{U}_T^{\text{sim}}(v) + \frac{\mathbf{U}_T^{\text{init}}(\Psi)}{v} + \sum_{t=1}^T \left(\frac{\mathcal{B}_R(\theta_t^*||\psi_t)}{v} + v\right)\sigma_t \\ &\leq \mathbf{U}_T^{\text{sim}}(V_\Psi) + \min\left\{\frac{\mathbf{U}_T^{\text{init}}(\Psi)}{V_\Psi}, 2\sqrt{\mathbf{U}_T^{\text{init}}(\Psi)\sigma_{1:T}}\right\} + 2V_\Psi\sigma_{1:T}\end{aligned}$$

where the last line follows by substituting $v = \max\left\{V_\Psi, \sqrt{\mathbf{U}_T^{\text{init}}(\Psi)/\sigma_{1:T}}\right\}$. $\square$

Similar Tasks in Static Environments: By Theorem 3.1, if we can specify algorithms INIT and SIM with sublinear regret over f_t^{init} and f_t^{sim} (3), respectively, then the average regret will converge to $\mathcal{O}(V_\Psi\sqrt{m})$ as desired. We first show an approach in the case when the optimal actions θ_t^* are close to a fixed point in Θ, i.e. for fixed $\psi_t = \bar{\theta}^* = \frac{1}{T}\theta_{1:T}^*$. Henceforth we assume the Lipschitz constant G and number of rounds m are the same across tasks; detailed statements are in the supplement.

Note that if $R(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ then $\{f_t^{\text{init}}\}_t$ are quadratic functions, so playing $\phi_{t+1} = \frac{1}{t}\theta_{1:t}^*$ has logarithmic regret [48, Corollary 2.2]. We use a novel strongly convex coupling argument to show that this holds for any such sequence of Bregman divergences, *even for nonconvex* $\mathcal{B}_R(\theta_t^*||\cdot)$. The second sequence $\{f_t^{\text{sim}}\}_t$ is harder because it is not smooth near 0 and not strongly convex if $\theta_t^* = \phi_t$. We study a regularized sequence $\tilde{f}_t^{\text{sim}}(v) = f_t^{\text{sim}}(v) + \varepsilon^2/v$ for $\varepsilon \geq 0$. Assuming a bound of D^2 on the Bregman divergence and setting $\varepsilon = 1/\sqrt[4]{T}$, we achieve $\tilde{\mathcal{O}}(\sqrt{T})$ regret on the original sequence by running exponentially-weighted online-optimization (EWOO) [28] on the regularized sequence:

$$v_t = \frac{\int_0^{\sqrt{D^2+\varepsilon^2}} v\exp(-\gamma\sum_{s<t}\tilde{f}_s^{\text{sim}}(v))dv}{\int_0^{\sqrt{D^2+\varepsilon^2}} \exp(-\gamma\sum_{s<t}\tilde{f}_s^{\text{sim}}(v))dv} \qquad \text{for} \qquad \gamma = \frac{2}{DG\sqrt{m}}\min\left\{\frac{\varepsilon^2}{D^2}, 1\right\} \tag{4}$$

Note that while EWOO is inefficient in high dimensions, we require only single-dimensional integrals. In the supplement we also show that simply setting $v_{t+1}^2 = \varepsilon^2 t + \sum_{s\leq t}\mathcal{B}_R(\theta_s^*||\phi_t)$ has only a slightly worse regret of $\tilde{\mathcal{O}}(T^{3/5})$. These guarantees suffice to show the following:

Theorem 3.2. *Under the assumptions of Theorem 3.1 and boundedness of* $\mathcal{B}_R$ *over* Θ*, if* INIT *plays* $\phi_{t+1} = \frac{1}{t}\theta_{1:t}^*$ *and* SIM *uses* ε*-EWOO* (4) *with* $\varepsilon = 1/\sqrt[4]{T}$ *then Algorithm 1 achieves average regret*

$$\bar{\mathbf{R}}_T \leq \bar{\mathbf{U}}_T = \tilde{\mathcal{O}}\left(\min\left\{\frac{1+\frac{1}{V}}{\sqrt{T}}, \frac{1}{\sqrt[4]{T}}\right\} + V\right)\sqrt{m} \qquad \textit{for} \qquad V^2 = \min_{\phi\in\Theta}\frac{1}{T}\sum_{t=1}^T \mathcal{B}_R(\theta_t^*||\phi)$$

Observe that if V, the average deviation of θ_t^*, is $\Omega_T(1)$ then the bound becomes $\mathcal{O}(V\sqrt{m})$ at rate $\tilde{\mathcal{O}}(1/\sqrt{T})$, while if $V = o_T(1)$ the bound tends to zero. Theorem 3.1 can be compared to the main result of Khodak et al. [34], who set the learning rate via a doubling trick. We improve upon their result in two aspects. First, their asymptotic regret is $\mathcal{O}(D^*\sqrt{m})$, where D^* is the maximum distance between *any two optimal actions*. Note that V is always at most D^*, and indeed may be much smaller in the presence of outliers. Second, our result is more general, as we do not need convex $\mathcal{B}_R(\theta_t^*||\cdot)$.

Remark 3.1. *We assume an oracle giving a unique* $\theta^* \in \arg\min_{\theta\in\Theta}\sum_{\ell\in S}\ell(\theta)$ *for any finite loss sequence* $\mathcal{S}$*, which may be inefficient or undesirable. One can instead use the last or average iterate of within-task OMD/FTRL for the meta-update; in the supplement we show that this incurs an additional* $o(\sqrt{m})$ *regret term under a quadratic growth assumption that holds in many practical settings [34].*

Related Tasks in Changing Environments: In many settings we have a changing environment and so it is natural to study dynamic regret. This has been widely analyzed by the online learning community [15, 30], often by showing a dynamic regret bound consisting of a sublinear term plus a bound on the variation in the action or function space. Using Theorem 3.1 we can show dynamic guarantees for GBML via reduction to such bounds. We provide an example in the Euclidean geometry using the popular path-length-bound $P_\Psi = \sum_{t=2}^T \|\psi_t - \psi_{t-1}\|_2$ for reference actions $\Psi = \{\psi_t\}_{t=1}^T$ [55]. We use a result showing that OGD with learning rate $\eta \leq 1/\beta$ over α-strongly-convex, β-strongly-smooth, and L-Lipschitz functions has a bound of $\mathcal{O}(L(1+P_\Psi))$ on its dynamic regret [42, Corollary 1]. Observe that in the case of $R(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ the sequence f_t^{init} in Theorem 3.1 consists of $DG\sqrt{m}$-Lipschitz quadratic functions. Thus using Theorem 3.1 we achieve the following:

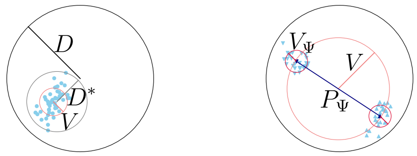

Figure 1: Left - Theorem 3.2 improves upon [34, Theorem 2.1] via its dependence on the average deviation V rather than the maximal deviation D^* of the optimal task-parameters θ_t^* (light blue). Right - a case where Theorem 3.3 yields a strong task-similarity-based guarantee via a dynamic comparator Ψ despite the deviation V being large.

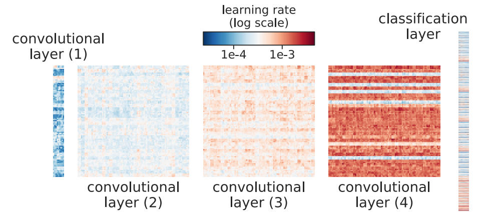

Figure 2: Learning rate variation across layers of a convolutional net trained on Mini-ImageNet using Algorithm 2. Following intuition outlined in Section 6, shared feature extractors are not updated much if at all compared to higher layers.

Theorem 3.3. *Under Theorem 3.1 assumptions, bounded Θ, and $R(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, if* INIT *is OGD with learning rate $\frac{1}{G\sqrt{m}}$ and* SIM *uses ε-EWOO* (4) *with $\varepsilon = 1/\sqrt[4]{T}$ then by using OGD within-task Algorithm 1 will achieve for any fixed comparator sequence $\Psi = \{\psi_t\}_{t\in[T]} \subset \Theta$ the average regret*

$$\bar{\mathbf{R}}_T \le \bar{\mathbf{U}}_T = \tilde{\mathcal{O}}\left(\min\left\{\frac{1+\frac{1}{V_\Psi}}{\sqrt{T}}, \frac{1}{\sqrt[4]{T}}\right\} + \min\left\{\frac{1+P_\Psi}{V_\Psi T}, \sqrt{\frac{1+P_\Psi}{T}}\right\} + V_\Psi\right)\sqrt{m}$$

for $V_\Psi^2 = \frac{1}{2T}\sum_{t=1}^T \|\theta_t^* - \psi_t\|_2^2$ and $P_\Psi = \sum_{t=2}^T \|\psi_t - \psi_{t-1}\|_2$.

This bound controls the average regret across tasks using the deviation V_Φ of the optimal task parameters θ_t^* from some reference sequence Φ, which is assumed to vary slowly or sparsely so that the path length P_Φ is small. Figure 1 illustrates when such a guarantee improves over Theorem 3.2. Note also that Theorem 3.3 specifies OGD as the meta-update algorithm INIT, so under the approximation that each task t's last iterate is close to θ_t^* this suggests that simple GBML methods such as Reptile [44] or FedAvg [41] are adaptive. The generality of ARUBA also allows for the incorporation of other dynamic regret bounds [25, 53] and other non-static notions of regret [27].

4 Adapting to the Inter-Task Geometry

Previously we gave improved guarantees for learning OMD under a simple notion of task-similarity: closeness of the optimal actions θ_t^*. We now turn to new algorithms that can adapt to a more sophisticated task-similarity structure. Specifically, we study a class of learning algorithms parameterized by an initialization $\phi \in \Theta$ and a symmetric positive-definite matrix $H \in \mathcal{M} \subset \mathbb{R}^{d\times d}$ which plays

$$\theta_{t,i} = \underset{\theta\in\Theta}{\arg\min}\ \frac{1}{2}\|\theta - \phi\|_{H^{-1}}^2 + \langle \nabla_{t,1:i-1}, \theta\rangle \tag{5}$$

This corresponds $\theta_{t,i+1} = \theta_{t,i} - H\nabla_{t,i}$, so if the optimal actions θ_t^* vary strongly in certain directions, a matrix emphasizing those directions improves within-task performance. By strong-convexity of $\frac{1}{2}\|\theta-\phi\|_{H^{-1}}^2$ w.r.t. $\|\cdot\|_{H^{-1}}$, the regret-upper-bound is $\mathbf{U}_t(\phi, H) = \frac{1}{2}\|\theta_t^* - \phi\|_{H^{-1}}^2 + \sum_{i=1}^m \|\nabla_{t,i}\|_H^2$ [48, Theorem 2.15]. We first study the diagonal case, i.e. learning a per-coordinate learning rate $\eta \in \mathbb{R}^d$ to get iteration $\theta_{t,i+1} = \theta_{t,i} - \eta_t \odot \nabla_{t,i}$. We propose to set η_t at each task t as follows:

$$\eta_t = \sqrt{\frac{\sum_{s<t}\varepsilon_s^2 + \frac{1}{2}(\theta_s^* - \phi_s)^2}{\sum_{s<t}\zeta_s^2 + \sum_{i=1}^{m_s}\nabla_{s,i}^2}} \quad \text{for } \varepsilon_t^2 = \frac{\varepsilon^2}{(t+1)^p},\ \zeta_t^2 = \frac{\zeta^2}{(t+1)^p}\ \forall\, t\ge 0, \text{ where } \varepsilon,\zeta,p > 0 \tag{6}$$

Observe the similarity between this update AdaGrad [21], which is also inversely related to the sum of the element-wise squares of all gradients seen so far. Our method adds multi-task information by setting the numerator to depend on the sum of squared distances between the initializations ϕ_t set by the algorithm and that task's optimal action θ_t^*. This algorithm has the following guarantee:

Theorem 4.1. *Let Θ be a bounded convex subset of $\mathbb{R}^d$, let $\mathcal{D} \subset \mathbb{R}^{d\times d}$ be the set of positive definite diagonal matrices, and let each task $t \in [T]$ consist of a sequence of m convex Lipschitz loss functions $\ell_{t,i} : \Theta \mapsto \mathbb{R}$. Suppose for each task t we run the iteration in Equation 5 setting $\phi = \frac{1}{t-1}\theta_{1:t-1}^*$ and setting $H = \mathrm{Diag}(\eta_t)$ via Equation 6 for $\varepsilon = 1, \zeta = \sqrt{m}$, and $p = \frac{2}{5}$. Then we achieve*

$$\bar{\mathbf{R}}_T \le \bar{\mathbf{U}}_T = \min_{\substack{\phi\in\Theta\\ H\in\mathcal{D}}} \tilde{\mathcal{O}}\left(\sum_{j=1}^d \min\left\{\frac{\frac{1}{H_{jj}} + H_{jj}}{T^{\frac{2}{5}}}, \frac{1}{\sqrt[5]{T}}\right\}\right)\sqrt{m} + \frac{1}{T}\sum_{t=1}^T \frac{\|\theta_t^* - \phi\|_{H^{-1}}^2}{2} + \sum_{i=1}^m \|\nabla_{t,i}\|_H^2$$

As $T \to \infty$ the average regret converges to the minimum over ϕ, H of the last two terms, which corresponds to running OMD with the optimal initialization and per-coordinate learning rate on every task. The rate of convergence of $T^{-2/5}$ is slightly slower than the usual $1/\sqrt{T}$ achieved in the previous section; this is due to the algorithm's adaptivity to within-task gradients, whereas previously we simply assumed a known Lipschitz bound G_t when setting η_t. This adaptivity makes the algorithm much more practical, leading to a method for adaptively learning a within-task learning rate using multi-task information; this is outlined in Algorithm 2 and shown to significantly improve GBML performance in Section 6. Note also the per-coordinate separation of the left term, which shows that the algorithm converges more quickly on non-degenerate coordinates. The per-coordinate specification of η_t (6) can be further generalized to learning a full-matrix adaptive regularizer, for which we show guarantees in Theorem 4.2. However, the rate is much slower, and without further assumptions such methods will have $\Omega(d^2)$ computation and memory requirements.

Theorem 4.2. *Let Θ be a bounded convex subset of $\mathbb{R}^d$ and let each task $t \in [T]$ consist of a sequence of m convex Lipschitz loss functions $\ell_{t,i} : \Theta \mapsto \mathbb{R}$. Suppose for each task t we run the iteration in Equation 5 with $\phi = \frac{1}{t-1}\theta^*_{1:t-1}$ and H the unique positive definite solution of $B_t^2 = HG_t^2H$ for*

$$B_t^2 = t\varepsilon^2 I_d + \frac{1}{2}\sum_{s<t}(\theta^*_s - \phi_s)(\theta^*_s - \phi_s)^T \qquad \textit{and} \qquad G_t^2 = t\zeta^2 I_d + \sum_{s<t}\sum_{i=1}^m \nabla_{s,i}\nabla_{s,i}^T$$

for $\varepsilon = 1/\sqrt[8]{T}$ and $\zeta = \sqrt{m}/\sqrt[8]{T}$. Then for λ_j corresponding to the jth largest eigenvalue we have

$$\bar{\mathbf{R}}_T \leq \bar{\mathbf{U}}_T = \tilde{\mathcal{O}}\left(\frac{1}{\sqrt[8]{T}}\right)\sqrt{m} + \min_{\substack{\phi\in\Theta\\ H\succ 0}} \frac{2\lambda_1^2(H)}{\lambda_d(H)}\frac{1+\log T}{T} + \sum_{t=1}^T \frac{\|\theta^*_t - \phi^*\|^2_{H^{-1}}}{2} + \sum_{i=1}^m \|\nabla_{t,i}\|_H^2$$

5 Fast Rates and High Probability Bounds for Statistical Learning-to-Learn

Batch-setting transfer risk bounds have been an important motivation for studying LTL via online learning [2, 34, 19]. If the regret-upper-bounds are convex, which is true for most practical variants of OMD/FTRL, ARUBA yields several new results in the classical distribution over task-distributions setup of Baxter [8]. In Theorem 5.1 we present bounds on the risk $\ell_{\mathcal{P}}(\bar{\theta})$ of the parameter $\bar{\theta}$ obtained by running OMD/FTRL on i.i.d. samples from a new task distribution $\mathcal{P}$ and averaging the iterates.

Theorem 5.1. *Assume $\Theta, \mathcal{X}$ are convex Euclidean subsets. Let convex losses $\ell_{t,i} : \Theta \mapsto [0,1]$ be drawn i.i.d. $\mathcal{P}_t \sim \mathcal{Q}, \{\ell_{t,i}\}_i \sim \mathcal{P}_t^m$ for distribution $\mathcal{Q}$ over tasks. Suppose they are passed to an algorithm with average regret upper-bound $\bar{\mathbf{U}}_T$ that at each t picks $x_t \in \mathcal{X}$ to initialize a within-task method with convex regret upper-bound $\mathbf{U}_t : \mathcal{X} \mapsto [0, B\sqrt{m}]$, for $B \geq 0$. If the within-task algorithm is initialized by $\bar{x} = \frac{1}{T}x_{1:T}$ and it takes actions $\theta_1, \dots, \theta_m$ on m i.i.d. losses from new task $\mathcal{P} \sim \mathcal{Q}$ then $\bar{\theta} = \frac{1}{m}\theta_{1:m}$ satisfies the following transfer risk bounds for any $\theta^* \in \Theta$ (all w.p. $1-\delta$):*

1. **general case:** $\mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}_{\mathcal{P}^m}\ell_{\mathcal{P}}(\bar{\theta}) \leq \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\ell_{\mathcal{P}}(\theta^*) + \mathcal{L}_T$ *for* $\mathcal{L}_T = \frac{\bar{\mathbf{U}}}{m} + B\sqrt{\frac{8}{mT}\log\frac{1}{\delta}}$.
2. **ρ-self-bounded losses ℓ:** *if $\exists\, \rho > 0$ s.t. $\rho\,\mathbb{E}_{\ell\sim\mathcal{P}}\Delta\ell(\theta) \geq \mathbb{E}_{\ell\sim\mathcal{P}}(\Delta\ell(\theta) - \mathbb{E}_{\ell\sim\mathcal{P}}\Delta\ell(\theta))^2$ for all distributions $\mathcal{P} \sim \mathcal{Q}$, where $\Delta\ell(\theta) = \ell(\theta) - \ell(\theta^*)$ for any $\theta^* \in \arg\min_{\theta\in\Theta}\ell_{\mathcal{P}}(\theta)$, then for $\mathcal{L}_T$ as above we have* $\mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\ell_{\mathcal{P}}(\bar{\theta}) \leq \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\ell_{\mathcal{P}}(\theta^*) + \mathcal{L}_T + \sqrt{\frac{2\rho\mathcal{L}_T}{m}\log\frac{2}{\delta}} + \frac{3\rho+2}{m}\log\frac{2}{\delta}$.
3. **α-strongly-convex, G-Lipschitz regret-upper-bounds $\mathbf{U}_t$:** *in parts 1 and 2 above we can substitute* $\mathcal{L}_T = \frac{\bar{\mathbf{U}} + \min_x \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbf{U}(x)}{m} + \frac{4G}{T}\sqrt{\frac{\bar{\mathbf{U}}}{\alpha m}\log\frac{8\log T}{\delta}} + \frac{\max\{16G^2, 6\alpha B\sqrt{m}\}}{\alpha m T}\log\frac{8\log T}{\delta}$.

In the **general case**, Theorem 5.1 provides bounds on the excess transfer risk decreasing with $\bar{\mathbf{U}}/m$ and $1/\sqrt{mT}$. Thus if $\bar{\mathbf{U}}$ improves with task-similarity so will the transfer risk as $T \to \infty$. Note that the second term is $1/\sqrt{mT}$ rather than $1/\sqrt{T}$ as in most-analyses [34, 19]; this is because regret is m-bounded but the OMD regret-upper-bound is $\mathcal{O}(\sqrt{m})$-bounded. The results also demonstrate ARUBA's ability to utilize specialized results from the online-to-batch conversion literature. This is witnessed by the guarantee for **self-bounded losses**, a class which Zhang [54] shows includes linear regression; we use a result by the same author to obtain high-probability bounds, whereas previous GBML bounds are in-expectation [34, 19]. We also apply a result due to Kakade and Tewari [33] for the case of **strongly-convex regret-upper-bounds**, enabling fast rates in the number of tasks T. The strongly-convex case is especially relevant for GBML since it holds for OGD with fixed learning rate.

Algorithm 2: ARUBA: an approach for modifying a generic batch GBML method to learn a per-coordinate learning rate. Two specialized variants provided below.

Input: T tasks, update method for meta-initialization, within-task descent method, settings $\varepsilon, \zeta, p > 0$

Initialize $b_1 \leftarrow \varepsilon^2 1_d$, $g_1 \leftarrow \zeta^2 1_d$

for task $t = 1, 2, \dots, T$ **do**

Set ϕ_t according to update method, $\eta_t \leftarrow \sqrt{b_t / g_t}$

Run descent method from ϕ_t with learning rate η_t:

observe gradients $\nabla_{t,1}, \dots, \nabla_{t,m_t}$

obtain within-task parameter $\hat{\theta}_t$

$b_{t+1} \leftarrow b_t + \frac{\varepsilon^2 1_d}{(t+1)^p} + \frac{1}{2}(\phi_t - \hat{\theta}_t)^2$

$g_{t+1} \leftarrow g_t + \frac{\zeta^2 1_d}{(t+1)^p} + \sum_{i=1}^{m_t} \nabla_{t,i}^2$

Result: initialization ϕ_T, learning rate $\eta_T = \sqrt{b_T / g_T}$

ARUBA++: starting with $\eta_{T,1} = \eta_T$ and $g_{T,1} = g_T$, adaptively reset the learning rate by setting $\hat{g}_{T,i+1} \leftarrow \hat{g}_{T,i} + c\nabla_i^2$ for some $c > 0$ and then updating $\eta_{T,i+1} \leftarrow \sqrt{b_T / g_{T,i+1}}$.

Isotropic: b_t and g_t are scalars tracking the sum of squared distances and sum of squared gradient norms, respectively.

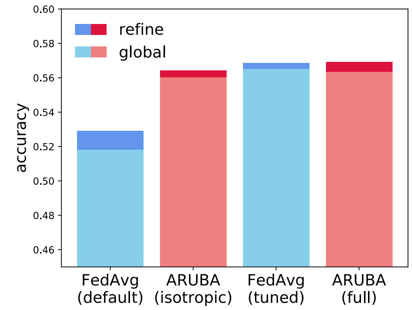

Figure 3: Next-character prediction performance for recurrent networks trained on the Shakespeare dataset [12] using FedAvg [41] and its modifications by Algorithm 2. Note that the two ARUBA methods require no learning rate tuning when personalizing the model (refine), unlike *both* FedAvg methods; this is a critical improvement in federated settings. Furthermore, isotropic ARUBA has negligible overhead by only communicating scalars.

We present two consequences of these results for the algorithms from Section 3 when run on i.i.d. data. To measure task-similarity we use the **variance** $V_{\mathcal{Q}}^2 = \min_{\phi \in \Theta} \mathbb{E}_{\mathcal{P} \sim \mathcal{Q}} \mathbb{E}_{\mathcal{P}^m} \|\theta^* - \phi\|_2^2$ of the empirical risk minimizer θ^* of an m-sample task drawn from $\mathcal{Q}$. If $V_{\mathcal{Q}}$ is known we can use strong-convexity of the regret-upper-bounds to obtain a fast rate for learning the initialization, as shown in the first part of Corollary 5.1. The result can be loosely compared to Denevi et al. [19], who provide a similar asymptotic improvement but with a slower rate of $\mathcal{O}(1/\sqrt{T})$ in the second term. However, their task-similarity measures the deviation of the true, not empirical, risk-minimizers, so the results are not directly comparable. Corollary 5.1 also gives a guarantee for when we do *not* know $V_{\mathcal{Q}}$ and must learn the learning rate η in addition to the initialization; here we match the rate of Denevi et al. [19], who do not learn η, up to some additional fast $o(1/\sqrt{m})$ terms.

Corollary 5.1. *In the setting of Theorems 3.2 & 5.1, if $\delta \leq 1/e$ and Algorithm 1 uses within-task OGD with initialization $\phi_{t+1} = \frac{1}{t}\theta_{1:t}^*$ and step-size $\eta_t = \frac{V_{\mathcal{Q}} + 1/\sqrt{T}}{G\sqrt{m}}$ for $V_{\mathcal{Q}}$ as above, then w.p. $1 - \delta$*

$$\mathop{\mathbb{E}}_{\mathcal{P} \sim \mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} \ell_{\mathcal{P}}(\bar{\theta}) \leq \mathop{\mathbb{E}}_{\mathcal{P} \sim \mathcal{Q}} \ell_{\mathcal{P}}(\theta^*) + \tilde{\mathcal{O}}\left(\frac{V_{\mathcal{Q}}}{\sqrt{m}} + \left(\frac{1}{\sqrt{mT}} + \frac{1}{T}\right)\log\frac{1}{\delta}\right)$$

If η_t is set adaptively using ε-EWOO as in Theorem 3.2 for $\varepsilon = 1/\sqrt[4]{mT} + 1/\sqrt{m}$ then w.p. $1 - \delta$

$$\mathop{\mathbb{E}}_{\mathcal{P} \sim \mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} \ell_{\mathcal{P}}(\bar{\theta}) \leq \mathop{\mathbb{E}}_{\mathcal{P} \sim \mathcal{Q}} \ell_{\mathcal{P}}(\theta^*) + \tilde{\mathcal{O}}\left(\frac{V_{\mathcal{Q}}}{\sqrt{m}} + \min\left\{\frac{\frac{1}{\sqrt{m}} + \frac{1}{\sqrt{T}}}{V_{\mathcal{Q}} m}, \frac{1}{\sqrt[4]{m^3 T}} + \frac{1}{m}\right\} + \sqrt{\frac{1}{T}\log\frac{1}{\delta}}\right)$$

6 Empirical Results: Adaptive Methods for Few-Shot & Federated Learning

A generic GBML method does the following at iteration t: (1) initialize a descent method at ϕ_t; (2) take gradient steps with learning rate η to get task-parameter $\hat{\theta}_t$; (3) update meta-initialization to ϕ_{t+1}. Motivated by Section 4, in Algorithm 2 we outline a generic way of replacing η by a per-coordinate rate learned on-the-fly. This entails keeping track of two quantities: (1) $b_t \in \mathbb{R}^d$, a per-coordinate sum over $s < t$ of the squared distances from the initialization ϕ_s to within-task parameter $\hat{\theta}_s$; (2) $g_t \in \mathbb{R}^d$, a per-coordinate sum of the squared gradients seen so far. At task t we set η to be the element-wise square root of b_t / g_t, allowing multi-task information to inform the trajectory. For example, if along coordinate j the $\hat{\theta}_{t,j}$ is usually not far from initialization then b_j will be small and thus so will η_j; then if on a new task we get a high noisy gradient along coordinate j the performance will be less adversely affected because it will be down-weighted by the learning rate. Single-task algorithms such as AdaGrad [21] and Adam [36] also work by reducing the learning rate along frequent directions.

		20-way Omniglot		5-way Mini-ImageNet	
		1-shot	5-shot	1-shot	5-shot
1st Order	1st-Order MAML [23]	89.4 ± 0.5	$\underline{97.9} \pm 0.1$	48.07 ± 1.75	63.15 ± 0.91
	Reptile [44] w. Adam [36]	89.43 ± 0.14	97.12 ± 0.32	49.97 ± 0.32	$\underline{\mathbf{65.99}} \pm 0.58$
	Reptile w. ARUBA	86.67 ± 0.17	96.61 ± 0.13	$\underline{\mathbf{50.73}} \pm 0.32$	65.69 ± 0.61
	Reptile w. ARUBA++	$\underline{89.66} \pm 0.3$	97.49 ± 0.28	50.35 ± 0.74	65.89 ± 0.34
2nd Order	2nd-Order MAML	95.8 ± 0.3	98.9 ± 0.2	48.7 ± 1.84	63.11 ± 0.92
	Meta-SGD [38]	$\underline{\mathbf{95.93}} \pm 0.38$	$\underline{\mathbf{98.97}} \pm 0.19$	$\underline{50.47} \pm 1.87$	$\underline{64.03} \pm 0.94$

Table 1: Meta-test-time performance of GBML algorithms on few-shot classification benchmarks. 1st-order and 2nd-order results obtained from Nichol et al. [44] and Li et al. [38], respectively.

However, in meta-learning some coordinates may be frequently updated during meta-training because good task-weights vary strongly from the best initialization along them, and thus their gradients should not be downweighted; ARUBA encodes this intuition in the numerator using the distance-traveled per-task along each direction, which increases the learning rate along high-variance directions. We show in Figure 2 that this is realized in practice, as ARUBA assigns a faster rate to deeper layers than to lower-level feature extractors, following standard intuition in parameter-transfer meta-learning. As described in Algorithm 2, we also consider two variants: ARUBA++, which updates the meta-learned learning-rate at meta-test-time in a manner similar to AdaGrad, and Isotropic ARUBA, which only tracks scalar quantities and is thus useful for communication-constrained settings.

Few-Shot Classification: We first examine if Algorithm 2 can improve performance on Omniglot [37] and Mini-ImageNet [46], two standard few-shot learning benchmarks, when used to modify Reptile, a simple meta-learning method [44]. In its serial form Reptile is roughly the algorithm we study in Section 3 when OGD is used within-task and η is fixed. Thus we can set Reptile+ARUBA to be Algorithm 2 with $\hat{\theta}_t$ the last iterate of OGD and the meta-update a weighted sum of $\hat{\theta}_t$ and ϕ_t. In practice, however, Reptile uses Adam [36] to exploit multi-task gradient information. As shown in Table 1, ARUBA matches or exceeds this baseline on Mini-ImageNet, although on Omniglot it requires the additional within-task updating of ARUBA++ to show improvement.

It is less clear how ARUBA can be applied to MAML [23], as by only taking one step the distance traveled will be proportional to the gradient, so η will stay fixed. We also do not find that ARUBA improves multi-step MAML – perhaps not surprising as it is further removed from our theory due to its use of held-out data. In Table 1 we compare to Meta-SGD [38], which does learn a per-coordinate learning rate for MAML by automatic differentiation. This requires more computation but does lead to consistent improvement. As with the original Reptile, our modification performs better on Mini-ImageNet but worse on Omniglot compared to MAML and its modification Meta-SGD.

Federated Learning: A main goal in this setting is to use data on heterogeneous nodes to learn a global model without much communication; leveraging this to get a personalized model is an auxiliary goal [50], with a common application being next-character prediction on mobile devices. A popular method is FedAvg [41], where at each communication round r the server sends a global model ϕ_r to a batch of nodes, which then run local OGD; the server then sets ϕ_{r+1} to the average of the returned models. This can be seen as a GBML method with each node a task, making it easy to apply ARUBA: each node simply sends its accumulated squared gradients to the server together with its model. The server can use this information and the squared difference between ϕ_r and ϕ_{r+1} to compute a learning rate η_{r+1} via Algorithm 2 and send it to each node in the next round. We use FedAvg with ARUBA to train a character LSTM [29] on the Shakespeare dataset, a standard benchmark of a thousand users with varying amounts of non-i.i.d. data [41, 12]. Figure 3 shows that ARUBA significantly improves over non-tuned FedAvg and matches the performance of FedAvg with a tuned learning rate schedule. Unlike both baselines we also do not require step-size tuning when refining the global model for personalization. This reduced need for hyperparameter optimization is crucial in federated settings, where the number of user-data accesses are extremely limited.

7 Conclusion

In this paper we introduced ARUBA, a framework for analyzing GBML that is both flexible and consequential, yielding new guarantees for adaptive, dynamic, and statistical LTL via online learning. As a result we devised a novel per-coordinate learning rate applicable to generic GBML procedures, improving their training and meta-test-time performance on few-shot and federated learning. We see great potential for applying ARUBA to derive many other new LTL methods in a similar manner.

Acknowledgments

We thank Jeremy Cohen, Travis Dick, Nikunj Saunshi, Dravyansh Sharma, Ellen Vitercik, and our three anonymous reviewers for helpful feedback. This work was supported in part by DARPA FA875017C0141, National Science Foundation grants CCF-1535967, CCF-1910321, IIS-1618714, IIS-1705121, IIS-1838017, and IIS-1901403, a Microsoft Research Faculty Fellowship, a Bloomberg Data Science research grant, an Amazon Research Award, an Amazon Web Services Award, an Okawa Grant, a Google Faculty Award, a JP Morgan AI Research Faculty Award, and a Carnegie Bosch Institute Research Award. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the National Science Foundation, or any other funding agency.

References

[1] Maruan Al-Shedivat, Trapit Bansal, Yura Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[2] Pierre Alquier, The Tien Mai, and Massimiliano Pontil. Regret bounds for lifelong learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[3] Ron Amit and Ron Meir. Meta-learning by adjusting priors based on extended PAC-Bayes theory. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[4] Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tôhoku Mathematical Journal*, 19:357–367, 1967.

[5] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. Efficient representations for lifelong learning and autoencoding. In *Proceedings of the Conference on Learning Theory*, 2015.

[6] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

[7] Peter L. Bartlett, Elad Hazan, and Alexander Rakhlin. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems*, 2008.

[8] Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

[9] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[10] Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

[11] Brian Bullins, Elad Hazan, Adam Kalai, and Roi Livni. Generalize across tasks: Efficient algorithms for linear representation learning. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, 2019.

[12] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A benchmark for federated settings. arXiv, 2018.

[13] Nicoló Cesa-Bianchi and Claudio Gentile. Improved risk tail bounds for on-line algorithms. In *Advances in Neural Information Processing Systems*, 2005.

[14] Nicoló Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

[15] Nicoló Cesa-Bianchi, Pierre Gaillard, Gabor Lugosi, and Gilles Stoltz. A new look at shifting regret. HAL, 2012.

[16] Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning for recommendation. arXiv, 2018.

[17] Chandler Davis. Notions generalizing convexity for functions defined on spaces of matrices. In *Proceedings of Symposia in Pure Mathematics*, 1963.

[18] Giulia Denevi, Carlo Ciliberto, Dimitris Stamos, and Massimiliano Pontil. Incremental learning-to-learn with statistical guarantees. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2018.

[19] Giulia Denevi, Carlo Ciliberto, Riccardo Grazzi, and Massimiliano Pontil. Learning-to-learn stochastic gradient descent with biased regularization. arXiv, 2019.

[20] Yan Duan, Marcin Andrychowicz, Bradly Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in Neural Information Processing Systems*, 2017.

[21] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[22] Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[23] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[24] Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergei Levine. Online meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. To Appear.

[25] Eric C. Hall and Rebecca M. Willet. Online optimization in dynamic environments. arXiv, 2016.

[26] Elad Hazan. Introduction to online convex optimization. In *Foundations and Trends in Optimization*, volume 2, pages 157–325. now Publishers Inc., 2015.

[27] Elad Hazan and C. Seshadri. Efficient learning algorithms for changing environments. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[28] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69:169–192, 2007.

[29] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9: 1735–1780, 1997.

[30] Ali Jadbabaie, Alexander Rakhlin, and Shahin Shahrampour. Online optimization : Competing with dynamic comparators. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015.

[31] Ghassen Jerfel, Erin Grant, Thomas L. Griffiths, and Katherine Heller. Online gradient-based mixtures for transfer modulation in meta-learning. arXiv, 2018.

[32] Sham Kakade and Shai Shalev-Shwartz. Mind the duality gap: Logarithmic regret algorithms for online optimization. In *Advances in Neural Information Processing Systems*, 2008.

[33] Sham Kakade and Ambuj Tewari. On the generalization ability of online strongly convex programming algorithms. In *Advances in Neural Information Processing Systems*, 2008.

[34] Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Provable guarantees for gradient-based meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. To Appear.

[35] Jaehong Kim, Sangyeul Lee, Sungwan Kim, Moonsu Cha, Jung Kwon Lee, Youngduck Choi, Yongseok Choi, Dong-Yeon Choi, and Jiwon Kim. Auto-Meta: Automated gradient based meta learner search. arXiv, 2018.

[36] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[37] Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the Conference of the Cognitive Science Society (CogSci)*, 2017.

[38] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-SGD: Learning to learning quickly for few-shot learning. arXiv, 2017.

[39] Elliott H. Lieb. Convex trace functions and the Wigner-Yanase-Dyson conjecture. *Advances in Mathematics*, 11:267–288, 1973.

[40] H. Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. In *Proceedings of the Conference on Learning Theory*, 2010.

[41] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artifical Intelligence and Statistics*, 2017.

[42] Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *Proceedings of the 55th IEEE Conference on Decision and Control*, 2016.

[43] Ken-ichiro Moridomi, Kohei Hatano, and Eiji Takimoto. Online linear optimization with the log-determinant regularizer. *IEICE Transactions on Information and Systems*, E101-D(6): 1511–1520, 2018.

[44] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. arXiv, 2018.

[45] Anastasia Pentina and Christoph H. Lampert. A PAC-Bayesian bound for lifelong learning. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[46] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[47] Ankan Saha, Prateek Jain, and Ambuj Tewari. The interplay between stability and regret in online learning. arXiv, 2012.

[48] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107—-194, 2011.

[49] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11, 2010.

[50] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, 2017.

[51] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

[52] Sebastian Thrun and Lorien Pratt. *Learning to Learn*. Springer Science & Business Media, 1998.

[53] Lijun Zhang, Tianbao Yang, Jinfeng Yi, and Rong Jin Zhi-Hua Zhou. Improved dynamic regret for non-degenerate functions. In *Advances in Neural Information Processing Systems*, 2017.

[54] Tong Zhang. Data dependent concentration bounds for sequential prediction algorithms. In *Proceedings of the International Conference on Learning Theory*, 2005.

[55] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.

A Background and Results for Online Convex Optimization

Throughout the appendix we assume all subsets are convex and in a finite-dimensional real vector space with inner product $\langle \cdot, \cdot \rangle$ unless explicitly stated. Let $\| \cdot \|_*$ be the dual norm of $\| \cdot \|$ and note that the dual norm of $\| \cdot \|_2$ is itself. For sequences of scalars $\sigma_1, \dots, \sigma_T \in \mathbb{R}$ we will use the notation $\sigma_{1:t}$ to refer to the sum of the first t of them. In the online learning setting, we will use the shorthand ∇_t to denote the subgradient of $\ell_t : \Theta \mapsto \mathbb{R}$ evaluated at action $\theta_t \in \Theta$. We will use $\operatorname{Conv}(S)$ to refer to the convex hull of a set of points S and $\operatorname{Proj}_S(\cdot)$ to be the projection to any convex subset S.

A.1 Convex Functions

We first state the related definitions of *strong convexity* and *strong smoothness*:

Definition A.1. *An everywhere sub-differentiable function $f : S \mapsto \mathbb{R}$ is α-**strongly-convex** w.r.t. norm $\| \cdot \|$ if*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\alpha}{2} \|y - x\|^2 \ \forall\, x, y \in S$$

Definition A.2. *An everywhere sub-differentiable function $f : S \mapsto \mathbb{R}$ is β-**strongly-smooth** w.r.t. norm $\| \cdot \|$ if*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\beta}{2} \|y - x\|^2 \ \forall\, x, y \in S$$

Finally, we will also consider functions that are exp-concave [28]:

Definition A.3. *An everywhere sub-differentiable function $f : S \mapsto \mathbb{R}$ is γ-**exp-concave** if $\exp(-\gamma f(x))$ is concave. For $S \subset \mathbb{R}$ we have that $\frac{\partial_{xx} f(x)}{(\partial_x f(x))^2} \geq \gamma \ \forall\, x \in S \implies f$ is γ-exp-concave.*

We now turn to the *Bregman divergence* and a discussion of several useful properties [10, 6]:

Definition A.4. *Let $f : S \mapsto \mathbb{R}$ be an everywhere sub-differentiable strictly convex function. Its* **Bregman divergence** *is defined as*

$$\mathcal{B}_f(x||y) = f(x) - f(y) - \langle \nabla f(y), x - y \rangle$$

The definition directly implies that $\mathcal{B}_f(\cdot||y)$ preserves the (strong or strict) convexity of f for any fixed $y \in S$. Strict convexity further implies $\mathcal{B}_f(x||y) \geq 0 \ \forall\, x, y \in S$, with equality iff $x = y$. Finally, if f is α-strongly-convex, or β-strongly-smooth, w.r.t. $\| \cdot \|$ then Definitions A.1 and A.2 imply $\mathcal{B}_f(x||y) \geq \frac{\alpha}{2}\|x - y\|^2$ or $\mathcal{B}_f(x||y) \leq \frac{\beta}{2}\|x - y\|^2$, respectively.

Claim A.1. *Let $f : S \mapsto \mathbb{R}$ be a strictly convex function on S, $\alpha_1, \dots, \alpha_n \in \mathbb{R}$ be a sequence satisfying $\alpha_{1:n} > 0$, and $x_1, \dots, x_n \in S$. Then*

$$\bar{x} = \frac{1}{\alpha_{1:n}} \sum_{i=1}^n \alpha_i x_i = \underset{y \in S}{\arg\min} \sum_{i=1}^n \alpha_i \mathcal{B}_f(x_i||y)$$

Proof. $\forall\, y \in S$ we have

$$\begin{aligned}
&\sum_{i=1}^n \alpha_i \left(\mathcal{B}_f(x_i||y) - \mathcal{B}_f(x_i||\bar{x})\right) \\
&\quad= \sum_{i=1}^n \alpha_i \left(f(x_i) - f(y) - \langle \nabla f(y), x_i - y \rangle - f(x_i) + f(\bar{x}) + \langle \nabla f(\bar{x}), x_i - \bar{x} \rangle\right) \\
&\quad= \left(f(\bar{x}) - f(y) + \langle \nabla f(y), y \rangle\right) \alpha_{1:n} + \sum_{i=1}^n \alpha_i \left(-\langle \nabla f(\bar{x}), \bar{x} \rangle + \langle \nabla f(\bar{x}) - \nabla f(y), x_i \rangle\right) \\
&\quad= \left(f(\bar{x}) - f(y) - \langle \nabla f(y), \bar{x} - y \rangle\right) \alpha_{1:n} \\
&\quad= \alpha_{1:n} \mathcal{B}_f(\bar{x}||y)
\end{aligned}$$

By Definition A.4 the last expression has a unique minimum at $y = \bar{x}$. $\square$

A.2 Online Algorithms

Here we provide a review of the online algorithms we use. Recall that in this setting our goal is minimizing regret:

Definition A.5. *The* **regret** *of an agent playing actions* $\{\theta_t \in \Theta\}_{t\in[T]}$ *on a sequence of loss functions* $\{\ell_t : \Theta \mapsto \mathbb{R}\}_{t\in[T]}$ *is*

$$\mathbf{R}_T = \sum_{t=1}^T \ell_t(\theta_t) - \min_{\theta\in\Theta} \sum_{t=1}^T \ell_t(\theta)$$

Within-task our focus is on two closely related meta-algorithms, Follow-the-Regularized-Leader (FTRL) and (linearized lazy) Online Mirror Descent (OMD).

Definition A.6. *Given a strictly convex function* $R : \Theta \mapsto \mathbb{R}$*, starting point* $\phi \in \Theta$*, fixed learning rate* $\eta > 0$*, and a sequence of functions* $\{\ell_t : \Theta \mapsto \mathbb{R}\}_{t\geq 1}$*,* **Follow-the-Regularized Leader (**$\mathrm{FTRL}_{\phi,\eta}^{(R)}$**)** *plays*

$$\theta_t = \operatorname*{arg\,min}_{\theta\in\Theta} \mathcal{B}_R(\theta||\phi) + \eta \sum_{s<t} \ell_s(\theta)$$

Definition A.7. *Given a strictly convex function* $R : \Theta \mapsto \mathbb{R}$*, starting point* $\phi \in \Theta$*, fixed learning rate* $\eta > 0$*, and a sequence of functions* $\{\ell_t : \Theta \mapsto \mathbb{R}\}_{t\geq 1}$*,* **lazy linearized Online Mirror Descent (**$\mathrm{OMD}_{\phi,\eta}^{(R)}$**)** *plays*

$$\theta_t = \operatorname*{arg\,min}_{\theta\in\Theta} \mathcal{B}_R(\theta||\phi) + \eta \sum_{s<t} \langle \nabla_s, \theta\rangle$$

These formulations make the connection between the two algorithms – their equivalence in the linear case $\ell_s(\cdot) = \langle \nabla_s, \cdot\rangle$ – very explicit. There exists a more standard formulation of OMD that is used to highlight its generalization of OGD – the case of $R(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ – and the fact that the update is carried out in the dual space induced by R [26, Section 5.3]. However, we will only need the following regret bound satisfied by both [48, Theorems 2.11 and 2.15]

Theorem A.1. *Let* $\{\ell_t : \Theta \mapsto \mathbb{R}\}_{t\in[T]}$ *be a sequence of convex functions that are* G_t*-Lipschitz w.r.t.* $\|\cdot\|$ *and let* $R : S \mapsto \mathbb{R}$ *be 1-strongly-convex. Then the regret of both* $\mathrm{FTRL}_{\eta,\phi}^{(R)}$ *and* $\mathrm{OMD}_{\eta,\phi}^{(R)}$ *is bounded by*

$$\mathbf{R}_T \leq \frac{\mathcal{B}_R(\theta^*||\phi)}{\eta} + \eta G^2 T$$

for all $\theta^* \in \Theta$ *and* $G^2 \geq \frac{1}{T}\sum_{t=1}^T G_t^2$*.*

We next review the online algorithms we use for the meta-update. The main requirement here is logarithmic regret guarantees for the case of strongly convex loss functions, which is satisfied by two well-known algorithms:

Definition A.8. *Given a sequence of strictly convex functions* $\{\ell_t : \Theta \mapsto \mathbb{R}\}_{t\geq 1}$*,* **Follow-the-Leader (FTL)** *plays arbitrary* $\theta_1 \in \Theta$ *and for* $t > 1$ *plays*

$$\theta_t = \operatorname*{arg\,min}_{\theta\in\Theta} \sum_{s<t} \ell_s(\theta)$$

Definition A.9. *Given a sequence of functions* $\{\ell_t : \Theta \mapsto \mathbb{R}\}_{t\geq 1}$ *that are* α_t*-strongly-convex w.r.t.* $\|\cdot\|_2$*,* **Adaptive OGD (AOGD)** *plays arbitrary* $\theta_1 \in \Theta$ *and for* $t > 1$ *plays*

$$\theta_{t+1} = \mathrm{Proj}_\Theta\left(\theta_t - \frac{1}{\alpha_{1:t}} \nabla f(\theta_t)\right)$$

Kakade and Shalev-Shwartz [32, Theorem 2] and Bartlett et al. [7, Theorem 2.1] provide for FTL and AOGD, respectively, the following regret bound:

Theorem A.2. *Let* $\{\ell_t : \Theta \mapsto \mathbb{R}\}_{t\in[T]}$ *be a sequence of convex functions that are* G_t*-Lipschitz and* α_t*-strongly-convex w.r.t.* $\|\cdot\|$*. Then the regret of both FTL and AOGD is bounded by*

$$\mathbf{R}_T \leq \frac{1}{2}\sum_{t=1}^T \frac{G_t^2}{\alpha_{1:t}}$$

Finally, we state the EWOO algorithm due to Hazan et al. [28]. While difficult to run in high-dimensions, we will be running this method in single dimensions, when computing it requires only one integral.

Definition A.10. *Given a sequence of γ-exp-concave functions $\{\ell_t : \Theta \mapsto \mathbb{R}\}$,* **Exponentially Weighted Online Optimization (EWOO)** *plays*

$$\theta_t = \frac{\int_\Theta \theta \exp(-\gamma \sum_{s<t} \ell_s(\theta))d\theta}{\int_\Theta \exp(-\gamma \sum_{s<t} \ell_s(\theta))d\theta}$$

Hazan et al. [28, Theorem 7] provide the following guarantee for EWOO, which is notable for its lack of explicit dependence on the Lipschitz constant.

Theorem A.3. *Let $\{\ell_t : \Theta \mapsto \mathbb{R}\}$ be a sequence of γ-exp-concave functions. Then the regret of EWOO is bounded by*

$$\mathbf{R}_T \leq \frac{d}{\gamma}(1 + \log(T+1))$$

A.3 Online-to-Batch Conversion

Finally, as we are also interested in distributional meta-learning, we discuss some techniques for converting regret guarantees into generalization bounds, which are usually named *online-to-batch conversions*. We first state some standard results.

Proposition A.1. *If a sequence of bounded convex loss functions $\{\ell_t : \Theta \mapsto \mathbb{R}\}_{t\in[T]}$ drawn i.i.d. from some distribution $\mathcal{D}$ is given to an online algorithm with regret bound $\mathbf{R}_T$ that generates a sequence of actions $\{\theta_t \in \Theta\}_{t\in[T]}$ then*

$$\mathop{\mathbb{E}}_{\mathcal{D}^T} \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}} \ell(\bar\theta) \leq \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}} \ell(\theta^*) + \frac{\mathbf{R}_T}{T}$$

for $\bar\theta = \frac{1}{T}\theta_{1:T}$ and any $\theta^* \in \Theta$.

Proof. Applying Jensen's inequality yields

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{D}^T} \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}} \ell(\bar\theta) &\leq \frac{1}{T} \mathop{\mathbb{E}}_{\mathcal{D}^T} \sum_{t=1}^T \mathop{\mathbb{E}}_{\ell'_t\sim\mathcal{D}} \ell'_t(\theta_t) \\
&= \frac{1}{T} \mathop{\mathbb{E}}_{\{\ell_t\}\sim\mathcal{D}^T} \left(\sum_{t=1}^T \mathop{\mathbb{E}}_{\ell'_t\sim\mathcal{D}} \ell'_t(\theta_t) - \ell_t(\theta_t)\right) + \frac{1}{T} \mathop{\mathbb{E}}_{\{\ell_t\}\sim\mathcal{D}^T} \left(\sum_{t=1}^T \ell_t(\theta_t)\right) \\
&\leq \frac{1}{T} \sum_{t=1}^T \mathop{\mathbb{E}}_{\{\ell_s\}_{s<t}\sim\mathcal{D}^{t-1}} \left(\mathop{\mathbb{E}}_{\ell'_t\sim\mathcal{D}} \ell'_t(\theta_t) - \mathop{\mathbb{E}}_{\ell_t\sim\mathcal{D}} \ell_t(\theta_t)\right) + \frac{\mathbf{R}_T}{T} + \frac{1}{T}\sum_{t=1}^T \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}} \ell(\theta^*) \\
&= \frac{\mathbf{R}_T}{T} + \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}} \ell(\theta^*)
\end{aligned}$$

where we used the fact that θ_t only depends on $\ell_1, \ldots, \ell_{t-1}$. □

For nonnegative bounded losses we have the following fact [14, Proposition 1]:

Proposition A.2. *If a sequence of loss functions $\{\ell_t : \Theta \mapsto [0,1]\}_{t\in[T]}$ drawn i.i.d. from some distribution $\mathcal{D}$ is given to an online algorithm that generates a sequence of actions $\{\theta_t \in \Theta\}_{t\in[T]}$ then*

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^T \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}} \ell(\theta_t) &\leq \frac{1}{T}\sum_{t=1}^T \ell_t(\theta_t) + \sqrt{\frac{2}{T}\log\frac{1}{\delta}} \qquad \textit{w.p. } 1-\delta \\
\frac{1}{T}\sum_{t=1}^T \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}} \ell(\theta_t) &\geq \frac{1}{T}\sum_{t=1}^T \ell_t(\theta_t) - \sqrt{\frac{2}{T}\log\frac{1}{\delta}} \qquad \textit{w.p. } 1-\delta
\end{aligned}$$

Note that Cesa-Bianchi et al. [14] only prove the first inequality; the second follows via the same argument but applying the symmetric version of the Azuma-Hoeffding inequality [4]. The inequalities above can be easily used to derive the following competitive bounds:

Corollary A.1. *If a sequence of loss functions $\{\ell_t : \Theta \mapsto [0,1]\}_{t\in[T]}$ drawn i.i.d. from some distribution $\mathcal{D}$ is given to an online algorithm with regret bound $\mathbf{R}_T$ that generates a sequence of actions $\{\theta_t \in \Theta\}_{t\in[T]}$ then*

$$\mathop{\mathbb{E}}_{t\sim\mathcal{U}[T]}\mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\theta_t) \le \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\theta^*) + \frac{\mathbf{R}_T}{T} + \sqrt{\frac{8}{T}\log\frac{1}{\delta}} \qquad \textit{w.p. } 1-\delta$$

for any $\theta^* \in \Theta$. If the losses are also convex then for $\bar\theta = \frac{1}{T}\theta_{1:T}$ we have

$$\mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\bar\theta) \le \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\theta^*) + \frac{\mathbf{R}_T}{T} + \sqrt{\frac{8}{T}\log\frac{1}{\delta}} \qquad \textit{w.p. } 1-\delta$$

Proof. By Proposition A.2 we have

$$\frac{1}{T}\sum_{t=1}^T \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\theta_t) \le \frac{1}{T}\sum_{t=1}^T \ell_t(\theta^*) + \frac{\mathbf{R}_T}{T} + \sqrt{\frac{2}{T}\log\frac{1}{\delta}} \le \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\theta^*) + \frac{\mathbf{R}_T}{T} + \sqrt{\frac{8}{T}\log\frac{1}{\delta}}$$

Apply linearity of expectations to get the first inequality and Jensen's inequality to get the second. □

We now discuss some stronger guarantees for certain classes of loss functions. The first, due to Kakade and Tewari [33, Theorem 2], yields faster rates for strongly convex losses:

Theorem A.4. *Let $\mathcal{D}$ be some distribution over loss functions $\ell : \Theta \mapsto [0,B]$ for some $B > 0$ that are G-Lipschitz w.r.t. $\|\cdot\|$ for some $G > 0$ and α-strongly-convex w.r.t $\|\cdot\|$ for some $\alpha > 0$. If a sequence of loss functions $\{\ell_t\}_{t\in[T]}$ is drawn i.i.d. from $\mathcal{D}$ and given to an online algorithm with regret bound $\mathbf{R}_T$ that generates a sequence of actions $\{\theta_t \in \Theta\}_{t\in[T]}$ then w.p. $1-\delta$ we have for $\bar\theta = \frac{1}{T}\theta_{1:T}$ and any $\theta^* \in \Theta$ that*

$$\mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\bar\theta) \le \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\theta^*) + \frac{\mathbf{R}_T}{T} + \frac{4G}{T}\sqrt{\frac{\mathbf{R}_T}{\alpha}\log\frac{4\log T}{\delta}} + \frac{\max\{16G^2, 6\alpha B\}}{\alpha T}\log\frac{4\log T}{\delta}$$

We can also obtain a data-dependent bound using a result of Zhang [54] under a self-bounding property. Cesa-Bianchi and Gentile [13, Proposition 2] show a similar but less general result.

Definition A.11. *A distribution $\mathcal{D}$ over $\ell : \Theta \mapsto \mathbb{R}$ has ρ-**self-bounding** losses if $\forall\ \theta \in \Theta$ we have*

$$\rho\mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\theta) \ge \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}(\ell(\theta) - \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\theta))^2$$

Theorem A.5. *Let $\mathcal{D}$ be some distribution over ρ-self-bounding convex loss functions $\ell : \Theta \mapsto [-1,1]$ for some $\rho > 0$. If a sequence of loss functions $\{\ell_t\}_{t\in[T]}$ is drawn i.i.d. from $\mathcal{D}$ and given to an online algorithm with regret bound $\mathbf{R}_T$ that generates a sequence of actions $\{\theta_t \in \Theta\}_{t\in[T]}$ then w.p. $1-\delta$ we have*

$$\mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\bar\theta) \le \bar L_T + \sqrt{\frac{2\rho\max\{0, \bar L_T\}}{T}\log\frac{1}{\delta}} + \frac{3\rho+2}{T}\log\frac{1}{\delta}$$

where $\bar\theta = \frac{1}{T}\theta_{1:T}$ and $\bar L_T = \frac{1}{T}\sum_{t=1}^T \ell_t(\theta_t)$ is the average loss suffered by the agent.

Proof. Apply Jensen's inequality and Zhang [54, Theorem 4]. □

Note that nonnegative 1-bounded convex losses satisfy the conditions of Theorem A.5 with $\rho = 1$. However, we are interested in a different result that can yield a data-dependent competitive bound:

Corollary A.2. *Let $\mathcal{D}$ be some distribution over convex loss functions $\ell : \Theta \mapsto [0,1]$ such that the functions $\ell(\theta) - \ell(\theta^*)$ are ρ-self-bounded for some $\theta^* \in \arg\min_{\theta\in\Theta}\mathbb{E}_{\ell\sim\mathcal{D}}\ell(\theta)$. If a sequence of loss functions $\{\ell_t\}_{t\in[T]}$ is drawn i.i.d. from $\mathcal{D}$ and given to an online algorithm with regret bound $\mathbf{R}_T$ that generates a sequence of actions $\{\theta_t \in \Theta\}_{t\in[T]}$ then w.p. $1-\delta$ we have*

$$\mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\bar\theta) \le \mathop{\mathbb{E}}_{\ell\sim\mathcal{D}}\ell(\theta^*) + \frac{\mathbf{R}_T}{T} + \frac{1}{T}\sqrt{2\rho\,\mathbf{R}_T\log\frac{1}{\delta}} + \frac{3\rho+2}{T}\log\frac{1}{\delta}$$

where $\bar\theta = \frac{1}{T}\theta_{1:T}$ and $\mathcal{E}^* = \arg\min_{\theta\in\Theta}\mathbb{E}\,\ell(\theta)$.

Proof. Apply Theorem A.5 over the sequence of functions $\{\ell_t(\theta) - \ell_t(\theta^*)\}_{t\in[T]}$ and by definition of regret substitute $\bar L_T = \frac{1}{T}\sum_{t=1}^T \ell_t(\theta) - \ell_t(\theta^*) \le \frac{\mathbf{R}_T}{T}$. □

Zhang [54, Lemma 7] shows that the conditions are satisfied for $\rho = 4$ by least-squares regression.

A.4 Dynamic Regret Guarantees

Here we review several results for optimizing dynamic regret. We first define this quantity:

Definition A.12. *The* **dynamic regret** *of an agent playing actions* $\{\theta_t \in \Theta\}_{t\in[T]}$ *on a sequence of loss functions* $\{\ell_t : \Theta \mapsto \mathbb{R}\}$ *w.r.t. a sequence of reference parameters* $\Psi = \{\psi_t\}_{t\in[T]}$ *is*

$$\mathbf{R}_T(\Psi) = \sum_{t=1}^{T} \ell_t(\theta_t) - \sum_{t=1}^{T} \ell_t(\psi_t)$$

Mokhtari et al. [42, Corollary 1] show the following guarantee for OGD over strongly convex functions:

Theorem A.6. *Let* $\{\ell_t : \Theta \mapsto \mathbb{R}\}_{t\in[T]}$ *be a sequence of* α*-strongly-convex,* β*-strongly-smooth, and* G*-Lipschitz functions w.r.t.* $\|\cdot\|_2$*. Then OGD with step-size* $\eta \leq \frac{1}{\beta}$ *achieves dynamic regret*

$$\mathbf{R}_T(\Psi) \leq \frac{GD}{1-\rho}\left(1 + \sum_{t=2}^{T} \|\psi_t - \psi_{t-1}\|_2\right)$$

w.r.t. reference sequence $\Psi = \{\psi_t\}_{t\in[T]}$ *for* $\rho = \sqrt{1 - \frac{h\alpha}{\eta}}$ *for any* $h \in (0,1]$ *and* D *the* ℓ_2*-diameter of* Θ*.*

B Strongly Convex Coupling

Our first result is a simple trick that we believe may be of independent interest. It allows us to bound the regret of FTL on any (possibly non-convex) sequence of Lipschitz functions so long as the actions played are identical to those played on a different strongly-convex sequence of Lipschitz functions. The result is formalized in Theorem B.1.

B.1 Derivation

We start with some standard facts about convex functions.

Claim B.1. *Let $f : S \mapsto \mathbb{R}$ be an everywhere sub-differentiable convex function. Then for any norm $\|\cdot\|$ we have*

$$f(x) - f(y) \leq \|\nabla f(x)\|_* \|x - y\| \ \forall\ x, y \in S$$

Claim B.2. *Let $f : S \mapsto \mathbb{R}$ be α-strongly-convex w.r.t. $\|\cdot\|$ with minimum $x^* \in \arg\min_{x \in S} f(x)$. Then x^* is unique and for all $x \in S$ we have*

$$f(x) \geq f(x^*) + \frac{\alpha}{2}\|x - x^*\|^2$$

Next we state some technical results, starting with the well-known be-the-leader lemma [48, Lemma 2.1].

Lemma B.1. *Let $\theta_1, \dots, \theta_{T+1} \in \Theta$ be the sequence of actions of FTL on the function sequence $\{\ell_t : \Theta \mapsto \mathbb{R}\}_{t \in [T]}$. Then*

$$\sum_{t=1}^{T} \ell_t(\theta_t) - \ell_t(\theta^*) \leq \sum_{t=1}^{T} \ell_t(\theta_t) - \ell_t(\theta_{t+1})$$

for all $\theta^* \in \Theta$.

The final result depends on a stability argument for FTL on strongly-convex functions adapted from Saha et al. [47]:

Lemma B.2. *Let $\{\ell_t : \Theta \mapsto \mathbb{R}\}_{t \in [T]}$ be a sequence of functions that are α_t-strongly-convex w.r.t. $\|\cdot\|$ and let $\theta_1, \dots, \theta_{T+1} \in \Theta$ be the corresponding sequence of actions of FTL. Then*

$$\|\theta_t - \theta_{t+1}\| \leq \frac{2\|\nabla_t\|_*}{\alpha_t + 2\alpha_{1:t-1}}$$

for all $t \in [T]$.

Proof. The proof slightly generalizes an argument in Saha et al. [47, Theorem 6]. For each $t \in [T]$ we have by Claim B.2 and the $\alpha_{1:t}$-strong-convexity of $\sum_{s=1}^{t} \ell_s(\cdot)$ that

$$\sum_{s=1}^{t} \ell_s(\theta_t) \geq \sum_{s=1}^{t} \ell_s(\theta_{t+1}) + \frac{\alpha_{1:t}}{2}\|\theta_t - \theta_{t+1}\|^2$$

We similarly have

$$\sum_{s=1}^{t-1} \ell_s(\theta_{t+1}) \geq \sum_{s=1}^{t-1} \ell_s(\theta_t) + \frac{\alpha_{1:t-1}}{2}\|\theta_{t+1} - \theta_t\|^2$$

Adding these two inequalities and applying Claim B.1 yields

$$\left(\frac{\alpha_t}{2} + \alpha_{1:t-1}\right)\|\theta_t - \theta_{t+1}\|^2 \leq \ell_t(\theta_t) - \ell_t(\theta_{t+1}) \leq \|\nabla_t\|_* \|\theta_t - \theta_{t+1}\|$$

Dividing by $\|\theta_t - \theta_{t+1}\|$ yields the result. $\square$

Theorem B.1. *Let $\{\ell_t : \Theta \mapsto \mathbb{R}\}_{t\in[T]}$ be a sequence of functions that are G_t-Lipschitz in $\|\cdot\|_A$ and let $\theta_1, \dots, \theta_{T+1}$ be the sequence of actions produced by FTL. Let $\{\ell'_t : \Theta \mapsto \mathbb{R}\}_{t\in[T]}$ be a sequence of functions on which FTL also plays $\theta_1, \dots, \theta_{T+1}$ but which are G'_t-Lipschitz and α_t-strongly-convex in $\|\cdot\|_B$. Then*

$$\sum_{t=1}^{T} \ell_t(\theta_t) - \ell_t(\theta^*) \leq 2C \sum_{t=1}^{T} \frac{G_t G'_t}{\alpha_t + 2\alpha_{1:t-1}}$$

for all $\theta^* \in \Theta$ and some constant C s.t. $\|\theta\|_A \leq C\|\theta\|_B \;\forall\, \theta \in \Theta$. If the functions ℓ_t are also convex then we have

$$\sum_{t=1}^{T} \ell_t(\theta_t) - \ell_t(\theta^*) \leq 2C \sum_{t=1}^{T} \frac{\|\nabla_t\|_{A,*}\|\nabla'_t\|_{B,*}}{\alpha_t + 2\alpha_{1:t-1}}$$

or all $\theta^* \in \Theta$

Proof. By Lemma B.2,

$$\|\theta_t - \theta_{t+1}\|_A \leq C\|\theta_t - \theta_{t+1}\|_B \leq \frac{2CG'_t}{\alpha_t + 2\alpha_{1:t-1}}$$

for all $t \in [T]$. Then by Lemma B.1 and the G_t-Lipschitzness of ℓ_t we have for all $\theta^* \in \Theta$ that

$$\sum_{t=1}^{T} \ell_t(\theta_t) - \ell_t(\theta^*) \leq \sum_{t=1}^{T} \ell_t(\theta_t) - \ell(\theta_{t+1}) \leq \sum_{t=1}^{T} G_t\|\theta_t - \theta_{t+1}\|_A \leq 2C \sum_{t=1}^{T} \frac{G_t G'_t}{\alpha_t + 2\alpha_{1:t-1}}$$

In the convex case we instead apply Claim B.1 and Lemma B.2 to get

$$\sum_{t=1}^{T} \ell_t(\theta_t) - \ell_t(\theta^*) \leq \sum_{t=1}^{T} \ell_t(\theta_t) - \ell(\theta_{t+1}) \leq \sum_{t=1}^{T} \|\nabla_t\|_{A,*}\|\theta_t - \theta_{t+1}\|_A \leq 2C \sum_{t=1}^{T} \frac{\|\nabla_t\|_{A,*}\|\nabla'_t\|_{B,*}}{\alpha_t + 2\alpha_{1:t-1}}$$

□

B.2 Applications

We now show two applications of strongly convex coupling. The first shows logarithmic regret for FTL run on a sequence of Bregman regularizers. Note that these functions are nonconvex in general.

Proposition B.1. *Let $R : \Theta \mapsto \mathbb{R}$ be 1-strongly-convex w.r.t. $\|\cdot\|$ and consider any $\theta_1, \dots, \theta_T \in \Theta$. Then when run on the loss sequence $\alpha_1\mathcal{B}_R(\theta_1||\cdot), \dots, \alpha_T\mathcal{B}_R(\theta_T||\cdot)$ for any positive scalars $\alpha_1, \dots, \alpha_T \in \mathbb{R}_+$, FTL obtains regret*

$$\mathbf{R}_T \leq 2CD \sum_{t=1}^{T} \frac{\alpha_t^2 G_t}{\alpha_t + 2\alpha_{1:t-1}}$$

for C s.t. $\|\theta\| \leq C\|\theta\|_2 \;\forall\, \theta \in \Theta$, $D = \max_{\theta,\phi\in\Theta} \|\theta - \phi\|_2$ the ℓ_2-diameter of Θ, and G_t the Lipschitz constant of $\mathcal{B}_R(\theta_t||\cdot)$ over Θ w.r.t. $\|\cdot\|$. Note that for $\|\cdot\| = \|\cdot\|_2$ we have $C = 1$ and $G_t \leq D \;\forall\, t \in [T]$.

Proof. Note that $\alpha_t\mathcal{B}_R(\theta_t||\cdot)$ is $\alpha_t G_t$-Lipschitz w.r.t. $\|\cdot\|$. Let $R'(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, so $\mathcal{B}_{R'}(\theta_t||\phi) = \frac{1}{2}\|\theta_t - \phi\|_2^2 \;\forall\, \phi \in \Theta, t \in [T]$. The function $\alpha_t\mathcal{B}_{R'}(\theta_t||\cdot)$ is thus α_t-strongly-convex and D-Lipschitz w.r.t. $\|\cdot\|_2$. Now by Claim A.1 FTL run on this new sequence plays the same actions as FTL run on the original sequence. Applying Theorem B.1 yields the result. □

In the next application we use coupling to give a $\tilde{\mathcal{O}}(T^{\frac{3}{5}})$-regret algorithm for a sequence of non-Lipschitz convex functions.

Proposition B.2. *Let $\{\ell_t : \mathbb{R}_+ \mapsto \mathbb{R}\}_{t\geq 1}$ be a sequence of functions of form $\ell_t(x) = \left(\frac{B_t^2}{x} + x\right)\alpha_t$ for any positive scalars $\alpha_1, \dots, \alpha_T \in \mathbb{R}_+$ and adversarially chosen $B_t \in [0, D]$. Then the ε-FTL algorithm, which for $\varepsilon > 0$ uses the actions of FTL run on the functions $\tilde{\ell}_t(x) = \left(\frac{B_t^2+\varepsilon^2}{x} + x\right)\alpha_t$ over the domain $[\varepsilon, \sqrt{D^2+\varepsilon^2}]$ to determine x_t, achieves regret*

$$\mathbf{R}_T \leq \min\left\{\frac{\varepsilon^2}{x^*}, \varepsilon\right\}\alpha_{1:T} + 2D\max\left\{\frac{D^3}{\varepsilon^3}, 1\right\}\sum_{t=1}^T \frac{\alpha_t^2}{\alpha_t + 2\alpha_{1:t-1}}$$

for all $x^* > 0$.

Proof. Define $\tilde{B}_t^2 = B_t^2 + \varepsilon^2$ and note that FTL run on the functions $\tilde{\ell}_t'(x) = \left(\frac{x^2}{2} - \tilde{B}_t^2 \log x\right)\alpha_t$ plays the exact same actions $x_t^2 = \frac{\sum_{s<t}\alpha_s \tilde{B}_s^2}{\alpha_{1:t-1}}$ as FTL run on $\tilde{\ell}_t$. We have that

$$|\partial_x \tilde{\ell}_t| = \alpha_t \left|1 - \frac{\tilde{B}_t^2}{x^2}\right| \leq \frac{\alpha_t D^2}{\varepsilon^2}$$

$$|\partial_x \tilde{\ell}_t'| = \alpha_t \left|x - \frac{\tilde{B}_t^2}{x}\right| \leq \alpha_t \max\left\{D, \frac{D^2}{\varepsilon}\right\} \qquad \partial_{xx}\tilde{\ell}_t' = \alpha_t\left(1 + \frac{\tilde{B}_t^2}{x^2}\right) \geq \alpha_t$$

so the functions $\tilde{\ell}_t$ are $\frac{\alpha_t D^2}{\varepsilon^2}$-Lipschitz while the functions $\tilde{\ell}_t'$ are $\alpha_t D \max\left\{\frac{D}{\varepsilon}, 1\right\}$-Lipschitz and α_t-strongly-convex. Therefore by Theorem B.1 we have that

$$\sum_{t=1}^T \tilde{\ell}_t(x_t) - \tilde{\ell}_t(x^*) \leq 2D\max\left\{\frac{D^3}{\varepsilon^3}, 1\right\}\sum_{t=1}^T \frac{\alpha_t^2}{\alpha_t + 2\alpha_{1:t-1}}$$

for any $x^* \in [\varepsilon, \sqrt{D^2+\varepsilon^2}]$. Since $\sum_{t=1}^T \tilde{\ell}_t$ is minimized on $[\varepsilon, \sqrt{D^2+\varepsilon^2}]$, the above also holds for all $x^* > 0$. Therefore we have that

$$\begin{aligned}
\sum_{t=1}^T \ell_t(x_t) &\leq \sum_{t=1}^T \left(\frac{B_t^2+\varepsilon^2}{x_t} + x_t\right)\alpha_t \\
&= \sum_{t=1}^T \tilde{\ell}_t(x_t) \\
&\leq \min_{x^*>0} 2D\max\left\{\frac{D^3}{\varepsilon^3}, 1\right\}\sum_{t=1}^T \frac{\alpha_t^2}{\alpha_t + 2\alpha_{1:t-1}} + \sum_{t=1}^T \tilde{\ell}_t(x^*) \\
&= \min_{x^*>0} 2D\max\left\{\frac{D^3}{\varepsilon^3}, 1\right\}\sum_{t=1}^T \frac{\alpha_t^2}{\alpha_t + 2\alpha_{1:t-1}} + \sum_{t=1}^T \left(\frac{B_t^2+\varepsilon^2}{x^*} + x^*\right)\alpha_t \\
&= \min_{x^*>0} \frac{\varepsilon^2}{x^*}\alpha_{1:T} + 2D\max\left\{\frac{D^3}{\varepsilon^3}, 1\right\}\sum_{t=1}^T \frac{\alpha_t^2}{\alpha_t + 2\alpha_{1:t-1}} + \sum_{t=1}^T \ell_t(x^*)
\end{aligned}$$

Note that substituting $x^* = \sqrt{\frac{\sum_{t=1}^T \alpha_t \tilde{B}_t^2}{\alpha_{1:T}}}$ into the second-to-last line yields

$$\min_{x^*>0}\sum_{t=1}^T \left(\frac{B_t^2+\varepsilon^2}{x^*} + x^*\right)\alpha_t \leq 2\sqrt{\alpha_{1:T}\sum_{t=1}^T \alpha_t \tilde{B}_t^2} \leq 2\varepsilon\alpha_{1:T} + \min_{x^*>0}\sum_{t=1}^T \ell_t(x^*)$$

completing the proof. ∎

C Adaptive and Dynamic Guarantees

Throughout Appendices C, D, and E we assume that $\arg\min_{\theta\in\Theta}\sum_{\ell\in\mathcal{S}}\ell(\theta)$ returns a unique minimizer of the sum of the loss functions in the sequence $\mathcal{S}$. Formally, this can be defined to be the one minimizing an appropriate Bregman divergence $\mathcal{B}_R(\cdot|\phi_R)$ from some fixed $\phi_R\in\Theta$, e.g. the origin in Euclidean space or the uniform distribution over the simplex, which is unique by strong-convexity of $\mathcal{B}_R(\cdot|\phi_R)$ and convexity of the set of optimizers of a convex function.

Theorem C.1. *Let each task $t\in[T]$ consist of a sequence of m_t convex loss functions $\ell_{t,i}:\Theta\mapsto\mathbb{R}$ that are $G_{t,i}$-Lipschitz w.r.t. $\|\cdot\|$. For $G_t^2=G_{1:m_t}^2/m_t$ and $R:\Theta\mapsto\mathbb{R}$ a 1-strongly-convex function w.r.t. $\|\cdot\|$ define the following online algorithms:*

1. INIT*: a method that has dynamic regret $\mathbf{U}_T^{init}(\Psi)=\sum_{t=1}^T f_t^{init}(\phi_t)-f_t^{init}(\psi_t)$ w.r.t. reference actions $\Psi=\{\psi_t\}_{t=1}^T\subset\Theta$ over the sequence $f_t^{init}(\cdot)=\mathcal{B}_R(\theta_t^*||\cdot)G_t\sqrt{m_t}$.*

2. SIM*: a method that has (static) regret $\mathbf{U}_T^{sim}(x)$ decreasing in $x>0$ over the sequence of functions $f_t^{sim}(x)=\left(\frac{\mathcal{B}_R(\theta_t^*||\phi_t)}{x}+x\right)G_t\sqrt{m_t}$.*

Then if Algorithm 1 sets $\phi_t=\mathrm{INIT}(t)$ and $\eta_t=\frac{\mathrm{SIM}(t)}{G_t\sqrt{m_t}}$ it will achieve

$$\bar{\mathbf{R}}_T\le\bar{\mathbf{U}}_T\le\frac{\mathbf{U}_T^{sim}(V_\Psi)}{T}+\frac{1}{T}\min\left\{\frac{\mathbf{U}_T^{init}(\Psi)}{V_\Psi},2\sqrt{\mathbf{U}_T^{init}(\Psi)\sum_{t=1}^T G_t\sqrt{m_t}}\right\}+\frac{2V_\Psi}{T}\sum_{t=1}^T G_t\sqrt{m_t}$$

for $V_\Psi^2=\frac{1}{\sum_{t=1}^T G_t\sqrt{m_t}}\sum_{t=1}^T\mathcal{B}_R(\theta_t^*||\psi_t)G_t\sqrt{m_t}$.

Proof. Letting $x_t=\mathrm{SIM}(t)$ be the output of SIM at time t, defining $\sigma_t=G_t\sqrt{m_t}$ and $\sigma_{1:T}=\sum_{t=1}^T\sigma_t$, and substituting into the regret-upper-bound of OMD/FTRL (2), we have that

$$\begin{aligned}\bar{\mathbf{U}}_T\,T=\sum_{t=1}^T\left(\frac{\mathcal{B}_R(\theta_t^*||\phi_t)}{x_t}+x_t\right)\sigma_t&\le\min_{x>0}\mathbf{U}_T^{\mathrm{sim}}(x)+\sum_{t=1}^T\left(\frac{\mathcal{B}_R(\theta_t^*||\phi_t)}{x}+x\right)\sigma_t\\&\le\min_{x>0}\mathbf{U}_T^{\mathrm{sim}}(x)+\frac{\mathbf{U}_T^{\mathrm{init}}(\Psi)}{x}+\sum_{t=1}^T\left(\frac{\mathcal{B}_R(\theta_t^*||\psi_t)}{x}+x\right)\sigma_t\\&\le\mathbf{U}_T^{\mathrm{sim}}(V_\Psi)+\min\left\{\frac{\mathbf{U}_T^{\mathrm{init}}(\Psi)}{V_\Psi},2\sqrt{\mathbf{U}_T^{\mathrm{init}}(\Psi)\sigma_{1:T}}\right\}+2V_\Psi\sigma_{1:T}\end{aligned}$$

where the last line follows by substituting $x=\max\left\{V_\Psi,\sqrt{\frac{\mathbf{U}_T^{\mathrm{init}}(\Psi)}{\sigma_{1:T}}}\right\}$. □

Corollary C.1. *Under the assumptions of Theorem C.1 and boundedness of $\mathcal{B}_R$ over Θ, if* INIT *uses FTL, or AOGD in the case of $R(\cdot)=\frac{1}{2}\|\cdot|_2^2$, and* SIM *uses ε-FTL as defined in Proposition B.2, then Algorithm 1 achieves*

$$\bar{\mathbf{U}}_T\,T\le\min\left\{\frac{\varepsilon^2}{V},\varepsilon\right\}\sigma_{1:T}+2D\max\left\{\frac{D^3}{\varepsilon^3},1\right\}\sum_{t=1}^T\frac{\sigma_t^2}{\sigma_{1:t}}+\sqrt{8CD\sigma_{1:T}\sum_{t=1}^T\frac{\sigma_t^2}{\sigma_{1:t}}}+2V\sigma_{1:T}$$

for $V^2=\min_{\phi\in\Theta}\sum_{t=1}^T\sigma_t\mathcal{B}_R(\theta_t^*||\phi)$ and constant C the product of the constant C from Proposition B.1 and the bound on the gradient of the Bregman divergence. Assuming $\sigma_t=G\sqrt{m}\ \forall\ t$ and substituting $\varepsilon=\frac{1}{\sqrt[5]{T}}$ yields

$$\bar{\mathbf{R}}_T\le\bar{\mathbf{U}}_T=\tilde{\mathcal{O}}\left(\min\left\{\frac{1}{VT^{\frac{2}{5}}}+\frac{1}{\sqrt{T}},\frac{1}{\sqrt[5]{T}}\right\}+V\right)\sqrt{m}$$

Proof. Substitute Propositions B.1 and B.2 into Theorem C.1. □

Proposition C.1. *Let $\{\ell_t : \mathbb{R}_+ \mapsto \mathbb{R}\}_{t\geq 1}$ be a sequence of functions of form $\ell_t(x) = \left(\frac{B_t^2}{x} + x\right)\alpha_t$ for any positive scalars $\alpha_1, \dots, \alpha_T \in \mathbb{R}_+$ and adversarially chosen $B_t \in [0, D]$. Then the losses $\tilde{\ell}_t(x) = \left(\frac{B_t^2+\varepsilon^2}{x} + x\right)\alpha_t$ over the domain $[\varepsilon, \sqrt{D^2+\varepsilon^2}]$ are $\frac{\alpha_t D^2}{\varepsilon^2}$-Lipschitz and $\frac{2}{\alpha_t D}\min\left\{\frac{\varepsilon^2}{D^2}, 1\right\}$-exp-concave.*

Proof. Lipschitzness follows by taking derivatives as in Proposition B.2. Define $\tilde{B}_t^2 = B_t^2 + \varepsilon^2$. We then have

$$\partial_x \tilde{\ell}_t = \alpha_t\left(1 - \frac{\tilde{B}_t^2}{x^2}\right) \qquad\qquad \partial_{xx}\tilde{\ell}_t = \frac{2\alpha_t \tilde{B}_t^2}{x^3}$$

The γ-exp-concavity of the functions $\tilde{\ell}_t$ can be determined by finding the largest γ satisfying

$$\gamma \leq \frac{\partial_{xx}\tilde{\ell}_t}{(\partial_x \tilde{\ell}_t)^2} = \frac{2\tilde{B}_t^2 x}{\alpha_t(\tilde{B}_t^2 - x^2)^2}$$

for all $x \in [\varepsilon, \sqrt{D^2+\varepsilon^2}]$ and all $t \in [T]$. We first minimize jointly over choice of $x, \tilde{B}_t \in [\varepsilon, \sqrt{D^2+\varepsilon^2}]$. The derivatives of the objective w.r.t. x and $\tilde{B}_t$, respectively, are

$$\frac{2\tilde{B}_t^2(\tilde{B}_t^2 + 3x^2)}{(\tilde{B}_t^2 - x^2)^3} \qquad\qquad -\frac{4\tilde{B}_t x(\tilde{B}_t^2 + x^2)}{(\tilde{B}_t^2 - x^2)^3}$$

Note that the objective approaches ∞ as the coordinates approach the line $x = \tilde{B}_t$. For $x < \tilde{B}_t$ the derivative w.r.t. x is always positive while the derivative w.r.t. $\tilde{B}_t$ is always negative. Since we have the constraints $x \geq \varepsilon$ and $\tilde{B}_t^2 \leq D^2 + \varepsilon^2$, the optimum over $x < \tilde{B}_t$ is thus attained at $x = \varepsilon$ and $\tilde{B}_t^2 = D^2 + \varepsilon^2$. Substituting into the original objective yields

$$\frac{2(D^2+\varepsilon^2)\varepsilon}{\alpha_t D^4} \geq \frac{2\varepsilon}{\alpha_t D^2}$$

For $x > \tilde{B}_t$ the derivative w.r.t. x is always negative while the derivative w.r.t. $\tilde{B}_t$ is always positive. Since we have the constraints $x \leq \sqrt{D^2+\varepsilon^2}$ and $\tilde{B}_t^2 \geq \varepsilon^2$, the optimum over $x > \tilde{B}_t$ is thus attained at $x = \sqrt{D^2+\varepsilon^2}$ and $\tilde{B}_t^2 = \varepsilon^2$. Substituting into the original objective yields

$$\frac{2\varepsilon^2\sqrt{D^2+\varepsilon^2}}{\alpha_t D^4} \geq \frac{2\varepsilon^2}{\alpha_t D^3}$$

Thus we have that the functions $\tilde{\ell}_t$ are $\frac{2}{\alpha_t D}\min\left\{\frac{\varepsilon^2}{D^2}, 1\right\}$-exp-concave. □

Corollary C.2. *Let $\{\ell_t : \mathbb{R}_+ \mapsto \mathbb{R}\}_{t\geq 1}$ be a sequence of functions of form $\ell_t(x) = \left(\frac{B_t^2}{x} + x\right)\alpha_t$ for any positive scalars $\alpha_1, \dots, \alpha_T \in \mathbb{R}_+$ and adversarially chosen $B_t \in [0, D]$. Then the ε-EWOO algorithm, which for $\varepsilon > 0$ uses the actions of EWOO run on the functions $\tilde{\ell}_t(x) = \left(\frac{B_t^2+\varepsilon^2}{x} + x\right)\alpha_t$ over the domain $[\varepsilon, \sqrt{D^2+\varepsilon^2}]$ to determine x_t, achieves regret*

$$\mathbf{R}_T \leq \min_{x^*>0}\left\{\frac{\varepsilon^2}{x^*}, \varepsilon\right\}\alpha_{1:T} + \frac{D\alpha_{\max}}{2}\max\left\{\frac{D^2}{\varepsilon^2}, 1\right\}(1 + \log(T+1))$$

for all $x^* > 0$.

Proof. Since $\sum_{t=1}^T \tilde{\ell}_t$ is minimized on $[\varepsilon, \sqrt{D^2+\varepsilon^2}]$, we apply Theorem A.3 and follow a similar argument to that concluding Proposition B.2 to get

$$\begin{aligned}\sum_{t=1}^T \ell_t(x_t) &\leq \frac{D\alpha_{\max}}{2}\max\left\{\frac{D^2}{\varepsilon^2}, 1\right\}(1 + \log(T+1)) + \sum_{t=1}^T \tilde{\ell}_t(x^*) \\ &= \min_{x^*>0}\left\{\frac{\varepsilon^2}{x^*}, \varepsilon\right\}\alpha_{1:T} + \frac{D\alpha_{\max}}{2}\max\left\{\frac{D^2}{\varepsilon^2}, 1\right\}(1 + \log(T+1)) + \sum_{t=1}^T \ell_t(x^*)\end{aligned}$$

□

Corollary C.3. *Under the assumptions of Theorem C.1 and boundedness of $\mathcal{B}_R$ over Θ, if* INIT *uses FTL, or AOGD in the case of $R(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, and* SIM *uses ε-EWOO as defined in Proposition C.2, then Algorithm 1 achieves*

$$\bar{\mathbf{U}}_T T \leq \min\left\{\frac{\varepsilon^2}{V}, \varepsilon\right\}\sigma_{1:T} + \frac{D\sigma_{\max}}{2}\max\left\{\frac{D^2}{\varepsilon^2}, 1\right\}(1+\log(T+1)) + \sqrt{8CD\sigma_{1:T}\sum_{t=1}^T \frac{\sigma_t^2}{\sigma_{1:t}}} + 2V\sigma_{1:T}$$

for $V^2 = \min_{\phi\in\Theta}\sum_{t=1}^T \sigma_t \mathcal{B}_R(\theta_t^*||\phi)$ and constant C the product of the constant C from Proposition B.1 and the bound on the gradient of the Bregman divergence. Assuming $\sigma_t = G\sqrt{m}\ \forall\, t$ and substituting $\varepsilon = \frac{1}{\sqrt[4]{T}}$ yields

$$\bar{\mathbf{R}}_T \leq \bar{\mathbf{U}}_T = \tilde{\mathcal{O}}\left(\min\left\{\frac{1+\frac{1}{V}}{\sqrt{T}}, \frac{1}{\sqrt[4]{T}}\right\} + V\right)\sqrt{m}$$

Proof. Substitute Proposition B.1 and Corollary C.2 into Theorem C.1. □

Corollary C.4. *Under the assumptions of Theorem 3.1 and boundedness of Θ, if* INIT *is OGD with learning rate $\frac{1}{\sigma_{\max}}$ and* SIM *uses ε-EWOO as defined in Proposition C.2 then Algorithm 1 achieves*

$$\begin{aligned}\bar{\mathbf{U}}_T T \leq \min\left\{\frac{\varepsilon^2}{V_\Psi}, \varepsilon\right\}\sigma_{1:T} + \frac{D\sigma_{\max}}{2}\max\left\{\frac{D^2}{\varepsilon^2}, 1\right\}(1+\log(T+1)) \\ + 2D\min\left\{\frac{D\sigma_{\max}}{V_\Psi}(1+P_\Psi), \sqrt{2\sigma_{\max}\sigma_{1:T}(1+P_\Psi)}\right\} + 2V_\Psi\sigma_{1:T}\end{aligned}$$

for $P_T(\Psi) = \sum_{t=2}^T \|\psi_t - \psi_{t-1}\|_2$. Assuming $\sigma_t = G\sqrt{m}\ \forall\, t$ and substituting $\varepsilon = \frac{1}{\sqrt[4]{T}}$ yields

$$\bar{\mathbf{R}}_T \leq \bar{\mathbf{U}}_T = \tilde{\mathcal{O}}\left(\min\left\{\frac{1+\frac{1}{V_\Psi}}{\sqrt{T}}, \frac{1}{\sqrt[4]{T}}\right\} + \min\left\{\frac{1+P_\Psi}{V_\Psi T}, \sqrt{\frac{1+P_\Psi}{T}}\right\} + V_\Psi\right)\sqrt{m}$$

Proof. Substitute Theorem 3.3 and Corollary C.2 into Theorem C.1. □

D Adapting to the Inter-Task Geometry

For clarity, vectors and matrices in this section will be **bolded**, although scalar regret quantities will continue to be as well. For any two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, $\boldsymbol{x} \odot \boldsymbol{y}$ will denote element-wise multiplication, $\frac{\boldsymbol{x}}{\boldsymbol{y}}$ will denote element-wise division, $\boldsymbol{x}^p$ will denote raising each element of $\boldsymbol{x}$ to the power p, and $\max\{\boldsymbol{x}, \boldsymbol{y}\}$ and $\min\{\boldsymbol{x}, \boldsymbol{y}\}$ will denote element-wise maximum and minimum, respectively. For any nonnegative $\boldsymbol{a} \in \mathbb{R}^d$ we will use the notation $\|\cdot\|_{\boldsymbol{a}} = \langle\sqrt{\boldsymbol{a}}, \cdot\rangle$; note that if all elements of $\boldsymbol{a}$ are positive then $\|\cdot\|_{\boldsymbol{a}}$ is a norm on $\mathbb{R}^d$ with dual norm $\|\cdot\|_{\boldsymbol{a}^{-1}}$.

Claim D.1. *For $t \geq 1$ and $p \in (0,1)$ we have*

$$\sum_{s=0}^{t-1} \frac{1}{(s+1)^p} \geq \sum_{s=1}^{t} \frac{1}{(s+1)^p} \geq \underline{c}_p t^{1-p} \qquad \textit{and} \qquad \sum_{s=1}^{t} \frac{1}{s^p} \leq \overline{c}_p t^{1-p}$$

for $\underline{c}_p = \frac{1-\left(\frac{2}{3}\right)^{1-p}}{1-p}$ *and* $\overline{c}_p = \frac{1}{1-p}$.

Proof.

$$\sum_{s=0}^{t-1} \frac{1}{(s+1)^p} \geq \sum_{s=1}^{t} \frac{1}{(s+1)^p} \geq \int_1^{t+1} \frac{ds}{(s+1)^p} = \frac{(t+2)^{1-p} - 2^{1-p}}{1-p} \geq \underline{c}_p (t+2)^{1-p} \geq \underline{c}_p t^{1-p}$$

$$\sum_{s=1}^{t} \frac{1}{s^p} \leq 1 + \int_1^t \frac{ds}{s^p} = 1 + \frac{t^{1-p} - 1}{1-p} \leq \overline{c}_p t^{1-p}$$

$\square$

Claim D.2. *For any $\boldsymbol{x} \in \mathbb{R}^d$ we have $\|\boldsymbol{x}^2\|_2^2 \leq \|\boldsymbol{x}\|_2^4$.*

Proof.

$$\|\boldsymbol{x}^2\|_2^2 = \sum_{j=1}^d x_j^4 \leq \left(\sum_{j=1}^d x_j^2\right)^2 = \|\boldsymbol{x}\|_2^4$$

$\square$

We now review some facts from matrix analysis. Throughout this section we will use matrices in $\mathbb{R}^{d\times d}$; we denote the subset of symmetric matrices by $\mathbb{S}^d$, the subset of symmetric PSD matrices by $\mathbb{S}_+^d$, and the subset of symmetric positive-definite matrices by $\mathbb{S}_{++}^d$. Note that every symmetric matrix $\boldsymbol{A} \in \mathbb{S}^d$ has diagonalization $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$ for diagonal matrix $\boldsymbol{\Lambda} \in \mathbb{S}^d$ containing the eigenvalues of $\boldsymbol{A}$ along the diagonal and a matrix $\boldsymbol{V} \in \mathbb{R}^{d\times d}$ of orthogonal eigenvectors. For such matrices we will use $\lambda_j(\boldsymbol{A})$ to denote the jth largest eigenvalue of $\boldsymbol{A}$ and for any function $f : [\lambda_d(\boldsymbol{A}), \lambda_1(\boldsymbol{A})] \mapsto \mathbb{R}$ we will use the notation

$$f(\boldsymbol{A}) = \boldsymbol{V} \begin{pmatrix} f(\boldsymbol{\Lambda}_{11}) & & \\ & \ddots & \\ & & f(\boldsymbol{\Lambda}_{dd}) \end{pmatrix} \boldsymbol{V}^{-1}$$

We will denote the spectral norm by $\|\cdot\|_2$ and the Frobenius norm by $\|\cdot\|_F$.

Claim D.3. *[9, Section A.4.1] $f(\boldsymbol{X}) = \log\det \boldsymbol{X}$ has gradient $\nabla_{\boldsymbol{X}} f = \boldsymbol{X}^{-1}$ over $\mathbb{S}_{++}^d$*

Claim D.4. *[43, Theorem 3.1] The function $f(\boldsymbol{X}) = -\log\det \boldsymbol{X}$ is $\frac{1}{\sigma^2}$-strongly-convex w.r.t. $\|\cdot\|_2$ over the set of symmetric positive-definite matrices with spectral norm bounded by σ.*

Definition D.1. *A function* $f : (0, \infty) \mapsto \mathbb{R}$ *is* **operator convex** *if* $\forall\, \boldsymbol{X}, \boldsymbol{Y} \in \mathbb{S}^d_{++}$ *and any* $t \in [0, 1]$ *we have*

$$f(t\boldsymbol{X} + (1-t)\boldsymbol{Y}) \preceq tf(\boldsymbol{X}) + (1-t)f(\boldsymbol{Y})$$

Claim D.5. *If* $\boldsymbol{A} \in \mathbb{S}^d_+$ *and* $f : (0, \infty) \mapsto \mathbb{R}$ *is operator convex then* $\mathrm{Tr}(\boldsymbol{A}f(\boldsymbol{X}))$ *is convex on* $\mathbb{S}^d_{++}$.

Proof. Consider any $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{S}^d_{++}$ and any $t \in [0, 1]$. By the operator convexity of f, positive semi-definiteness of $\boldsymbol{A}$, and linearity of the trace functional we have that

$$\begin{aligned} 0 &\preceq \mathrm{Tr}(\boldsymbol{A}(tf(\boldsymbol{X}) + (1-t)f(\boldsymbol{Y}) - f(t\boldsymbol{X} + (1-t)\boldsymbol{Y}))) \\ &= t\,\mathrm{Tr}(\boldsymbol{A}(f(\boldsymbol{X}))) + (1-t)\,\mathrm{Tr}(\boldsymbol{A}f(\boldsymbol{Y})) - \mathrm{Tr}(\boldsymbol{A}(f(t\boldsymbol{X} + (1-t)\boldsymbol{Y}))) \end{aligned}$$

□

Corollary D.1. *If* $\boldsymbol{A} \in \mathbb{S}^d_+$ *then* $\mathrm{Tr}(\boldsymbol{A}\boldsymbol{X}^{-1})$ *and* $\mathrm{Tr}(\boldsymbol{A}\boldsymbol{X})$ *are convex over* $\mathbb{S}^d_{++}$.

Proof. By the Löwner-Heinz theorem [17], x^{-1}, x, and x^2 are operator convex. The result follows by applying Claim D.5. □

Corollary D.2. *[39, Corollary 1.1] If* $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{S}^d_+$ *then* $\mathrm{Tr}(\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}\boldsymbol{X})$ *is convex over* $\mathbb{S}^d_+$.

Proposition D.1. *Let $\{\ell_t : \mathbb{R}_+ \mapsto \mathbb{R}\}_{t\geq 1}$ be of form $\ell_t(\boldsymbol{x}) = \left\|\frac{\boldsymbol{b}_t^2}{\boldsymbol{x}} + \boldsymbol{g}_t^2 \odot \boldsymbol{x}\right\|_1$ for adversarially chosen $\boldsymbol{b}_t, \boldsymbol{g}_t$ satisfying $\|\boldsymbol{b}_t\|_2 \leq D$, $\|\boldsymbol{g}_t\|_2 \leq G$. Then the (ε, ζ, p)-FTL algorithm, which for $\varepsilon, \zeta > 0$ and $p \in (0, \frac{2}{3})$ uses the actions of FTL run on the functions $\tilde{\ell}_t(\boldsymbol{x}) = \left\|\frac{\boldsymbol{b}_t^2 + \varepsilon_t^2 \mathbf{1}_d}{\boldsymbol{x}} + (\boldsymbol{g}_t^2 + \zeta_t^2 \mathbf{1}_d) \odot \boldsymbol{x}\right\|_1$, where $\varepsilon_t^2 = \varepsilon^2(t+1)^{-p}, \zeta_t^2 = \zeta^2(t+1)^{-p}$ for $t \geq 0$ and $\boldsymbol{b}_0 = \boldsymbol{g}_0 = \mathbf{0}_d$, to determine $\boldsymbol{x}_t$, has regret*

$$\mathbf{R}_T \leq C_p \sum_{j=1}^d \min\left\{\left(\frac{\varepsilon^2}{\boldsymbol{x}_j^*} + \zeta^2 \boldsymbol{x}_j^*\right) T^{1-p}, \sqrt{\zeta^2 \boldsymbol{b}_{j,1:T}^2 + \varepsilon^2 \boldsymbol{g}_{j,1:T}^2}\, T^{\frac{1-p}{2}} + 2\varepsilon\zeta T^{1-p}\right\}$$
$$+ C_p\left(\frac{D+\varepsilon}{\zeta^3} G^4 + \frac{G+\zeta}{\varepsilon^3} D^4\right) T^{\frac{3}{2}p} + C_p(D\zeta + G\varepsilon + \varepsilon\zeta)d$$

for any $\boldsymbol{x} > 0$ and some constant C_p depending only on p.

Proof. Define $\tilde{\boldsymbol{b}}_t^2 = \boldsymbol{b}_t^2 + \varepsilon_t^2 \mathbf{1}_d, \tilde{\boldsymbol{g}}_t^2 = \boldsymbol{g}_t^2 + \zeta_t^2 \mathbf{1}_d$ and note that FTL run on the modified functions $\tilde{\ell}_t'(\boldsymbol{x}) = \left\|\frac{\tilde{\boldsymbol{g}}_t^2 \odot \boldsymbol{x}^2}{2} - \tilde{\boldsymbol{b}}_t^2 \odot \log(\boldsymbol{x})\right\|_1$ plays the exact same actions $\boldsymbol{x}_t^2 = \frac{\tilde{\boldsymbol{b}}_{0:t-1}^2}{\tilde{\boldsymbol{g}}_{0:t-1}^2}$ as FTL run $\tilde{\ell}_t$. Since both sequences of loss functions are separable across coordinates, we consider d per-coordinate problems, with loss functions of form $\tilde{\ell}_t(x) = \frac{\tilde{b}_t^2}{x} + \tilde{g}_t^2 x$ and $\tilde{\ell}_t'(x) = \frac{\tilde{g}_t^2 x^2}{2} - \tilde{b}_t^2 \log x$. We have that

$$|\nabla_t| = \left|\tilde{g}_t^2 - \frac{\tilde{b}_t^2}{x_t^2}\right| = \frac{|\tilde{g}_t^2 x_t^2 - \tilde{b}_t^2|}{x_t^2} \qquad |\nabla_t'| = \left|\tilde{g}_t^2 x_t - \frac{\tilde{b}_t^2}{x_t}\right| = \frac{|\tilde{g}_t^2 x_t^2 - \tilde{b}_t^2|}{x_t} \qquad \partial_{xx}\tilde{\ell}_t' = \tilde{g}_t^2 + \frac{\tilde{b}_t^2}{x^2} \geq \tilde{g}_t^2$$

so by Theorem B.1 and substituting the action $x_t^2 = \frac{\tilde{b}_{0:t-1}^2}{\tilde{g}_{0:t-1}^2}$ we have per-coordinate regret

$$\begin{aligned}
\sum_{t=1}^T \tilde{\ell}_t(x_t) - \tilde{\ell}_t(x^*) \leq 2\sum_{t=1}^T \frac{|\nabla_t||\nabla_t'|}{\tilde{g}_{1:t}^2} &= 2\sum_{t=1}^T \frac{|\tilde{g}_t^2 x_t^2 - \tilde{b}_t^2|^2}{x_t^3 \tilde{g}_{1:t}^2} \\
&\leq 2\sum_{t=1}^T \frac{\tilde{g}_t^4 x_t}{\tilde{g}_{1:t}^2} + \frac{\tilde{b}_t^4}{x_t^3 \tilde{g}_{1:t}^2} \\
&= 2\sum_{t=1}^T \frac{\tilde{g}_t^4 \sqrt{\tilde{b}_{0:t-1}^2}}{\tilde{g}_{1:t}^2 \sqrt{\tilde{g}_{0:t-1}^2}} + \frac{\tilde{b}_t^4}{\tilde{g}_{1:t}^2 \left(\frac{\tilde{b}_{0:t-1}^2}{\tilde{g}_{0:t-1}^2}\right)^{\frac{3}{2}}} \\
&\leq 2\sum_{t=1}^T \frac{\tilde{g}_t^4 \sqrt{\tilde{b}_{0:t-1}^2}}{\tilde{g}_{1:t}^2 \sqrt{\tilde{g}_{0:t-1}^2}} + \frac{\tilde{b}_t^4 \sqrt{2\tilde{g}_{1:t}^2}}{(\tilde{b}_{0:t-1}^2)^{\frac{3}{2}}} + \frac{\tilde{b}_t^4 \tilde{g}_0^3 \sqrt{2}}{\tilde{g}_{1:t}^2 (\tilde{b}_{0:t-1}^2)^{\frac{3}{2}}}
\end{aligned}$$

Taking the summation over the coordinates yields

$$\begin{aligned}
&\sum_{t=1}^T \tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_t(\boldsymbol{x}^*) \\
&\quad\leq 4\sum_{t=1}^T \left(\frac{(D+\varepsilon)(\|\boldsymbol{g}_t^2\|_2^2 + \zeta_t^4 d)}{\zeta_{1:t}^2 \sqrt{2\zeta_{0:t-1}^2}} + \frac{(G+\zeta)(\|\boldsymbol{b}_t^2\|_2^2 + \varepsilon_t^4 d)}{(\varepsilon_{0:t-1}^2)^{\frac{3}{2}}} + \frac{(\|\boldsymbol{b}_t^2\|_2^2 + \varepsilon_t^4 d)\zeta^3}{\tilde{\zeta}_{0:t-1}^2 (\tilde{\varepsilon}_{0:t-1}^2)^{\frac{3}{2}}}\right)\sqrt{2t} \\
&\quad\leq 4\sum_{t=1}^T \left(\frac{(D+\varepsilon)(G^4 + \zeta_t^4 d)}{(\underline{c}_p \zeta^2 t^{1-p})^{\frac{3}{2}} \sqrt{2}} + \frac{(G+\zeta)(D^4 + \varepsilon_t^4 d)}{(\underline{c}_p \varepsilon^2 t^{1-p})^{\frac{3}{2}}} + \frac{(D^4 + \varepsilon_t^4 d)\zeta}{\varepsilon^3 (\underline{c}_p t^{1-p})^{\frac{5}{2}}}\right)\sqrt{2t} \\
&\quad\leq 4\sqrt{2}\frac{1 + \frac{1}{\underline{c}_p}}{\underline{c}_p^{\frac{3}{2}}} \sum_{t=1}^T \left(\frac{D+\varepsilon}{\zeta^3} G^4 + \frac{G+\zeta}{\varepsilon^3} D^4\right) t^{\frac{3}{2}p - 1} + \frac{D\zeta + G\varepsilon + 2\varepsilon\zeta}{t^{1+\frac{p}{2}}} d \\
&\quad\leq C_{p,1}\left(\frac{D+\varepsilon}{\zeta^3} G^4 + \frac{G+\zeta}{\varepsilon^3} D^4\right) T^{\frac{3}{2}p} + C_{p,2}(D\zeta + G\varepsilon + 2\varepsilon\zeta)d
\end{aligned}$$

for $C_{p,1} = 4\bar{c}_{1-\frac{3}{2}p}\sqrt{2}\left(1+\frac{1}{\underline{c}_p}\right)/\underline{c}_p^{3/2}$ and $C_{p,2} = 4\sqrt{2}\left(1+\frac{1}{\underline{c}_p}\right)\sum_{t=1}^{\infty}\frac{1}{t^{1+\frac{p}{2}}}/\underline{c}_p^{3/2}$. Thus we have

$$\begin{aligned}
\sum_{t=1}^{T}\ell_t(\boldsymbol{x}_t) &\leq \sum_{t=1}^{T}\tilde{\ell}_t(\boldsymbol{x}_t) \\
&\leq \min_{\boldsymbol{x}^*>0} C_{p,1}\left(\frac{D+\varepsilon}{\zeta^3}G^4 + \frac{G+\zeta}{\varepsilon^3}D^4\right)T^{\frac{3}{2}p} + C_{p,2}(D\zeta + G\varepsilon + 2\varepsilon\zeta)d + \sum_{t=1}^{T}\tilde{\ell}_t(\boldsymbol{x}^*) \\
&= C_{p,1}\left(\frac{D+\varepsilon}{\zeta^3}G^4 + \frac{G+\zeta}{\varepsilon^3}D^4\right)T^{\frac{3}{2}p} + C_{p,2}(D\zeta + G\varepsilon + 2\varepsilon\zeta)d \\
&\quad + \min_{\boldsymbol{x}^*>0}\sum_{t=1}^{T}\left\|\frac{\boldsymbol{b}_t^2 + \varepsilon_t^2\mathbf{1}_d}{\boldsymbol{x}^*} + (\boldsymbol{g}_t^2 + \zeta_t^2\mathbf{1}_d)\odot\boldsymbol{x}^*\right\|_1 \\
&\leq C_{p,1}\left(\frac{D+\varepsilon}{\zeta^3}G^4 + \frac{G+\zeta}{\varepsilon^3}D^4\right)T^{\frac{3}{2}p} + C_{p,2}(D\zeta + G\varepsilon + 2\varepsilon\zeta)d \\
&\quad \min_{\boldsymbol{x}^*>0}\bar{c}_p T^{1-p}\sum_{j=1}^{d}\frac{\varepsilon^2}{\boldsymbol{x}_j^*} + \zeta^2\boldsymbol{x}_j^* + \sum_{t=1}^{T}\ell_t(\boldsymbol{x}^*)
\end{aligned}$$

Separating again per-coordinate we have that

$$\sum_{t=1}^{T}\frac{\tilde{b}_t^2}{x^*} + \tilde{g}_t^2 x^* \leq \bar{c}_p T^{1-p}\frac{\varepsilon^2}{x^*} + \zeta^2 x^* + \sum_{t=1}^{T}\ell_t(x^*)$$

However, substituting $x^* = \sqrt{\frac{\tilde{b}_{1:T}^2}{\tilde{g}_{1:T}}}$ also yields

$$\begin{aligned}
\min_{x^*>0}\sum_{t=1}^{T}\frac{\tilde{b}_t^2}{x^*} + \tilde{g}_t^2 x^* &\leq 2\sqrt{\tilde{b}_{1:T}^2\tilde{g}_{1:T}^2} \\
&\leq 2\sqrt{\bar{c}_p\left(\zeta^2 b_{1:T}^2 + \varepsilon^2 g_{1:T}^2\right)}T^{\frac{1-p}{2}} + 2\bar{c}_p\varepsilon\zeta T^{1-p} + \min_{x^*>0}\sum_{t=1}^{T}\ell_t(x^*)
\end{aligned}$$

completing the proof. $\square$

Theorem D.1. *Let Θ be a bounded convex subset of $\mathbb{R}^d$, let $\mathcal{D} \subset \mathbb{R}^{d\times d}$ be the set of positive definite diagonal matrices, and let each task $t \in [T]$ consist of a sequence of m convex Lipschitz loss functions $\ell_{t,i} : \Theta \mapsto \mathbb{R}$. Suppose for each task t we run the iteration in Equation 5 setting $\boldsymbol{\phi} = \frac{1}{t-1}\boldsymbol{\theta}^*_{1:t-1}$ and setting $\boldsymbol{H} = \mathrm{Diag}(\boldsymbol{\eta}_t)$ via Equation 6 for $\varepsilon = 1, \zeta = \sqrt{m}$, and $p = \frac{2}{5}$. Then we achieve*

$$\bar{\mathbf{R}}_T \le \bar{\mathbf{U}}_T = \min_{\substack{\boldsymbol{\phi}\in\Theta\\ \boldsymbol{H}\in\mathcal{D}}} \tilde{\mathcal{O}}\left(\sum_{j=1}^d \min\left\{\frac{\frac{1}{\boldsymbol{H}_{jj}} + \boldsymbol{H}_{jj}}{T^{\frac{2}{5}}}, \frac{1}{\sqrt[5]{T}}\right\}\right)\sqrt{m} + \frac{1}{T}\sum_{t=1}^T \frac{\|\boldsymbol{\theta}^*_t - \boldsymbol{\phi}\|^2_{\boldsymbol{H}^{-1}}}{2} + \sum_{i=1}^m \|\boldsymbol{\nabla}_{t,i}\|^2_{\boldsymbol{H}}$$

Proof. Define $\boldsymbol{b}_t^2 = \frac{1}{2}(\boldsymbol{\theta}^*_t - \boldsymbol{\phi}_t)^2$ and $\boldsymbol{g}_t^2 = \boldsymbol{\nabla}^2_{1:m}$. Then applying Proposition D.1 yields

$$\begin{aligned}
\bar{\mathbf{U}}_T\, T &= \sum_{t=1}^T \frac{\|\boldsymbol{\theta}^*_t - \boldsymbol{\phi}_t\|^2_{\boldsymbol{\eta}_t^{-1}}}{2} + \sum_{i=1}^m \|\boldsymbol{\nabla}_{t,i}\|^2_{\boldsymbol{\eta}_t} \\
&= \sum_{t=1}^T \left\|\frac{(\boldsymbol{\theta}^*_t - \boldsymbol{\phi}_t)^2}{2\boldsymbol{\eta}_t} + \boldsymbol{\eta}_t \odot \boldsymbol{\nabla}^2_{t,1:m}\right\|_1 \\
&\le \min_{\boldsymbol{\eta}>0} \sum_{t=1}^T \left\|\frac{(\boldsymbol{\theta}^*_t - \boldsymbol{\phi}_t)^2}{2\boldsymbol{\eta}} + \boldsymbol{\eta} \odot \boldsymbol{\nabla}^2_{t,1:m}\right\|_1 \\
&\quad + C_p \sum_{j=1}^d \min\left\{\left(\frac{\varepsilon^2}{\boldsymbol{\eta}_j} + \zeta^2\boldsymbol{\eta}_j\right)T^{1-p}, \sqrt{\zeta^2 \boldsymbol{b}^2_{j,1:T} + \varepsilon^2 \boldsymbol{g}^2_{j,1:T}}\,T^{\frac{1-p}{2}} + 2\varepsilon\zeta T^{1-p}\right\} \\
&\quad + C_p\left(\frac{D+\varepsilon}{\zeta^3}G^4 m^2 + \frac{G\sqrt{m}+\zeta}{\varepsilon^3}D^4\right)T^{\frac{3}{2}p} + C_p(D\zeta + G\sqrt{m}\varepsilon + \varepsilon\zeta)d \\
&\le \min_{\substack{\boldsymbol{\phi}\in\Theta\\ \boldsymbol{\eta}>0}} \sum_{t=1}^T \frac{\|\boldsymbol{\theta}^*_t - \boldsymbol{\phi}\|^2_{\boldsymbol{\eta}^{-1}}}{2} + \sum_{i=1}^{m_t} \|\boldsymbol{\nabla}_{t,i}\|^2_{\boldsymbol{\eta}} + \frac{D_\infty^2}{2}\|\boldsymbol{\eta}^{-1}\|_1(1+\log T) \\
&\quad + C_p \sum_{j=1}^d \min\left\{\left(\frac{\varepsilon^2}{\boldsymbol{\eta}_j} + \zeta^2\boldsymbol{\eta}_j\right)T^{1-p}, \sqrt{\zeta^2 \boldsymbol{b}^2_{j,1:T} + \varepsilon^2 \boldsymbol{g}^2_{j,1:T}}\,T^{\frac{1-p}{2}} + 2\varepsilon\zeta T^{1-p}\right\} \\
&\quad + C_p\left(\frac{D+\varepsilon}{\zeta^3}G^4 m^2 + \frac{G\sqrt{m}+\zeta}{\varepsilon^3}D^4\right)T^{\frac{3}{2}p} + C_p(D\zeta + G\sqrt{m}\varepsilon + \varepsilon\zeta)d
\end{aligned}$$

Substituting $\boldsymbol{\eta} + \frac{\mathbf{1}_d}{\sqrt{mT}}$ for the optimum and the values of ε, ζ, p completes the proof. □

Proposition D.2. *Let $\{\ell_t : \mathbb{R}_+ \mapsto \mathbb{R}\}_{t\geq 1}$ be of form $\ell_t(\boldsymbol{X}) = \operatorname{Tr}(\boldsymbol{X}^{-1}\boldsymbol{B}_t^2) + \operatorname{Tr}(\boldsymbol{X}\boldsymbol{G}_t^2)$ for adversarially chosen $\boldsymbol{B}_t, \boldsymbol{G}_t$ satisfying $\|\boldsymbol{B}_t\|_2 \leq \sigma_B, \|\boldsymbol{G}_t\|_2 \leq \sigma_G\sqrt{m}$ for $m \geq 1$. Then the (ε,ζ)-FTL algorithm, which for $\varepsilon,\zeta > 0$ uses the actions of FTL on the alternate function sequence $\tilde{\ell}_t(\boldsymbol{X}) = \operatorname{Tr}((\boldsymbol{B}^2 + \varepsilon^2\boldsymbol{I}_d)\boldsymbol{X}^{-1}) + \operatorname{Tr}((\boldsymbol{G}^2 + \zeta^2\boldsymbol{I}_d)\boldsymbol{X})$, achieves regret*

$$\mathbf{R}_T \leq \frac{C_\sigma m^2}{\varepsilon^4\zeta^3}(1+\log T) + ((1+\sigma_G^2)\varepsilon\sqrt{m} + (1+\sigma_B^2)\zeta)T$$

for constant C_σ depending only on σ_B, σ_G.

Proof. Define $\tilde{\boldsymbol{B}}_t^2 = \boldsymbol{B}_t^2 + \varepsilon^2\boldsymbol{I}_d, \tilde{\boldsymbol{G}}_t^2 = \boldsymbol{G}_t^2 + \zeta^2\boldsymbol{I}_d$ and note that FTL run on modified functions $\tilde{\ell}_t'(\boldsymbol{X}) = \frac{1}{2}\operatorname{Tr}(\tilde{\boldsymbol{B}}_t^{-2}\boldsymbol{X}\tilde{\boldsymbol{G}}_t^2\boldsymbol{X}) - \log\det\boldsymbol{X}$ has the same solution $\tilde{\boldsymbol{B}}_{1:T}^2 = \boldsymbol{X}\tilde{\boldsymbol{G}}_{1:T}^2\boldsymbol{X}$.

$$\|\nabla_{\boldsymbol{X}}\tilde{\ell}_t(\boldsymbol{X})\|_2 = \|\tilde{\boldsymbol{G}}_t^2 - \boldsymbol{X}^{-1}\tilde{\boldsymbol{B}}_t^2\boldsymbol{X}^{-1}\|_2 \leq \|\tilde{\boldsymbol{G}}_t\|_2^2 + \|\boldsymbol{X}^{-1}\|_2^2\|\tilde{\boldsymbol{B}}_t\|_2^2 \leq \frac{\sigma_B^2}{\varepsilon^2} + m\sigma_G^2 + \zeta^2$$

$$\begin{aligned}\|\nabla_{\boldsymbol{X}}\tilde{\ell}_t'(\boldsymbol{X})\|_2 = \|\tilde{\boldsymbol{G}}_t^2\boldsymbol{X}\tilde{\boldsymbol{B}}_t^{-2} - \boldsymbol{X}^{-1}\|_2 &\leq \|\tilde{\boldsymbol{G}}_t\|_2^2\|\boldsymbol{X}\|_2\|\tilde{\boldsymbol{B}}_t^{-1}\|_2^2 + \|\boldsymbol{X}^{-1}\|_2 \\ &\leq \frac{(m\sigma_G^2+\zeta^2)\sqrt{\sigma_B^2+\varepsilon^2}}{\varepsilon^2\zeta} + \frac{\sqrt{m\sigma_G^2+\zeta^2}}{\zeta}\end{aligned}$$

Since by Claim D.4 $-\log\det|\boldsymbol{X}|$ is $\frac{\zeta^2}{\sigma_B^2+\varepsilon^2}$-strongly-convex we have by Theorem B.1 that

$$\sum_{t=1}^T \tilde{\ell}_t(\boldsymbol{X}_t) - \tilde{\ell}_t(\boldsymbol{X}^*) \leq \frac{C_\sigma m^2}{\varepsilon^4\zeta^3}(1+\log T)$$

for some C_σ depending on σ_B^2, σ_G^2. Therefore

$$\begin{aligned}\sum_{t=1}^T \ell_t(\boldsymbol{X}) &\leq \sum_{t=1}^T \tilde{\ell}_t(\boldsymbol{X}) \\ &\leq \frac{C_\sigma m^2}{\varepsilon^4\zeta^3}(1+\log T) + \min_{\boldsymbol{X}\succ 0}\sum_{t=1}^T \tilde{\ell}_t(\boldsymbol{X}) \\ &\leq \frac{C_\sigma m^2}{\varepsilon^4\zeta^3}(1+\log T) + \min_{\boldsymbol{X}\succ 0}\varepsilon^2 T\operatorname{Tr}(\boldsymbol{X}^{-1}) + \zeta^2 T\operatorname{Tr}(\boldsymbol{X}) + \sum_{t=1}^T \ell_t(\boldsymbol{X}) \\ &\leq \frac{C_\sigma m^2}{\varepsilon^4\zeta^3}(1+\log T) + (1+\sigma_G^2)\varepsilon T\sqrt{m} + \min_{\boldsymbol{X}\succ 0}\zeta^2 T\operatorname{Tr}(\boldsymbol{X}) + \sum_{t=1}^T \ell_t(\boldsymbol{X}) \\ &\leq \frac{C_\sigma m^2}{\varepsilon^4\zeta^3}(1+\log T) + ((1+\sigma_G^2)\varepsilon\sqrt{m} + (1+\sigma_B^2)\zeta)T + \min_{\boldsymbol{X}\succ 0}\sum_{t=1}^T \ell_t(\boldsymbol{X})\end{aligned}$$

□

Theorem D.2. *Let Θ be a bounded convex subset of $\mathbb{R}^d$ and let each task $t \in [T]$ consist of a sequence of m convex Lipschitz loss functions $\ell_{t,i} : \Theta \mapsto \mathbb{R}$. Suppose for each task t we run the iteration in Equation 5 with $\boldsymbol{\phi} = \frac{1}{t-1}\boldsymbol{\theta}^*_{1:t-1}$ and $\boldsymbol{H}$ the unique positive definite solution of $\boldsymbol{B}_t^2 = \boldsymbol{H}\boldsymbol{G}_t^2\boldsymbol{H}$ for*

$$\boldsymbol{B}_t^2 = t\varepsilon^2 \boldsymbol{I}_d + \sum_{s<t}(\boldsymbol{\theta}^*_s - \boldsymbol{\phi}_s)(\boldsymbol{\theta}^*_s - \boldsymbol{\phi}_s)^T \qquad \text{and} \qquad \boldsymbol{G}_t^2 = t\varepsilon^2 \boldsymbol{I}_d + \sum_{s<t}\sum_{i=1}^m \boldsymbol{\nabla}_{s,i}\boldsymbol{\nabla}_{s,i}^T$$

for $\varepsilon = 1/\sqrt[8]{T}$ and $\zeta = \sqrt{m}/\sqrt[8]{T}$. Then we achieve

$$\bar{\mathbf{R}}_T \leq \bar{\mathbf{U}}_T = \tilde{\mathcal{O}}\left(\frac{1}{\sqrt[8]{T}}\right)\sqrt{m} + \min_{\substack{\boldsymbol{\phi}\in\Theta \\ \boldsymbol{H}\succ 0}} \frac{2\lambda_1^2(\boldsymbol{H})}{\lambda_d(\boldsymbol{H})}\frac{1+\log T}{T} + \sum_{t=1}^T \frac{\|\boldsymbol{\theta}^*_t - \boldsymbol{\phi}^*\|^2_{\boldsymbol{H}^{-1}}}{2} + \sum_{i=1}^m \|\boldsymbol{\nabla}_{t,i}\|^2_{\boldsymbol{H}}$$

Proof. Let D and G be the diameter of Θ and Lipschitz bound on the losses, respectively. Then applying Proposition D.2 yields

$$\begin{aligned}
\bar{\mathbf{U}}_T T &= \sum_{t=1}^T \frac{\|\boldsymbol{\theta}^*_t - \boldsymbol{\phi}_t\|^2_{\boldsymbol{H}_t^{-1}}}{2} + \sum_{i=1}^m \|\boldsymbol{\nabla}_{t,i}\|^2_{\boldsymbol{H}_t} \\
&= \sum_{t=1}^T \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{H}_t^{-1}(\boldsymbol{\theta}^*_t - \boldsymbol{\phi}_t)(\boldsymbol{\theta}^*_t - \boldsymbol{\phi}_t)^T\right) + \operatorname{Tr}\left(\boldsymbol{H}_t \sum_{i=1}^m \boldsymbol{\nabla}_{t,i}\boldsymbol{\nabla}_{t,i}^T\right) \\
&\leq \min_{\boldsymbol{H}\succ 0} \sum_{t=1}^T \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{H}^{-1}(\boldsymbol{\theta}^*_t - \boldsymbol{\phi}_t)(\boldsymbol{\theta}^*_t - \boldsymbol{\phi}_t)^T\right) + \operatorname{Tr}\left(\boldsymbol{H} \sum_{i=1}^m \boldsymbol{\nabla}_{t,i}\boldsymbol{\nabla}_{t,i}^T\right) \\
&\quad + \frac{C_\sigma m^2}{\varepsilon^4\zeta^3}(1+\log T) + ((1+G^2)\varepsilon\sqrt{m} + (1+D^2)\zeta)T \\
&= \min_{\boldsymbol{H}\succ 0} \sum_{t=1}^T \frac{\|\boldsymbol{\theta}^*_t - \boldsymbol{\phi}_t\|^2_{\boldsymbol{H}^{-1}}}{2} + \operatorname{Tr}\left(\boldsymbol{H} \sum_{i=1}^m \boldsymbol{\nabla}_{t,i}\boldsymbol{\nabla}_{t,i}^T\right) \\
&\quad + \frac{C_\sigma m^2}{\varepsilon^4\zeta^3}(1+\log T) + ((1+G^2)\varepsilon\sqrt{m} + (1+D^2)\zeta)T \\
&\leq \min_{\substack{\boldsymbol{\phi}\in\Theta \\ \boldsymbol{H}\succ 0}} \frac{2\lambda_1^2(\boldsymbol{H})}{\lambda_d(\boldsymbol{H})}\sum_{t=1}^T \frac{1}{t} + \sum_{t=1}^T \frac{\|\boldsymbol{\theta}^*_t - \boldsymbol{\phi}^*\|^2_{\boldsymbol{H}^{-1}}}{2} + \sum_{i=1}^m \|\boldsymbol{\nabla}_{t,i}\|^2_{\boldsymbol{H}} \\
&\quad + \frac{C_\sigma m^2}{\varepsilon^4\zeta^3}(1+\log T) + ((1+G^2)\varepsilon\sqrt{m} + (1+D^2)\zeta)T \\
&= \min_{\substack{\boldsymbol{\phi}\in\Theta \\ \boldsymbol{H}\succ 0}} \frac{2\lambda_1^2(\boldsymbol{H})}{\lambda_d(\boldsymbol{H})}\sum_{t=1}^T \frac{1}{t} + \sum_{t=1}^T \frac{\|\boldsymbol{\theta}^*_t - \boldsymbol{\phi}^*\|^2_{\boldsymbol{H}^{-1}}}{2} + \sum_{i=1}^m \|\boldsymbol{\nabla}_{t,i}\|^2_{\boldsymbol{H}} \\
&\quad + \frac{C_\sigma m^2}{\varepsilon^4\zeta^3}(1+\log T) + ((1+G^2)\varepsilon\sqrt{m} + (1+D^2)\zeta)T
\end{aligned}$$

$\square$

E Online-to-Batch Conversion for Task-Averaged Regret

Theorem E.1. *Let $\mathcal{Q}$ be a distribution over distributions $\mathcal{P}$ over convex loss functions $\ell : \Theta \mapsto [0, 1]$. A sequence of sequences of loss functions $\{\ell_{t,i}\}_{t\in[T],i\in[m]}$ is generated by drawing m loss functions i.i.d. from each in a sequence of distributions $\{\mathcal{P}_t\}_{t\in[T]}$ themselves drawn i.i.d. from $\mathcal{Q}$. If such a sequence is given to an meta-learning algorithm with task-averaged regret bound $\bar{\mathbf{R}}_T$ that has states $\{s_t\}_{t\in[T]}$ at the beginning of each task t then we have w.p. $1-\delta$ for any $\theta^* \in \Theta$ that*

$$\underset{t\sim\mathcal{U}[T]}{\mathbb{E}}\ \underset{\mathcal{P}\sim\mathcal{Q}}{\mathbb{E}}\ \underset{\mathcal{P}^m}{\mathbb{E}}\ \underset{\ell\sim\mathcal{P}}{\mathbb{E}}\ \ell(\bar{\theta}) \leq \underset{\mathcal{P}\sim\mathcal{Q}}{\mathbb{E}}\ \underset{\ell\sim\mathcal{P}}{\mathbb{E}}\ \ell(\theta^*) + \frac{\bar{\mathbf{R}}_T}{m} + \sqrt{\frac{8}{T}\log\frac{1}{\delta}}$$

where $\bar{\theta} = \frac{1}{m}\theta_{1:m}$ is generated by randomly sampling $t \in \mathcal{U}[T]$, running the online algorithm with state s_t, and averaging the actions $\{\theta_i\}_{i\in[m]}$. If on each task the meta-learning algorithm runs an online algorithm with regret upper bound $\mathbf{U}_m(s_t)$ a convex, nonnegative, and $B\sqrt{m}$-bounded function of the state $s_t \in \mathcal{X}$, where $\mathcal{X}$ is a convex Euclidean subset, and the total regret upper bound is $\bar{\mathbf{U}}_T$, then we also have the bound

$$\underset{\mathcal{P}\sim\mathcal{Q}}{\mathbb{E}}\ \underset{\mathcal{P}^m}{\mathbb{E}}\ \underset{\ell\sim\mathcal{P}}{\mathbb{E}}\ \ell(\bar{\theta}) \leq \underset{\mathcal{P}\sim\mathcal{Q}}{\mathbb{E}}\ \underset{\ell\sim\mathcal{P}}{\mathbb{E}}\ \ell(\theta^*) + \frac{\bar{\mathbf{U}}_T}{m} + B\sqrt{\frac{8}{mT}\log\frac{1}{\delta}}$$

where $\bar{\theta} = \frac{1}{m}\theta_{1:m}$ is generated by running the online algorithm with state $\bar{s} = \frac{1}{T}s_{1:T}$ and averaging the actions $\{\theta_i\}_{i\in[m]}$.

Proof. For the second inequality, applying Proposition A.1, Jensen's inequality, and Proposition A.2 yields

$$\begin{aligned}
\underset{\mathcal{P}\sim\mathcal{Q}}{\mathbb{E}}\ \underset{\mathcal{P}^m}{\mathbb{E}}\ \underset{\ell\sim\mathcal{P}}{\mathbb{E}}\ \ell(\bar{\theta}) &\leq \underset{\mathcal{P}\sim\mathcal{Q}}{\mathbb{E}}\left(\underset{\ell\sim\mathcal{P}}{\mathbb{E}}\ \ell(\theta^*) + \frac{\mathbf{U}_m(\bar{s})}{m}\right)\\
&\leq \underset{\mathcal{P}\sim\mathcal{Q}}{\mathbb{E}}\ \underset{\ell\sim\mathcal{P}}{\mathbb{E}}\ \ell(\theta^*) + \frac{1}{T}\sum_{t=1}^T \underset{\mathcal{P}\sim\mathcal{Q}}{\mathbb{E}}\left(\frac{\mathbf{U}_m(s_t)}{m}\right)\\
&= \underset{\mathcal{P}\sim\mathcal{Q}}{\mathbb{E}}\ \underset{\ell\sim\mathcal{P}}{\mathbb{E}}\ \ell(\theta^*) + \frac{2B}{T\sqrt{m}}\sum_{t=1}^T \underset{\mathcal{P}\sim\mathcal{Q}}{\mathbb{E}}\left(\frac{\mathbf{U}_m(s_t)}{2B\sqrt{m}} + \frac{\sqrt{m}}{2B}\right) - 1\\
&\leq \underset{\mathcal{P}\sim\mathcal{Q}}{\mathbb{E}}\ \underset{\ell\sim\mathcal{P}}{\mathbb{E}}\ \ell(\theta^*) + \frac{\bar{\mathbf{U}}_T}{m} + B\sqrt{\frac{8}{mT}\log\frac{1}{\delta}}
\end{aligned}$$

The first inequality follows similarly except using $\mathbf{R}_m$ instead of $\mathbf{U}_m$, linearity of expectation instead of Jensen's inequality, 1 instead of B, and $\bar{\mathbf{R}}_T$ instead of $\bar{\mathbf{U}}_T$. □

Note that since regret-upper-bounds are nonnegative one can easily replace 8 by 2 in the second inequality by simply multiplying and dividing by $B\sqrt{m}$ in the third line of the above proof.

Claim E.1. *In the setup of Theorem E.1, let $\theta_t^* \in \arg\min_{\theta\in\Theta}\sum_{i=1}^m \ell_{t,i}(\theta)$ and define the quantities $V_\mathcal{Q}^2 = \arg\min_{\phi\in\Theta}\mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}_{\mathcal{P}^m}\|\theta^*-\phi\|_2^2$ and D the ℓ_2-radius of Θ. Then w.p. $1-\delta$ we have*

$$V^2 = \min_{\phi\in\Theta}\frac{1}{T}\sum_{t=1}^T\|\theta_t^*-\phi\|_2^2 \leq \mathcal{O}\left(V_\mathcal{Q}^2 + \frac{D^2}{T}\log\frac{1}{\delta}\right)$$

Proof. Define $\hat{\phi} = \arg\min_{\phi\in\Theta}\sum_{t=1}^T\|\theta_t^*-\phi\|_2^2$ and $\phi^* = \arg\min_{\phi\in\Theta}\mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}_{\mathcal{P}^m}\|\theta^*-\phi\|_2^2$. Then by a multiplicative Chernoff's inequality w.p. at least $1-\delta$ we have

$$\begin{aligned}
TV^2 = \sum_{t=1}^T\|\theta_t^*-\hat{\phi}\|_2^2 \leq \sum_{t=1}^T\|\theta_t^*-\phi^*\|_2^2 &\leq \left(1+\max\left\{1, \frac{3D^2}{V_\mathcal{Q}^2 T}\log\frac{1}{\delta}\right\}\right)T\ \underset{\mathcal{P}\sim\mathcal{Q}}{\mathbb{E}}\ \underset{\mathcal{P}^m}{\mathbb{E}}\ \|\theta^*-\phi^*\|_2^2\\
&\leq 2TV_\mathcal{Q}^2 + 3D^2\log\frac{1}{\delta}
\end{aligned}$$

□

Corollary E.1. *Under the assumptions of Theorems 3.2 and 5.1, if the loss functions are Lipschitz and we use Algorithm 1 with η_t also learned, using ε-EWOO as in Theorem 3.2 for $\varepsilon = 1/\sqrt[4]{mT} + 1/\sqrt{m}$, and set the initialization using $\phi_{t+1} = \frac{1}{t}\sum_{s\le t}\theta_s^*$, then w.p. $1-\delta$ we have*

$$\mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\mathop{\mathbb{E}}_{\mathcal{P}^m}\ell_{\mathcal{P}}(\bar{\theta}) \le \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\ell_{\mathcal{P}}(\theta^*) + \tilde{\mathcal{O}}\left(\frac{V_{\mathcal{Q}}}{\sqrt{m}} + \min\left\{\frac{\frac{1}{\sqrt{T}}+\frac{1}{\sqrt{m}}}{V_{\mathcal{Q}}m}, \frac{1}{\sqrt[4]{m^3T}} + \frac{1}{m}\right\} + \sqrt{\frac{1}{T}\log\frac{1}{\delta}}\right)$$

where $V_{\mathcal{Q}}^2 = \min_{\phi\in\Theta}\mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}_{\mathcal{P}^m}\|\theta^*-\phi\|_2^2$.

Proof. Substitute Corollary C.3 into Theorem E.1 using the fact the the regret-upper-bounds are $\mathcal{O}(\frac{\sqrt{m}}{\varepsilon})$-bounded. Conclude by applying Claim E.1. □

Theorem E.2. *Let $\mathcal{Q}$ be a distribution over distributions $\mathcal{P}$ over convex losses $\ell : \Theta \mapsto [0,1]$ such that the functions $\ell(\theta) - \ell(\theta^*)$ are ρ-self-bounded for some $\rho > 0$ and $\theta^* \in \arg\min_{\theta\in\Theta}\mathbb{E}_{\ell\sim\mathcal{P}}(\theta)$. A sequence of sequences of loss functions $\{\ell_{t,i}\}_{t\in[T],i\in[m]}$ is generated by drawing m loss functions i.i.d. from each in a sequence of distributions $\{\mathcal{P}_t\}_{t\in[T]}$ themselves drawn i.i.d. from $\mathcal{Q}$. If such a sequence is given to an meta-learning algorithm with task-averaged regret bound $\bar{\mathbf{R}}_T$ that has states $\{s_t\}_{t\in[T]}$ at the beginning of each task t then we have w.p. $1-\delta$ for any $\theta^*\in\Theta$ that*

$$\mathop{\mathbb{E}}_{t\sim\mathcal{U}[T]}\mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\mathop{\mathbb{E}}_{\ell\sim\mathcal{P}}\ell(\bar{\theta}) \le \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\mathop{\mathbb{E}}_{\ell\sim\mathcal{P}}\ell(\theta^*) + \frac{\bar{\mathbf{R}}_T}{m} + \sqrt{\frac{2\rho}{m}\left(\frac{\bar{\mathbf{R}}_T}{m} + \sqrt{\frac{8}{T}\log\frac{2}{\delta}}\right)\log\frac{2}{\delta}} + \sqrt{\frac{8}{T}\log\frac{2}{\delta}} + \frac{3\rho+2}{m}\log\frac{2}{\delta}$$

where $\bar{\theta} = \frac{1}{m}\theta_{1:m}$ is generated by randomly sampling $t\in\mathcal{U}[T]$, running the online algorithm with state s_t, and averaging the actions $\{\theta_i\}_{i\in[m]}$. If on each task the meta-learning algorithm runs an online algorithm with regret upper bound $\mathbf{U}_m(s_t)$ a convex, nonnegative, and $B\sqrt{m}$-bounded function of the state $s_t\in\mathcal{X}$, where $\mathcal{X}$ is a convex Euclidean subset, and the total regret upper bound is $\bar{\mathbf{U}}_T$, then we also have the bound

$$\mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\mathop{\mathbb{E}}_{\ell\sim\mathcal{P}}\ell(\bar{\theta}) \le \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\mathop{\mathbb{E}}_{\ell\sim\mathcal{P}}\ell(\theta^*) + \frac{\bar{\mathbf{U}}_T}{m} + \sqrt{\frac{2\rho}{m}\left(\frac{\bar{\mathbf{U}}_T}{m} + B\sqrt{\frac{8}{mT}\log\frac{2}{\delta}}\right)\log\frac{2}{\delta}} + B\sqrt{\frac{8}{mT}\log\frac{2}{\delta}} + \frac{3\rho+2}{m}\log\frac{2}{\delta}$$

where $\bar{\theta} = \frac{1}{m}\theta_{1:m}$ is generated by running the online algorithm with state $\bar{s} = \frac{1}{T}s_{1:T}$ and averaging the actions $\{\theta_i\}_{i\in[m]}$.

Proof. By Corollary A.2 and Jensen's inequality we have w.p. $1-\frac{\delta}{2}$ that

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\mathop{\mathbb{E}}_{\ell\sim\mathcal{P}}\ell(\bar{\theta}) &\le \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\left(\mathop{\mathbb{E}}_{\ell\sim\mathcal{P}}\ell(\theta^*) + \frac{\mathbf{U}_m(\bar{s})}{m} + \frac{1}{m}\sqrt{2\rho\,\mathbf{U}_m(\bar{s})\log\frac{1}{\delta}} + \frac{3\rho+2}{m}\log\frac{1}{\delta}\right)\\ &\le \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\mathop{\mathbb{E}}_{\ell\sim\mathcal{P}}\ell(\theta^*) + \frac{1}{T}\sum_{t=1}^T\mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\left(\frac{\mathbf{U}_m(s_t)}{m}\right)\\ &\quad + \sqrt{\frac{2\rho}{mT}\sum_{t=1}^T\mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\left(\frac{\mathbf{U}_m(s_t)}{m}\right)\log\frac{2}{\delta}} + \frac{3\rho+2}{m}\log\frac{2}{\delta}\end{aligned}$$

As in the proof of Theorem E.1, by Proposition A.2 we further have w.p. $1-\frac{\delta}{2}$ that

$$\frac{1}{T}\sum_{t=1}^T\mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\left(\frac{\mathbf{U}_m(s_t)}{m}\right) \le \frac{\bar{\mathbf{U}}_T}{m} + B\sqrt{\frac{8}{mT}\log\frac{2}{\delta}}$$

Substituting the second inequality into the first yields the second bound. The first bound follows similarly except using $\mathbf{R}_m$ instead of $\mathbf{U}_m$, linearity of expectation instead of Jensen's inequality, 1 instead of B, and $\bar{\mathbf{R}}_T$ instead of $\bar{\mathbf{U}}_T$. □

Theorem E.3. *Let $\mathcal{Q}$ be a distribution over distributions $\mathcal{P}$ over convex loss functions $\ell : \Theta \mapsto [0, 1]$. A sequence of sequences of loss functions $\{\ell_{t,i}\}_{t\in[T],i\in[m]}$ is generated by drawing m loss functions i.i.d. from each in a sequence of distributions $\{\mathcal{P}_t\}_{t\in[T]}$ themselves drawn i.i.d. from $\mathcal{Q}$. If such a sequence is given to an meta-learning algorithm that on each task runs an online algorithm with regret upper bound $\mathbf{U}_m(s_t)$ a nonnegative, $B\sqrt{m}$-bounded, G-Lipschitz w.r.t. $\|\cdot\|$, and α-strongly-convex w.r.t. $\|\cdot\|$ function of the state $s_t \in \mathcal{X}$ at the beginning of each task t, where $\mathcal{X}$ is a convex Euclidean subset, and the total regret upper bound is $\bar{\mathbf{U}}_T$, then we have w.p. $1-\delta$ for any $\theta^* \in \Theta$ that*

$$\mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} \mathop{\mathbb{E}}_{\ell\sim\mathcal{P}} \ell(\bar{\theta}) \leq \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\ell\sim\mathcal{P}} \ell(\theta^*) + \mathcal{L}_T$$

for

$$\mathcal{L}_T = \frac{\mathbf{U}^* + \bar{\mathbf{U}}_T}{m} + \frac{4G}{T}\sqrt{\frac{\bar{\mathbf{U}}_T}{\alpha m}\log\frac{8\log T}{\delta}} + \frac{\max\{16G^2, 6\alpha B\sqrt{m}\}}{\alpha m T}\log\frac{8\log T}{\delta}$$

where $\mathbf{U}^* = \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbf{U}_m(s^*)$ for any valid s^* and $\bar{\theta} = \frac{1}{m}\theta_{1:m}$ is generated by running the online algorithm with state $\bar{s} = \frac{1}{T}s_{1:T}$ and averaging the actions $\{\theta_i\}_{i\in[m]}$. If we further assume that the functions $\ell(\theta) - \ell(\theta^*)$ are ρ-self-bounded for some $\rho > 0$ and $\theta^* \in \arg\min_{\theta\in\Theta}\mathbb{E}_{\ell\sim\mathcal{P}}(\theta)$ for all $\mathcal{P}$ in the support of $\mathcal{Q}$ then we also have the bound

$$\mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\ell\sim\mathcal{P}} \ell(\bar{\theta}) \leq \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\ell\sim\mathcal{P}} \ell(\theta^*) + \mathcal{L}_T + \sqrt{\frac{2\rho\mathcal{L}_T}{m}\log\frac{2}{\delta}} + \frac{3\rho+2}{m}\log\frac{2}{\delta}$$

Proof. Applying Proposition A.1 and Theorem A.4 we have w.p. $1-\frac{\delta}{2}$ that

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} \mathop{\mathbb{E}}_{\ell\sim\mathcal{P}} \ell(\bar{\theta}) &\leq \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\left(\mathop{\mathbb{E}}_{\ell\sim\mathcal{P}}\ell(\theta^*) + \frac{\mathbf{U}_m(\bar{s})}{m}\right) \\
&\leq \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\ell\sim\mathcal{P}} \ell(\theta^*) + \frac{1}{m}\mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}}\mathbf{U}_m(s^*) + \frac{\bar{\mathbf{U}}_T}{m} \\
&\qquad + \frac{4G}{T}\sqrt{\frac{\bar{\mathbf{U}}_T}{\alpha m}\log\frac{8\log T}{\delta}} + \frac{\max\{16G^2, 6\alpha B\sqrt{m}\}}{\alpha m T}\log\frac{8\log T}{\delta} \\
&\leq \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\ell\sim\mathcal{P}} \ell(\theta^*) + \mathcal{L}_T
\end{aligned}$$

This yields the first bound since. The second bound follows similarly except for the application of Corollary A.2 in the second step w.p. $1-\frac{\delta}{2}$. □

Corollary E.2. *Under the assumptions of Theorem 5.1 and boundedness of Θ, if the loss functions are G-Lipschitz and we use Algorithm 1 running OGD with fixed $\eta = \frac{V_{\mathcal{Q}}+1/\sqrt{T}}{G\sqrt{m}}$, where we have $V_{\mathcal{Q}}^2 = \min_{\phi\in\Theta}\mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}_{\mathcal{P}^m}\|\theta^* - \phi\|_2^2$, and set the initialization using $\phi_{t+1} = \frac{1}{t}\theta^*_{1:t}$, then w.p. $1-\delta$ we have*

$$\mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} \ell_{\mathcal{P}}(\bar{\theta}) \leq \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \ell_{\mathcal{P}}(\theta^*) + \tilde{\mathcal{O}}\left(\frac{V_{\mathcal{Q}}}{\sqrt{m}} + \left(\frac{1}{T} + \frac{1}{\sqrt{mT}}\right)\max\left\{\log\frac{1}{\delta}, \sqrt{\log\frac{1}{\delta}}\right\}\right)$$

Proof. Apply Theorem C.1 with $V_\Phi = V_{\mathcal{Q}} + 1/\sqrt{T}$, $\mathbf{U}^{\text{sim}} = 0$ (because the learning rate is fixed), and $\mathbf{U}^{\text{init}} = \tilde{\mathcal{O}}\left(\hat{V}\sqrt{m} + 1/\sqrt{T}\right)$ (for $\hat{V}^2 = \min_{\phi\in\Theta}\frac{1}{T}\sum_{t=1}^T\|\theta_t^* - \phi\|_2^2$). Substitute the result into Theorem E.3 using the fact that $\mathbf{U}_m$ is $\mathcal{O}\left(\left(\frac{1}{\varepsilon}+\varepsilon\right)\sqrt{m}\right)$-bounded, $\mathcal{O}\left(\frac{\sqrt{m}}{\varepsilon}\right)$-Lipschitz, and $\Omega\left(\frac{\sqrt{m}}{\varepsilon}\right)$-strongly-convex. Conclude by applying Claim E.1 to bound $\hat{V}$. □

F Adapting to Task-Similarity under Parameter Growth

In this appendix we cast the problem of adaptively learning the task-similarity in the framework of Khodak et al. [34]. We do this specifically to show that our basic results extend to approximate meta-updates under quadratic growth. We first provide a generalized version of their Ephemeral method in Algorithm 3. We then state the relevant approximation assumptions and proceed to prove guarantees on the average regret-upper-bound for the case of a fixed task-similarity in Theorem F.1 and for adaptively learning it in Theorem F.2. Then the quadratic-growth results of Khodak et al. [34], specifically Propositions B.1, B.2, and B.3, can be applied directly to show average regret-upper-bound guarantees of the same order as those in the main paper but with additional $o_m(1)$ terms inside the parentheses. Note that our results, especially in the batch-within-online setting, will in general be stronger because we do not incur the $\Delta_{\max}$-error term that is needed to account for the doubling trick in Khodak et al. [34].

Algorithm 3: Follow-the-Meta-Regularized-Leader (Ephemeral) meta-algorithm for meta-learning [34]. For the *Optimal Action* variant we assume $\arg\min_{\theta\in\Theta} L(\theta)$ returns θ minimizing $\mathcal{B}_R(\theta|\phi_R)$ over the set of all minimizers of L over Θ, where ϕ_R is some appropriate element of Φ such as the origin in Euclidean space or the uniform distribution over the simplex.

Data:

- action space $\Theta \subset \mathbb{R}^d$ with norm $\|\cdot\|$
- function $R : \Theta \mapsto \mathbb{R}$ that is 1-strongly-convex w.r.t. $\|\cdot\|$ and its corresponding Bregman divergence $\mathcal{B}_R$
- class of within-task algorithms $\{\text{TASK}_{\eta,\phi} : \eta > 0, \phi \in \Theta\}$
- meta-update algorithms INIT and SIM
- sequence of loss functions $\{\ell_{t,i} : \Theta \mapsto \mathbb{R}\}_{t\in[T],i\in[m_t]}$ where $\ell_{t,i}$ is $G_{t,i}$-Lipschitz w.r.t. $\|\cdot\|$

for $t \in [T]$ **do**

`// set learning rate and initialization using meta-update algorithms`

$D_t = \text{SIM}(\{\ell_{s,i}\}_{s<t,i\in[m_s]})$

$G_t \leftarrow \sqrt{\frac{1}{m_t}\sum_{i=1}^{m_t} G_{t,i}^2}$

$\eta_t \leftarrow \frac{D_t}{G_t\sqrt{m_t}}$

$\phi_t = \text{INIT}(\{\ell_{s,i}\}_{s<t,i\in[m_s]})$

`// run within-task algorithm`

for $i \in [m_t]$ **do**

$\theta_{t,i} \leftarrow \text{TASK}_{\eta_t,\phi_t}(\ell_{t,1},\dots,\ell_{t,i-1})$

suffer loss $\ell_{t,i}(\theta_{t,i})$

// compute meta-update vector θ_t according to Ephemeral variant

case Optimal Action **do**

$\theta_t \leftarrow \arg\min_{\theta\in\Theta}\sum_{i=1}^{m_t}\ell_{t,i}(\theta)$

case Last Iterate **do**

$\theta_t \leftarrow \text{TASK}_{\eta_t,\phi_t}(\ell_{t,1},\dots,\ell_{t,m_t})$

case Average Iterate **do**

$\theta_t \leftarrow \frac{1}{m_t}\sum_{i=1}^{m_t}\theta_{t,i}$

Assumption F.1. *Assume the data given to Algorithm 3 and define the following quantities:*

- *convenience coefficients* $\sigma_t = G_t\sqrt{m_t}$
- *sequence of update parameters* $\{\hat\theta_t \in \Theta\}_{t\in[T]}$ *with average update* $\hat\phi = \frac{1}{\sigma_{1:T}}\sum_{t=1}^T \sigma_t\hat\theta$
- *a sequence of reference parameters* $\{\theta'_t \in \Theta\}_{t\in[T]}$ *with average reference parameter* $\phi' = \frac{1}{\sigma_{1:T}}\sum_{t=1}^T \sigma_t\theta'_t$
- *a sequence* $\{\theta^*_t \in \Theta\}_{t\in[T]}$ *of optimal parameters in hindsight*
- *we will say we are in the "Exact" case if* $\hat\theta_t = \theta'_t = \theta^*_t\ \forall\ t$ *and the "Approx" case otherwise*
- $\kappa \geq 1, \Delta^*_t \geq 0$ *s.t.* $\sum_{t=1}^T \alpha_t\mathcal{B}_R(\theta^*_t||\phi_t) \leq \Delta^*_{1:T} + \kappa\sum_{t=1}^T \alpha_t\mathcal{B}_R(\hat\theta_t||\phi_t)$ *for some* $\alpha_t \geq 0$
- $\nu \geq 1, \Delta' \geq 0$ *s.t.* $\sum_{t=1}^T \sigma_t\mathcal{B}_R(\hat\theta_t||\hat\phi) \leq \Delta' + \nu\sum_{t=1}^T \sigma_t\mathcal{B}_R(\theta'_t||\phi')$
- *average deviation* $V^2 = \frac{1}{\sigma_{1:T}}\sum_{t=1}^T \sigma_t\mathcal{B}_R(\theta'_t||\phi')$ *of the reference parameters*
- *action diameter* $D^2 = \max\{D^{*2}, \max_{\theta\in\Theta}\mathcal{B}_R(\theta||\phi_1)\}$ *in the Exact case or* $\max_{\theta,\phi\in\Theta}\mathcal{B}_R(\theta||\phi)$ *in the Approx case*
- *constant* C' *s.t.* $\|\theta\| \leq C'\|\theta\|_2\ \forall\ \theta\in\Theta$ *and* ℓ_2*-diameter* $D' = \max_{\theta,\phi}\|\theta-\phi\|_2$ *of* Θ
- *effective action space* $\hat\Theta = \mathrm{Conv}(\{\hat\theta_t\}_{t\in[T]})$ *if* INIT *is FTL or* Θ *if* INIT *is AOGD*
- *upper bound* G' *on the Lipschitz constants of the functions* $\{\mathcal{B}_R(\hat\theta_t||\cdot)\}_{t\in[T]}$ *over* $\hat\Theta$
- *we will say we are in the "Nice" case if* $\mathcal{B}_R(\theta||\cdot)$ *is 1-strongly-convex and* β*-strongly-smooth w.r.t.* $\|\cdot\|\ \forall\ \theta\in\Theta$
- *in the general case* INIT *is FTL; in the Nice case* INIT *may instead be AOGD*
- *convenience indicator* $\iota = 1_{\mathrm{INIT=FTL}}$
- $\mathrm{TASK}_{\eta,\phi} = \mathrm{FTRL}^{(R)}_{\eta,\phi}$ *or* $\mathrm{OMD}^{(R)}_{\eta,\phi}$

We make the following assumptions:

- *the loss functions* $\ell_{t,i}$ *are convex* $\forall\ t, i$
- *at* $t = 1$ *the update algorithm* INIT *plays* $\phi_1 \in \Theta$ *satisfying* $\max_{\theta\in\Theta}\mathcal{B}_R(\theta||\phi_1) < \infty$
- *in the Approx case* R *is* β*-strongly-smooth for some* $\beta \geq 1$

F.1 Average Regret using Fixed Task Similarity

The following theorem does not appear in the main paper but is used in discussion. It shows guarantees for the case when the task-similarity is known in advance and so SIM always returns a constant.

Theorem F.1. *Make Assumption F.1 and suppose* SIM *always plays* $D_t = \varepsilon$*. Then Algorithm 3 has a regret upper-bound of*

$$\bar{\mathbf{U}}_M \leq \frac{1}{T}\left(\left(\frac{\kappa D^2}{\varepsilon}+\varepsilon\right)\iota\sigma_1 + \frac{\kappa C}{\varepsilon}\sum_{t=1}^T \frac{\sigma_t^2}{\sigma_{1:t}} + \left(\frac{\kappa\nu V^2}{\varepsilon}+\varepsilon\right)\sigma_{1:T} + \frac{\Delta^*_{1:T}}{\varepsilon} + \frac{\kappa\Delta'}{\varepsilon}\right)$$

for $C = \frac{G'^2}{2}$ *in the Nice case or otherwise* $C = 2C'D'G'$*.*

Proof. Let $\{\tilde{\phi}_t\}_{t\in[T]}$ be a "cheating" sequences such that $\tilde{\phi}_t = \phi_t$ on all t except if SIM is FTL and $t = 1$, in which case $\tilde{\phi}_1 = \hat{\theta}_1$. Note that by this definition all upper bounds of $\mathcal{B}_R(\hat{\theta}_t||\phi_t)$ also upper bound $\mathcal{B}_R(\hat{\theta}_t||\tilde{\phi}_t)$. We then use the fact that the actions of FTL at $t > 1$ do not depend on the action at time $t = 1$ to get

$$\begin{aligned}
&\bar{\mathbf{U}}_M T \\
&= \sum_{t=1}^T \frac{\mathcal{B}_R(\theta_t^*||\phi_t)}{\eta_t} + \eta_t G_t^2 m_t \\
&= \frac{\Delta^*_{1:T}}{\varepsilon} + \sum_{t=1}^T \left(\frac{\kappa\mathcal{B}_R(\hat{\theta}_t||\phi_t)}{\varepsilon}+\varepsilon\right)\sigma_t \qquad \text{(substitute } \eta_t = \frac{D_t}{G_t\sqrt{m_t}} \text{ and } D_t = \varepsilon\text{)} \\
&\leq \left(\frac{\kappa D^2}{\varepsilon}+\varepsilon\right)\iota\sigma_1 + \frac{\Delta^*_{1:T}}{\varepsilon} + \sum_{t=1}^T \left(\frac{\kappa\mathcal{B}_R(\hat{\theta}_t||\tilde{\phi}_t)}{\varepsilon}+\varepsilon\right)\sigma_t \qquad \text{(substitute cheating sequence)} \\
&= \left(\frac{\kappa D^2}{\varepsilon}+\varepsilon\right)\iota\sigma_1 + \frac{\Delta^*_{1:T}}{\varepsilon} + \frac{\kappa}{\varepsilon}\sum_{t=1}^T \left(\mathcal{B}_R(\hat{\theta}_t||\tilde{\phi}_t) - \mathcal{B}_R(\hat{\theta}_t||\hat{\phi})\right)\sigma_t + \sum_{t=1}^T \left(\frac{\kappa\mathcal{B}_R(\hat{\theta}_t||\hat{\phi})}{\varepsilon}+\varepsilon\right)\sigma_t \\
&\leq \left(\frac{\kappa D^2}{\varepsilon}+\varepsilon\right)\iota\sigma_1 + \frac{\Delta^*_{1:T}}{\varepsilon} + \frac{\kappa C}{\varepsilon}\sum_{t=1}^T \frac{\sigma_t^2}{\sigma_{1:t}} + \frac{\kappa\Delta'}{\varepsilon} \\
&\qquad + \sum_{t=1}^T \left(\frac{\kappa\nu\mathcal{B}_R(\theta_t'||\phi')}{\varepsilon}+\varepsilon\right)\sigma_t \qquad \text{(Thm. A.2 and Prop. B.1)} \\
&= \left(\frac{\kappa D^2}{\varepsilon}+\varepsilon\right)\iota\sigma_1 + \frac{\Delta^*_{1:T}}{\varepsilon} + \frac{\kappa C}{\varepsilon}\sum_{t=1}^T \frac{\sigma_t^2}{\sigma_{1:t}} + \frac{\kappa\Delta'}{\varepsilon} + \left(\frac{\kappa\nu V^2}{\varepsilon}+\varepsilon\right)\sigma_{1:T}
\end{aligned}$$

□

F.2 Average Regret when Learning Task Similarity

Theorem F.2. *Make Assumption F.1 and let* SIM *be an algorithm running on the sequence of pairs* $\{\mathcal{B}_R(\hat{\theta}_t||\phi_t), \sigma_t\}_{t\in[T]}$ *and at each time* t *having as output the action of an OCO algorithm on the function sequence* $\{\ell_t(x) = (\mathcal{B}_R(\hat{\theta}_t||\phi_t)/x + x)\sigma_t\}_{t\in[T]}$. *Let* $\mathbf{R}_T$ *be the associated regret of this algorithm and suppose it has a parameter* $\varepsilon > 0$ *controlling the minimum action taken. For simplicity assume that at time* $t = 1$ SIM *plays* D_1 *s.t.* $\frac{1}{2}(\max_{\theta\in\Theta}\sqrt{\mathcal{B}_R(\theta||\phi_1)} + \varepsilon) \leq D_1 \leq \max_{\theta\in\Theta}\sqrt{\mathcal{B}_R(\theta||\phi_1)} + \varepsilon$. *Then Algorithm 3 has a regret upper-bound of*

$$\bar{\mathbf{U}}_M \leq \frac{1}{T}\left((2\kappa D + \varepsilon)\iota\sigma_1 + \kappa\,\mathbf{R}_T + \frac{\kappa C}{V}\sum_{t=1}^T \frac{\sigma_t^2}{\sigma_{1:t}} + \kappa(\nu+1)V\sigma_{1:T} + \frac{\Delta^*_{1:T}}{\varepsilon} + \frac{\kappa\Delta'}{V}\right)$$

for $C = \frac{G'^2}{2}$ *in the Nice case or otherwise* $C = 2C'D'G'$.

Proof. Let $\{\tilde{\phi}_t\}_{t\in[T]}$ be "cheating" sequence such that $\tilde{\phi}_t = \phi_t$ on all t except if SIM is FTL and $t = 1$, in which case $\tilde{\phi}_1 = \hat{\theta}_1$. Note that by this definition all upper bounds of $\mathcal{B}_R(\hat{\theta}_t||\phi_t)$ also upper bound $\mathcal{B}_R(\hat{\theta}_t||\tilde{\phi}_t)$. We then have

$$\begin{aligned}
\bar{\mathbf{U}}_M\,T &= \sum_{t=1}^T \frac{\mathcal{B}_R(\theta_t^*||\phi_t)}{\eta_t} + \eta_t G_t^2 m_t \\
&= \frac{\Delta^*_{1:T}}{\varepsilon} + \sum_{t=1}^T\left(\frac{\kappa\mathcal{B}_R(\hat{\theta}_t||\phi_t)}{D_t} + D_t\right)\sigma_t \qquad \text{(substitute } \eta_t = \frac{D_t}{G_t\sqrt{m_t}} \text{ and } D_t \geq \varepsilon) \\
&\leq \left(\frac{\kappa\mathcal{B}_R(\hat{\theta}_t||\phi_t)}{D_1} + D_1\right)\iota\sigma_1 + \frac{\Delta^*_{1:T}}{\varepsilon} \\
&\quad + \sum_{t=1}^T\left(\frac{\kappa\mathcal{B}_R(\hat{\theta}_t||\tilde{\phi}_t)}{D_t} + D_t\right)\sigma_t \qquad \text{(substitute cheating sequences)} \\
&\leq ((\kappa+1)D + \varepsilon)\iota\sigma_1 + \frac{\Delta^*_{1:T}}{\varepsilon} + \kappa\,\mathbf{R}_T + \kappa\sum_{t=1}^T\left(\frac{\mathcal{B}_R(\hat{\theta}_t||\tilde{\phi}_t)}{V} + V\right)\sigma_t \\
&\leq (2\kappa D + \varepsilon)\iota\sigma_1 + \frac{\Delta^*_{1:T}}{\varepsilon} + \kappa\,\mathbf{R}_T + \frac{\kappa C}{V}\sum_{t=1}^T\frac{\sigma_t^2}{\sigma_{1:t}} \\
&\quad + \kappa\sum_{t=1}^T\left(\frac{\mathcal{B}_R(\hat{\theta}_t||\hat{\phi})}{V} + V\right)\sigma_t \qquad \text{(Thm. A.2 and Prop. B.1)} \\
&\leq (2\kappa D + \varepsilon)\iota\sigma_1 + \frac{\Delta^*_{1:T}}{\varepsilon} + \kappa\,\mathbf{R}_T + \frac{\kappa C}{V}\sum_{t=1}^T\frac{\sigma_t^2}{\sigma_{1:t}} + \frac{\kappa\Delta'}{V} + \kappa\sum_{t=1}^T\left(\frac{\nu\mathcal{B}_R(\theta_t'||\phi')}{V} + V\right)\sigma_t \\
&\leq (2\kappa D + \varepsilon)\iota\sigma_1 + \frac{\Delta^*_{1:T}}{\varepsilon} + \kappa\,\mathbf{R}_T + \frac{\kappa C}{V}\sum_{t=1}^T\frac{\sigma_t^2}{\sigma_{1:t}} + \frac{\kappa\Delta'}{V} + \kappa(\nu+1)V\sigma_{1:T}
\end{aligned}$$

□

F.3 Statistical Task-Similarity under Quadratic Growth

In this section we relate our task-similarity measure to that of Denevi et al. [19] under α-QG.

Proposition F.1. *For some distribution $\mathcal{P} \sim \mathcal{Q}$ over losses $\ell : \Theta \mapsto \mathbb{R}_+$ let $\theta^*_{\mathcal{P}} = \arg\min_{\theta\in\Theta} \ell_{\mathcal{P}}(\theta)$ and $\hat{\theta}_m = \arg\min_{\theta\in\Theta} \sum_{i=1}^m \ell_i(\theta)$ for m i.i.d. samples $\ell_i \sim \mathcal{P}$. Define task-similarity measures $V^2 = \min_{\phi\in\Theta} \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}} \|\theta^*_{\mathcal{P}} - \phi\|_2^2$ and $\hat{V}_m^2 = \min_{\phi\in\Theta} \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}} \mathbb{E}_{\mathcal{P}^m} \|\hat{\theta}_m - \phi\|_2^2$. If both $\ell_{\mathcal{P}}$ and $\frac{1}{m}\sum_{i=1}^m \ell_i$ are G-Lipschitz and α-QG a.s. then we have*

$$V^2 \le 2\hat{V}_m^2 + \frac{16G^2}{\alpha^2 m} \qquad \textit{and} \qquad \hat{V}_m^2 \le 2V^2 + \frac{16G^2}{\alpha^2 m}$$

Proof. Following the argument of Shalev-Shwartz et al. [49, Theorem 2] but applying α-QG instead of strong-convexity in Equation 8, which holds by definition of α-QG, we obtain

$$\mathop{\mathbb{E}}_{\mathcal{P}^m}(\ell_{\mathcal{P}}(\hat{\theta}_m) - \ell_{\mathcal{P}}(\theta^*_{\mathcal{P}})) \le \frac{4G^2}{\alpha m}$$

Then for $\phi^* = \arg\min \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}} \|\theta^*_{\mathcal{P}} - \phi\|_2^2$ and $\phi^*_m = \arg\min_{\phi\in\Theta} \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}} \mathbb{E}_{\mathcal{P}^m} \|\hat{\theta}_m - \phi\|_2^2$ we have by these definitions, the triangle inequality, Jensen's inequality, α-QG of $\frac{1}{m}\sum_{i=1}^m \ell_i$, and the above inequality we have

$$\begin{aligned}
\hat{V}_m^2 = \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} \|\hat{\theta}_m - \phi^*_m\|_2^2 &\le \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} \|\hat{\theta}_m - \phi^*\|_2^2 \\
&\le 2 \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} \left(\|\hat{\theta}_m - \theta^*_{\mathcal{P}}\|_2^2 + \|\theta^*_{\mathcal{P}} - \phi^*\|_2^2\right) \\
&\le \frac{4}{\alpha} \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} (\ell_{\mathcal{P}}(\hat{\theta}_m) - \ell_{\mathcal{P}}(\theta^*_{\mathcal{P}})) + 2V^2 \\
&\le \frac{16G^2}{\alpha^2 m} + 2V^2
\end{aligned}$$

Similarly,

$$\begin{aligned}
V^2 = \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} \|\theta_{\mathcal{P}} - \phi^*\|_2^2 &\le \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} \|\theta_{\mathcal{P}} - \phi^*_m\|_2^2 \\
&\le 2 \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} \left(\|\hat{\theta}_m - \theta^*_{\mathcal{P}}\|_2^2 + \|\hat{\theta}_m - \phi^*_m\|_2^2\right) \\
&\le \frac{4}{\alpha} \mathop{\mathbb{E}}_{\mathcal{P}\sim\mathcal{Q}} \mathop{\mathbb{E}}_{\mathcal{P}^m} (\ell_{\mathcal{P}}(\hat{\theta}_m) - \ell_{\mathcal{P}}(\theta^*_{\mathcal{P}})) + 2V^2 \\
&\le \frac{16G^2}{\alpha^2 m} + 2V^2
\end{aligned}$$

□

G Experimental Details

Code is available at `https://github.com/mkhodak/ARUBA`.

G.1 Reptile

For our Reptile experiments we use the code and default settings provided by Nichol et al. [44], except we tune the learning rate, which for ARUBA corresponds to ε/ζ, and the coefficient c in ARUBA++. In addition to the the parameters listed in the above tables, we set $\zeta = p = 1.0$ for all experiments. All evaluations are averages of three runs.

Omniglot 5-way	**1-shot** evaluation setting: regular	transductive	hyperparameters: $\eta = \frac{\varepsilon}{\zeta}$	c	**5-shot** evaluation setting: regular	transductive	hyperparameters: $\eta = \frac{\varepsilon}{\zeta}$	c
MAML (1) [23]		98.3 ± 0.5				99.2 ± 0.2		
Reptile [44]	95.39 ± 0.09	97.68 ± 0.04	$1e-3$		98.90 ± 0.10	99.48 ± 0.06	$1e-3$	
ARUBA	94.57 ± 1.04	97.44 ± 0.32	$1e-1$		98.64 ± 0.04	99.29 ± 0.07	$1e-2$	
ARUBA++	94.80 ± 1.10	97.58 ± 0.13	$1e-1$	10^3	98.93 ± 0.13	99.46 ± 0.02	$1e-2$	10^3
MAML (2)		98.7 ± 0.4				99.9 ± 0.1		
Meta-SGD [38]		99.53 ± 0.26				99.93 ± 0.09		

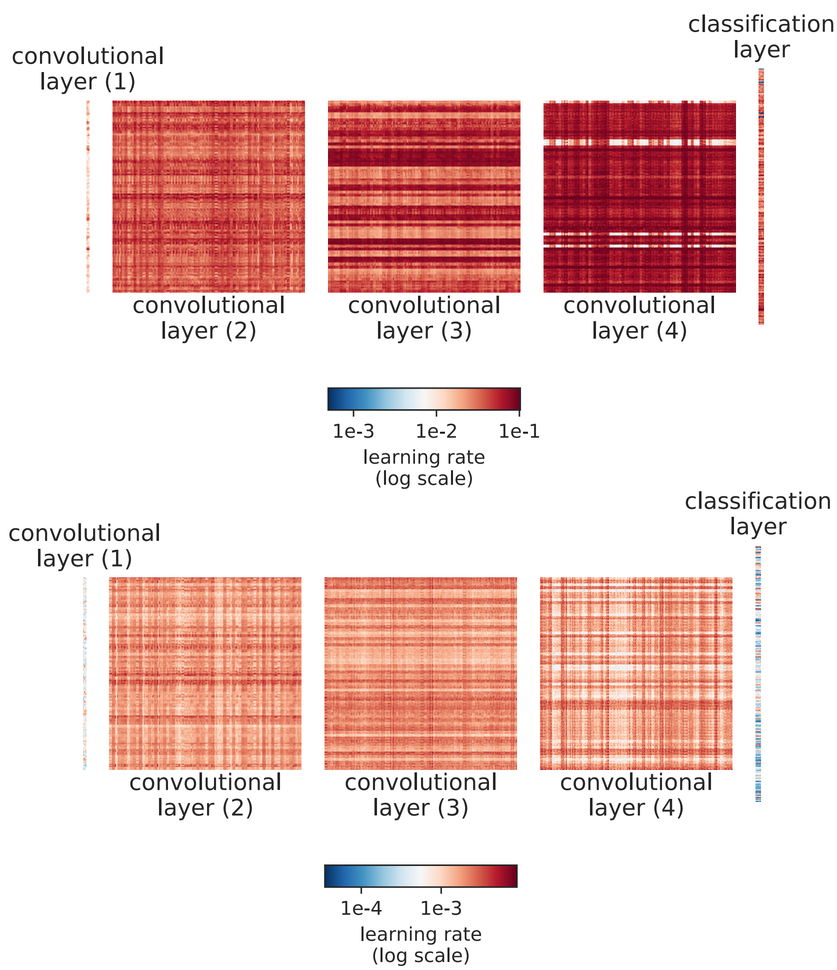

Figure 4: Final learning rate η_T across the layers of a convolutional network trained on 1-shot 5-way Omniglot (top) and 5-shot 5-way Omniglot (bottom) using Algorithm 2 applied to Reptile.

Omniglot 20-way	1-shot				5-shot			
	evaluation setting		hyperparameters		evaluation setting		hyperparameters	
	regular	transductive	$\eta = \frac{\varepsilon}{\zeta}$	c	regular	transductive	$\eta = \frac{\varepsilon}{\zeta}$	c
MAML (1) [23]		95.8 ± 0.3				98.9 ± 0.2		
Reptile [44]	88.14 ± 0.15	89.43 ± 0.14	$5e-4$		96.65 ± 0.33	97.12 ± 0.32	$5e-4$	
ARUBA	85.61 ± 0.25	86.67 ± 0.17	$5e-3$		96.02 ± 0.12	96.61 ± 0.13	$5e-3$	
ARUBA++	88.38 ± 0.24	89.66 ± 0.3	$5e-3$	10^3	96.99 ± 0.35	97.49 ± 0.28	$5e-3$	10
MAML (2)		95.8 ± 0.3				98.9 ± 0.2		
Meta-SGD [38]		95.93 ± 0.38				98.97 ± 0.19		

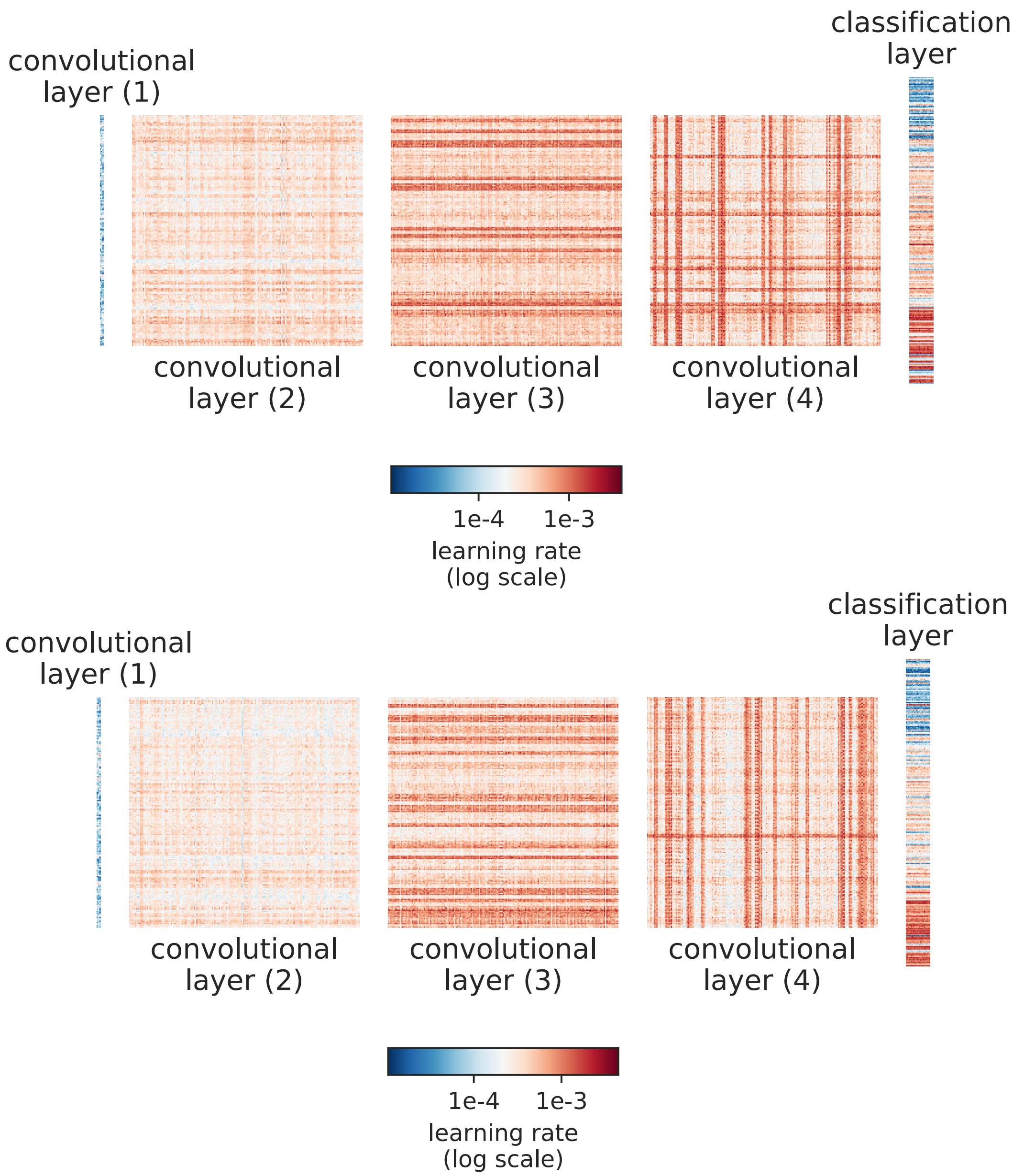

Figure 5: Final learning rate η_T across the layers of a convolutional network trained on 1-shot 20-way Omniglot (top) and 5-shot 20-way Omniglot (bottom) using Algorithm 2 applied to Reptile.

Mini-ImageNet 5-way	1-shot				5-shot			
	evaluation setting		hyperparameters		evaluation setting		hyperparameters	
	regular	transductive	$\eta = \frac{\varepsilon}{\zeta}$	c	regular	transductive	$\eta = \frac{\varepsilon}{\zeta}$	c
MAML (1) [23]		48.07 ± 1.75				63.15 ± 0.91		
Reptile [44]	47.07 ± 0.26	49.97 ± 0.32	$1e-3$		62.74 ± 0.37	65.99 ± 0.58	$1e-3$	
ARUBA	47.01 ± 0.37	50.73 ± 0.32	$5e-3$		62.35 ± 0.25	65.69 ± 0.61	$5e-3$	
ARUBA++	47.25 ± 0.61	50.35 ± 0.74	$5e-3$	10	62.69 ± 0.57	65.89 ± 0.34	$5e-3$	10^{-1}
MAML (2)		48.70 ± 1.84				63.11 ± 0.92		
Meta-SGD [38]		50.47 ± 1.87				64.03 ± 0.94		

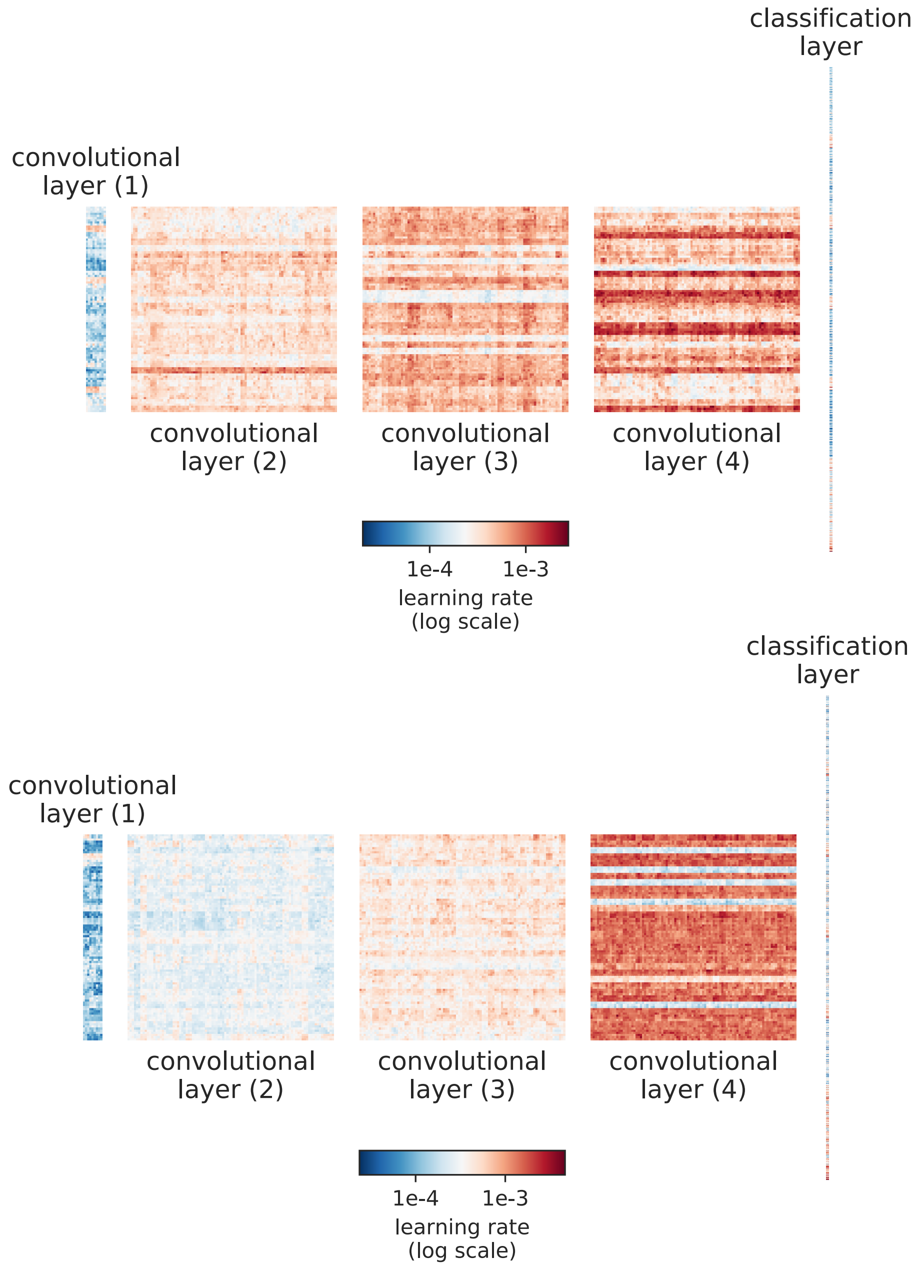

Figure 6: Final learning rate η_T across the layers of a convolutional network trained on 1-shot 5-way Mini-ImageNet (top) and 5-shot 5-way Mini-ImageNet (bottom) using Algorithm 2 applied to Reptile.

G.2 FedAvg

For FedAvg we train a 2-layer stacked LSTM model with 256 hidden units, 8-dimensional trained character embeddings, with a maximum input string size of 80 characters; these settings are used to match those of McMahan et al. [41]. Similarly, we take their approach of only removing those actors from the Shakespeare dataset with fewer than two lines and split each user temporally into train/test sets with a training fraction of 0.8. Unlike McMahan et al. [41], we also split the users into meta-training and meta-testing sets, also with a fraction of 0.8, in order to evaluate meta-test performance. We run both algorithms for 500 rounds with a batch of 10 users per round and a within-task batch-size of 10, as in Caldas et al. [12]. For unmodified FedAvg we found that an initial learning rate of $\eta = 1.0$ worked well – this is similar to those reported in McMahan et al. [41] and Caldas et al. [12] – and for the tuned variant we found that a multiplicative decay of 0.99. At meta-test-time we tuned the refinement learning rate over $\{10^{-3}, 10^{-2}, 10^{-1}\}$. For ARUBA and its isotropic variant we set $\varepsilon = \zeta = 0.05$ and $p = 1.0$, so that $\eta = \varepsilon/\zeta = 1.0$ in our setting as well.

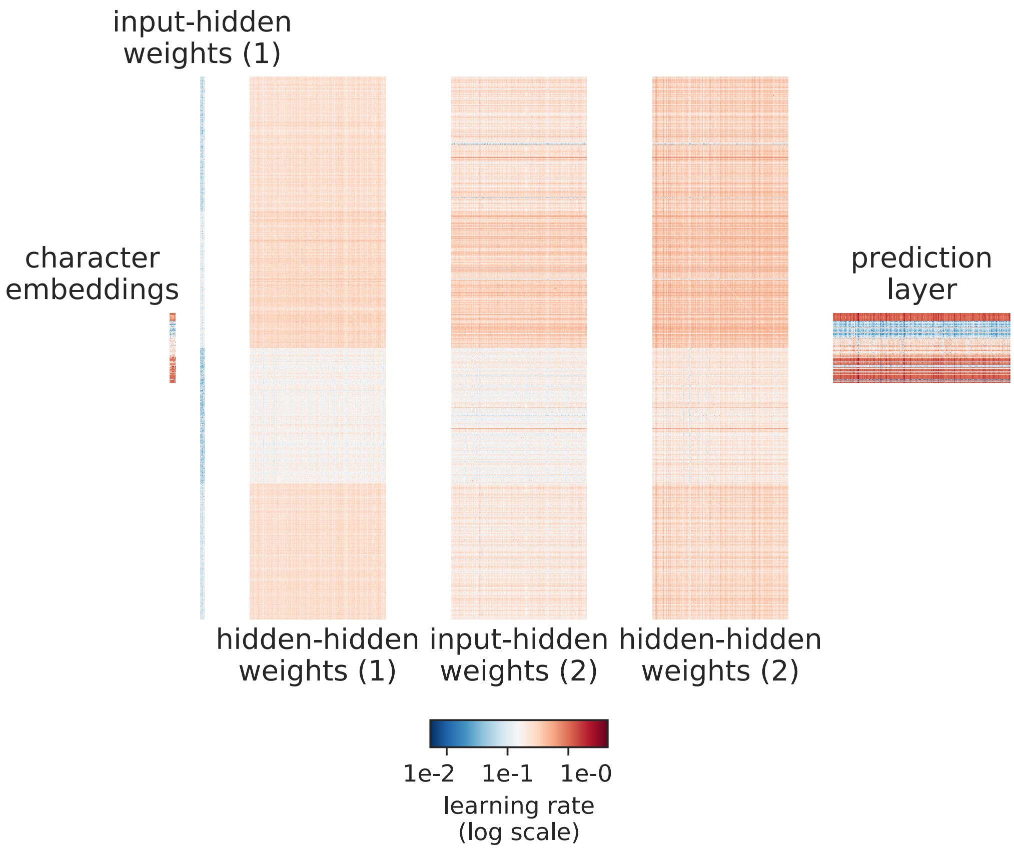

Figure 7: Final learning rate η_T across the layers of an LSTM trained for next-character prediction on the Shakespeare dataset using Algorithm 2 applied to FedAvg.