Optimized Projection-Free Algorithms for Online Learning: Construction and Worst-Case Analysis

Julien Weibel
Inria, D.I. École Normale Supérieure, CNRS
PSL Research University
julien.weibel@inria.fr

Pierre Gaillard
Université Grenoble Alpes, Inria, CNRS
Grenoble INP, LJK
pierre.gaillard@inria.fr

Wouter M. Koolen
Centrum Wiskunde & Informatica, Amsterdam
and Twente University, Enschede
wmkoolen@cwi.nl

Adrien Taylor
Inria, D.I. École Normale Supérieure, CNRS
PSL Research University
adrien.taylor@inria.fr

Abstract

This work studies and develops projection-free algorithms for online learning with linear optimization oracles (a.k.a. Frank–Wolfe) for handling the constraint set, and for convex loss functions. More precisely, this work (i) shows how to exploit semidefinite programming to jointly design and analyze online Frank–Wolfe-type algorithms numerically in a variety of settings, (ii) leverages those design techniques to propose an improved (optimized) variant of an online Frank–Wolfe algorithm along with its conceptually simple potential-based proof, and (iii) extends this proof to its anytime version, which benefits from a similar $O(T^{3/4})$ regret rate without requiring knowledge of the time horizon T in advance. We are not aware of other direct regret guarantees for an anytime version of online Frank–Wolfe without using the classical doubling trick.

Based on the semidefinite technique, we conclude with strong numerical evidence suggesting that no pure online Frank–Wolfe algorithm within our model class can have a regret guarantee better than $O(T^{3/4})$ without additional assumptions, that the current algorithms do not have optimal constants, and that multiple linear optimization rounds do not generally help to obtain better regret bounds.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.
 Copyright 2026 by the author(s).

1 INTRODUCTION

This work considers the online learning problem, where we sequentially query points $x_1, x_2, \dots, x_T$ in a domain $\mathcal{K}$. At each time step t, we play x_t, incur a cost $\ell_t(x_t)$ which we aim to minimize by convention, and observe a gradient $g_t = \nabla \ell_t(x_t)$. The quality of our guesses is evaluated by comparison with a reference $x_\star \in \mathcal{K}$. In this context, we aim to estimate (or bound) the *cumulative regret* R_T incurred by our choices $x_1, \dots, x_T$ as compared to a reference point $x_\star \in \mathcal{K}$:

$$R_T(x_1, \dots, x_T; x_\star) \triangleq \sum_{t=1}^{T} \Big\{ \ell_t(x_t) - \ell_t(x_\star) \Big\} \leqslant \sup_{x \in \mathcal{K}} \left\{ \sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(x) \right\}.$$

In many applications—such as online advertising, sensor networks, or mobile user applications—data is acquired and processed in real time, arriving as a continuous, high-rate flow. This necessitates the adoption of online learning methods, which aim to rapidly integrate large volumes of data as they arrive. Online learning algorithms and regret bounds form essential frameworks for the theoretical study and optimization of reinforcement learning algorithms [Kaelbling et al., 1996, Wang et al., 2022], recommender systems [Bobadilla et al., 2013], or for forecasting time series using expert advice [Cesa-Bianchi and Lugosi, 2006]. These methods find applications in areas like load forecasting [Devaine et al., 2013], finance and portfolio selection [Li and Hoi, 2014], generative adversarial networks [Kodali et al., 2017], and, more recently, large language models [Park et al., 2024]. Online learning has been extensively studied, and we refer to the classical work [Hazan, 2016] and its references for a comprehensive overview of the topic.

Assumptions and problem setup. In this work, we consider the standard situation where the loss functions ℓ_t are convex with bounded gradients $\|g_t\| \leqslant L$ for all $t \geqslant 1$ for simplicity. Similarly, we assume that the domain $\mathcal{K}$ is a closed convex bounded set of diameter D. Variations around those assumptions are discussed in Section 2 and in the appendix.

Related Work. As discussed in classical textbooks (see, e.g., [Hazan, 2016, Orabona, 2019]), a few *meta* algorithms drive the basis and intuitions behind most online learning schemes, which can often be seen as appropriate approximations to the meta algorithms. Prominent examples include the "follow the leader" (FTL) and "follow the regularized leader" (FTRL) methods, as well as online gradient descent (OGD) [Zinkevich, 2003]. These algorithms offer favorable regret guarantees of order $O(\sqrt{T})$. However, classical OGD-type methods rely on projections to manage the constraint set $\mathcal{K}$. In numerous applications, such as matrix completion or recommender systems [Hazan, 2016], these projections are computationally expensive and constitute a significant efficiency bottleneck. This challenge has led to the development of projection-free algorithms, which replace projections with potentially cheaper oracles, allowing for faster computations. In offline smooth convex optimization over polyhedral sets, the pioneering projection-free algorithm was introduced by Frank–Wolfe [Frank and Wolfe, 1956], that resort to a linear optimization oracle. In the context of online convex optimization, the first such algorithm was proposed by [Kalai and Vempala, 2005], but it was limited to linear losses. Subsequently, [Hazan and Kale, 2012] introduced an online version of the Frank–Wolfe algorithm—referred to as online Frank–Wolfe (OFW)—which guarantees a regret upper bound of order $O(T^{3/4})$ for convex losses. This regret rate is less favorable compared to standard online learning algorithms that permit projections. As a result, considerable effort has been directed toward achieving improved regret guarantees for online projection-free algorithms. Some variants have attained the optimal regret rate of $O(\sqrt{T})$ by using membership oracles, which differ from the linear optimization oracles initially used in Frank–Wolfe and may be less efficient in certain scenarios [Levy and Krause, 2019, Mhammedi, 2022]. For linear optimization oracles, regret rates better than $O(T^{3/4})$ have been achieved under additional assumptions. [Hazan and Minasyan, 2020] introduced a stochastic algorithm with $O(T^{2/3})$ expected regret for smooth functions. [Xie et al., 2020] proposed a stochastic algorithm with $\tilde{O}(\sqrt{T})$ expected regret for smooth functions but with $O(T)$ linear optimization oracle calls per round. [Wan and Zhang, 2021] proposed a variant of OFW for strongly convex feasible sets with $O(T^{2/3})$ regret for convex losses and $O(\sqrt{T})$ regret for strongly convex losses.

Despite these advancements, it remains an open question whether online Frank–Wolfe variants, relying on linear optimization oracles, are inherently limited to worst-case regret rate of $\Omega(T^{3/4})$ or if improved rates are possible in our setting.

Algorithmic setup. This work focuses on the specific case of online learning using projection-free algorithms with access to the domain $\mathcal{K}$ only through a linear optimization oracle. At each round t, we play $x_t \in \mathcal{K}$, incur some loss $\ell_t(x_t)$, and observe a gradient $g_t = \nabla \ell_t(x_t)$. We introduce an online procedure that encompasses a family of online Frank–Wolfe-type algorithms and unfolds as follows

$$\begin{aligned} \mathrm{dir}_t &= \sum_{s=1}^{t} \eta_{t,s}\, g_s + \sum_{s=1}^{t-1} \beta_{t,s}\,(v_s - x_1) \\ v_t &= \underset{v \in \mathcal{K}}{\arg\min} \langle \mathrm{dir}_t, v\rangle \\ x_{t+1} &= x_1 + \sum_{s=1}^{t} \gamma_{t+1,s}\,(v_s - x_1), \end{aligned} \tag{1}$$

where the algorithm is parameterized by $\{\eta_{t,s}\}_{1\leqslant s\leqslant t\leqslant T}$ (to combine previous gradient information), $\{\beta_{t,s}\}_{1\leqslant s< t\leqslant T}$ (to pick the linearization point $\bar{x}_t$), and $\{\gamma_{t,s}\}_{1\leqslant s< t\leqslant T}$ to combine the previous *atoms* v_t (extreme points of $\mathcal{K}$) of the linear optimization procedure. In other words, to choose the next query point $x_{t+1} \in \mathcal{K}$, we form the intermediate objective function $f_t(x) = \langle \sum_{s=1}^{t} \eta_{t,s}\, g_s,\, x\rangle + \frac{1}{2}\|x - x_1\|_2^2$, approximate it by the linear function $x \mapsto \langle x, \nabla f_t(\bar{x}_t)\rangle$ in the point $\bar{x}_t = x_1 + \sum_{s=1}^{t-1} \beta_{t,s}\,(v_s - x_1)$ for some sub-probability β_t, and use the linear optimization oracle to optimize that linear approximation over $\mathcal{K}$. The only constraints on algorithm parameters in (1) are that $\gamma_{t,s} \geqslant 0$ for all $s < t \leqslant T$ and $\sum_{s=1}^{t-1} \gamma_{t,s} \leqslant 1$ for all $t \leqslant T$ (we do not constrain $\eta_{t,s}$ and $\beta_{t,s}$ for wider algorithmic design possibilities). Note that translations of the domain $\mathcal{K}$ and initial point x_1 leave the search direction dir_t invariant, while the oracle responses v_t and iterates x_t translate along.

The case [Hazan and Kale, 2012, Algorithm 1 & Theorem 4.4] corresponds to the choices $\eta_{t,s} = \frac{D}{2LT^{3/4}}$, $\beta_{t,s} = \gamma_{t,s}$, $\gamma_{t,t-1} = \frac{1}{\sqrt{t}}$ and $\gamma_{t,s} = \gamma_{t-1,s}(1 - \gamma_{t,t-1})$ for $s < t-1$. As for [Hazan, 2016, Algorithm 27 & Theorem 7.3], they correspond to nearly the same choices with $\gamma_{t,t-1} = \min(1, \frac{2}{\sqrt{t}})$.

Contributions. We show how to leverage semidefinite programming to study (exact) worst-case regrets for online learning algorithms. The method is based on forming a program maximizing the algorithm's regret over the problem instance (loss functions and feasible set), and then reformulating it losslessly as a convex semidefinite program (SDP). Solving this SDP allows us to obtain (exact) worst-case regret values along with their associated worst-case instances. In the optimization literature, those programs are commonly referred to as a *performance estimation problem* (PEP, as coined by [Drori and Teboulle, 2014]). Those PEP methods have been extensively studied in classical (offline) optimization settings, but they have not been explored so far for regret minimization with adversarial functions that may change at every time step.

Most works on online learning algorithms report numerical experiments on synthetic stochastic data or real datasets, even though their theoretical results are stated for worst-case settings (see, e.g., [Kalhan et al., 2021, Moondra et al., 2021, Wan et al., 2022, Zhang et al., 2022]). Numerical experiments with this semidefinite programming method can bridge this gap allowing to investigate tightness of theoretical bounds. Numerical worst-case regret values can also be used to assess performance of algorithms and identify those worth studying theoretically. Proof schemes can then be inferred from numerical values of the SDP's dual variables (see Remark 2.1).

We exploit and exemplify the approach to study OFW algorithms, whose analyses remain relatively technical [Hazan and Kale, 2012, Hazan, 2016] in the literature, and whose known regret bounds of $O(T^{3/4})$ currently remain somewhat surprising. The numerical findings in Section 4 strongly suggest that this rate is non-improvable. As a bonus, Appendix C shows how to apply the same methodological tools to compute tight worst-case regret bounds for OGD and FTRL.

Among the strong points of the approach, it conceptually allows one to optimize the algorithm parameters for better worst-case regret values (through min-max formulation—optimizing worst-case guarantees). However, such problems are naturally non-convex (bilinear matrix inequalities [Toker and Ozbay, 1995]). This work tackles the OFW design PEP problem by providing a numerically-justified tight relaxation for it, which essentially follows from a few simplifications in the original problem along with an appropriate change of variables. This gives rise to a convex SDP that can efficiently be solved numerically. Most importantly, the feasible points of this new program can still be re-interpreted as OFW tunings and proofs. Similar relaxation techniques were, to the best of our knowledge, not used before beyond unconstrained offline gradient-based optimization. However, it is well-known that parametric closed-form solutions to SDPs are highly non-trivial in general (see, Remark 2.1), which is unfortunately the case for the design SDP we obtain for OFW. As a result, the design PEP cannot be directly used to obtain closed-forms for the optimal OFW tuning and its proof, or for a worst-case instance.

To circumvent this issue, we show how to adapt the SDP techniques to exploit/force structure (i.e., simplicity) in the resulting proofs, which we demonstrate with a certain potential-based proof strategy for OFW (see Section 3). Combining this with numerical insights in a constructive approach (see Section 3.3), we obtain an optimized regret bound of $1.74LDT$ for a new variant of OFW (improving on the previous best known bound of $8LDT$ from [Hazan, 2016]), along with its particularly simple and tight tuning for OFW. We then adapt this proof to an anytime variant tuning of OFW (where T is replaced by t in parameters), which to the best of our knowledge is the first direct regret guarantee for an anytime variant of OFW .

Finally, we present numerical PEP evaluations of the exact worst-case regret of OFW, up to moderate horizon $T = 100$, for Hazan's tuning [Hazan, 2016, Hazan and Kale, 2012], our new tuning, and the numerical optimal tuning from the design SDP (Section 4). Those numerics support the following claims: (a) The design PEP allows to construct a likely minimax optimal OFW algorithm, and observe that its regret rate is $\Theta(T^{3/4})$ (implying a regret rate of $\Omega(T^{3/4})$ for all OFW tunings). (b) Our (potential-based) bound for our version of OFW is not tight among non-potential-based bounds (best possible bound is about 2/3 of ours). Note that our theoretical bound for our OFW remains better than the best possible bound for Hazan's OFW (and Hazan's bound is also suboptimal). (c) The tight numerical bounds for our anytime tuning and our horizon-dependent tuning of OFW are close to each other. (d) Allowing for multiple rounds of linear optimization per iteartion in (1) only helps the constant in the regret bounds. (e) We can obtain optimized variations around the same theme, e.g., using $\beta_{t,s} = 0$ all the way. This allows obtaining parameter-free algorithms, at the cost of worsened regret rates.

2 CONSTRUCTIVE APPROACH TO OPTIMIZED REGRET BOUNDS

In this section we present how to leverage semidefinite programming [Vandenberghe and Boyd, 1996] to jointly design and analyze OFW-type algorithms (1) numerically in a variety of settings. Section 2.1 is dedicated to the computation of regret bounds, and Section 2.2 focuses on joint parameter optimization.

2.1 Worst-case regret bounds via semidefinite programming

The core idea for obtaining worst-case regret bounds (and corresponding examples) consists in casting the problem of computing the worst regret for a sequence generated by (1) as an optimization problem, as follows (we omit the (D, L) dependence of B_T for readability):

$$\begin{aligned} & B_T(\{(\eta_{t,s}, \beta_{t,s}, \gamma_{t,s})\}_{t,s}) \triangleq \\ & \sup_{\substack{\mathcal{K}, \{\ell_t\}_{t\in[\![1,T]\!]} \\ x_\star, \{x_t\}_{t\in[\![1,T]\!]} \\ d\in\mathbb{N}}} \quad R_T(x_1, \ldots, x_T; x_\star) \\ & \quad \text{s.t. } \ell_t \text{ is convex and } L\text{-Lipschitz for } t \in [\![1, T]\!], \\ & \qquad \mathcal{K} \text{ is a non-empty closed convex set of } \mathbb{R}^d, \\ & \qquad \mathrm{Diam}(\mathcal{K}) \leqslant D, \\ & \qquad \{x_t\}_{t=1,\ldots,T} \text{ is generated by (1).} \end{aligned} \tag{2}$$

This kind of problem seeks the worst-case dimension $d \in \mathbb{N}$, convex domain $\mathcal{K}$ with bounded diameter, and sequence of Lipschitz convex functions $\{\ell_t\}_{t\in[\![1,T]\!]}$ that together produce the largest possible regret when those losses are evaluated at points $\{x_t\}_{t\in[\![1,T]\!]}$ compatible with (1). In the optimization literature, (2) is commonly referred to as a *performance estimation problem* (PEP, as coined by [Drori and Teboulle, 2014]) which can be reformulated losslessly into a convex SDP [Taylor et al., 2017], as briefly outlined below and further detailed in Appendix B.1.

In a nutshell, (2) is a priori an infinite-dimensional problem (e.g., it includes functional variables such as the losses ℓ_t). A key idea to reformulate (2) as a tractable problem consists in sampling the losses ℓ_t at the query points x_t and $x_\star$, and to treat only the responses $(\nabla \ell_t(x_t), \ell_t(x_t))$ and $(\nabla \ell_t(x_\star), \ell_t(x_\star))$ as variables. By appropriately constraining these responses, we can force them to be compatible with some losses ℓ_t satisfying the desired assumptions (convexity and Lipschitzness of ℓ_t). With a bit more details, it suffices to require that the samples are compatible through the subgradient inequalities $\ell_t(x_t) - \ell_t(x_\star) \leqslant \langle g_t, x_t - x_\star\rangle$ for $t = 1, \cdots, T$ and the Lipschitz bounds $\|\nabla \ell_t(x_t)\| \leqslant L$. In particular, the linearized loss $f_t(x) = \ell_t(x_t) + \langle \ell_t(x_t), x - x_t\rangle$ satisfies the same subgradient inequality as ℓ_t and is also L-Lipschitz, but it yields a larger regret because $\ell_t(x_t) - \ell_t(x_*) \leqslant f_t(x_t) - f_t(x_*)$. Moreover, since the algorithm only observes its gradient at x_t, the linearized loss $f_t(x)$ is indistinguishable from ℓ_t for the algorithm (since $\ell_t(x_t) = f_t(x_t)$ and $\nabla \ell_t(x_t) = \nabla f_t(x_t)$). Thus, the worst-case regret is achieved by linear losses, and we restrict our attention to those losses without loss of generality. As for handling the domain $\mathcal{K}$, one possibility is to sample its indicator function at the query points v_t and impose similar compatibility constraints (see, e.g., [Taylor et al., 2017, Theorem 3.6]). Finally, the sampled version of (2) can be lifted to an SDP via a standard change of variables: all vectors and gradients appearing in (2) are replaced with their Gram matrix (which, recall, encodes all pairwise inner products between these vectors / gradients). The objective, as well as all constraints, then become linear functions of the entries of this Gram matrix. A more complete exposition is provided in Appendix B.1.

Remark 2.1 (Obtaining proofs from (2))**.** A key feature of (2) is that it enables the construction of *algorithm-dependent lower bounds* on the worst-case regret by solving semidefinite programs. That is, for given numerical values of T, L, D and the algorithm parameters, one can compute a worst-case example by solving a tractable convex problem. In order to obtain *algorithm-dependent upper bounds* on the worst-case regret, a natural procedure consists in formulating the *Lagrange dual* of (2) (which is also a semidefinite program; see, e.g., [Vandenberghe and Boyd, 1996, Boyd and Vandenberghe, 2004]), whose feasible points naturally corresponds to upper bounds on the regret. In this context, *finding a proof* consists in finding a feasible point to the dual problem [Goujaud et al., 2023]. While algebraic techniques exist for solving such parametric semidefinite programs in closed form, they typically suffer from exponential complexity in the problem size, number of variables/constraints, and number of parameters [Basu et al., 2006, Naldi et al., 2025]. This motivates the search for *simpler/structured* proofs—e.g., potential-based methods [Bansal and Gupta, 2019]—as exemplified in the proof of Theorem 3.1; see Section 3.3 for further discussion.

2.2 Jointly optimising algorithm parameters and regret bounds

A natural path forward is to use (2) to obtain worst-case optimal algorithms. That is, by solving

$$\begin{aligned} \min_{\{(\eta_{t,s}, \beta_{t,s}, \gamma_{t,s})\}_{t,s}} \quad & B_T(\{(\eta_{t,s}, \beta_{t,s}, \gamma_{t,s})\}_{t,s}) \\ \text{s.t.} \quad & \sum_{s=1}^{t-1} \gamma_{t,s} \leqslant 1,\ \gamma_{t,s} \geqslant 0. \end{aligned}$$

This problem can be formulated as a linear optimization problem with a bilinear matrix inequality constraint (see Appendix B.2), which is unfortunately NP-hard in general [Toker and Ozbay, 1995]. For this reason, we propose a slight relaxation of $B_T(\{(\eta_{t,s}, \beta_{t,s}, \gamma_{t,s})\}_{t,s})$ which corresponds to removing a few constraints from (2) (that we numerically observed to be inactive). More precisely, this relaxation is obtained through: (i) we observe that all x_t for $t = 2, \cdots, T$ are in the convex hull of $x_1, v_1, \cdots, v_{T-1}$, and thus the domain constraints for $\mathcal{K}$ are imposed only on vectors $x_1, v_1, \cdots, v_{T-1}, x_\star$; (ii) we only keep the boundary constraints corresponding to the optimality of v_t compared with $v_{t+1}, \cdots, v_{T-1}, x_\star$:

$$
\begin{aligned}
&W_T\big(\{\eta_{t,s},\beta_{t,s},\gamma_{t,s}\}_{t,s}\big) \triangleq \\
&\sup_{\substack{\{g_t\}_{t\in[\![1,T]\!]},\, x_\star, d\in\mathbb{N}\\ \{(x_t,v_t,\mathrm{dir}_t)\}_{t\in[\![1,T]\!]}}} \sum_{t=1}^{T}\langle g_t, x_t - x_\star\rangle \\
&\quad \text{s.t. } \{(x_t,\mathrm{dir}_t)\}_{t\in[\![1,T]\!]} \text{ compatible with (1)}, \\
&\qquad \langle -\mathrm{dir}_t, u - v_t\rangle \leqslant 0 \text{ for all } t\in[\![1,T-1]\!] \\
&\qquad\qquad \text{and } u \in \{v_{t+1},\cdots,v_{T-1},x_\star\}, \\
&\qquad \mathrm{Diam}(\{x_1,x_\star,v_1,\cdots,v_{T-1}\}) \leqslant D, \\
&\qquad \|g_t\| \leqslant L \text{ for } t=1,\dots,T,
\end{aligned}
$$

where the line "$\{(x_t,\mathrm{dir}_t)\}_{t\in[\![1,T]\!]}$ *compatible with* (1)" means that x_t and dir_t can be substituted by their expressions in (1), leading to $W_T\big(\{\eta_{t,s},\beta_{t,s},\gamma_{t,s}\}_{t,s}\big) \geqslant B_T\big(\{\eta_{t,s},\beta_{t,s},\gamma_{t,s}\}_{t,s}\big)$. For this problem, an appropriate change of variables allows us to cast

$$
\begin{aligned}
\min_{\{(\eta_{t,s},\beta_{t,s},\gamma_{t,s})\}_{t,s}} \; & W_T(\{(\eta_{t,s},\beta_{t,s},\gamma_{t,s})\}_{t,s}) \\
\text{s.t. } & \sum_{s=1}^{t-1}\gamma_{t,s} \leqslant 1,\; \gamma_{t,s}\geqslant 0
\end{aligned}
\tag{3}
$$

as a convex semidefinite program again (details in Appendix B.2). Numerical results illustrating the performance of the numerically optimized methods are provided in Figure 1; examples of numerically optimized parameters are further provided in Appendix B.5.

3 TUNING AND REGRET BOUND FOR ONLINE FRANK–WOLFE

In this section, we analyze the simple OFW scheme (Algorithm 1) with fixed parameters using PEPs to search for simpler potential-based proofs. In Section 3.1, we prove a regret upper bound for our optimized tuning of OFW. Section 3.2 presents the extension to the anytime setting. Section 3.3 explains how the SDP formulations were leveraged to obtain the proof of Section 3.1 and the optimal tuning (4).

Algorithm 1 Online Frank–Wolfe algo. (fixed η, σ)

Require: $T \geqslant 1$, $x_1 \in \mathcal{K}$, $\eta \geqslant 0$, $\sigma \in (0,1)$
1: **for** $t = 1$ to T **do**
2: Play x_t, pay cost $\ell_t(x_t)$, observe $g_t = \nabla\ell_t(x_t)$.
3: $\mathrm{dir}_t \leftarrow \eta\sum_{s=1}^t g_s + (x_t - x_1)$
4: $v_t \leftarrow \arg\min_{v\in\mathcal{K}}\langle \mathrm{dir}_t, v\rangle$
5: $x_{t+1} \leftarrow (1-\sigma)x_t + \sigma v_t$
6: **end for**

3.1 Simple tuning and regret bound

Throughout this section, we fix

$$\eta = \frac{D}{2L}\left(\frac{3}{T}\right)^{3/4} \quad \text{and} \quad \sigma = \min\left(1, \sqrt{\frac{3}{T}}\right). \tag{4}$$

We assume the time horizon is at least $T \geqslant 3$. Note that tuning (4) and the proof of the following theorem were obtained as the optimal solution of a design problem using PEPs for potential-based proofs, as explained in Section 3.3.

Theorem 3.1. *Fix $T \geqslant 3$. Assume that the cost functions ℓ_t are convex and L-Lipschitz for all $t \in [\![1,T]\!]$, and that the convex closed domain $\mathcal{K}$ of feasible points has a diameter bounded by D. Then, for any $x_\star \in \mathcal{K}$, the following upper bound on the regret of the online Frank–Wolfe Algorithm 1, with parameters defined in* (4)*, holds:*

$$
\begin{aligned}
R_T \leqslant \frac{2D}{L3^{3/4}T^{1/4}}\sum_{t=1}^{T}\|g_t\|^2 + \frac{L}{D3^{3/4}T^{1/4}}\sum_{t=1}^{T}\|x_t - v_t\|^2 \\
+ \frac{LT^{3/4}}{D3^{3/4}}\|x_\star - x_1\|^2,
\end{aligned}
$$

and in particular, $R_T \leqslant \frac{4}{3^{3/4}}LDT^{3/4} < 1.76LDT^{3/4}$.

Inspired by [Bansal and Gupta, 2019], see also [Taylor and Bach, 2019, Karimi and Vavasis, 2017], we use a potential-based proof to prove Theorem 3.1, which relies on the next lemma. Motivated by the fact that Algorithm 1 is seen as an approximation of FTRL (see [Hazan, 2016, Chapter 7]), the next lemma will consider a potential that relates the OFW iterates to the iterates of FTRL, which we display in Algorithm 2. To make the connection, it is necessary that OFW and FTRL encounter the same gradients. To ensure that, we formulate the below lemma for linear loss functions.

Algorithm 2 Follow The Regularized Leader (FTRL)

Require: $T \geqslant 1$, $y_1 \in \mathcal{K}$, $\eta \geqslant 0$
1: **for** $t = 1$ to T **do**
2: Play y_t, pay cost $\ell_t(y_t)$, observe $g_t = \nabla\ell_t(y_t)$.
3: $y_{t+1} \leftarrow \arg\min_{y\in\mathcal{K}} \eta\langle\sum_{s=1}^t g_s, y\rangle + \frac{1}{2}\|y - x_1\|^2$
4: **end for**

Lemma 3.2. *Let $T \geqslant 3$. Denote by $\ell_t : x \mapsto \langle g_t, x\rangle$ the linear cost function at time $t \in [\![1,T]\!]$. Assume that $\|g_t\| \leqslant L$ for all $t \in [\![1,T]\!]$ and that the convex closed domain $\mathcal{K}$ of feasible points has a diameter bounded by D. Define the sequence of potentials $(\phi_t)_{0\leqslant t\leqslant T}$ as:*

$$
\begin{aligned}
\phi_t \triangleq \sum_{s=1}^{t}\langle g_s, x_s - y_{t+1}\rangle - \frac{1}{2\eta}\|y_{t+1} - x_1\|^2 \\
+ \frac{1}{6\eta}\|x_{t+1} - y_{t+1}\|^2.
\end{aligned}
$$

Then, Algorithm 1 and 2, run with parameters in (4) *and initialized in $y_1 = x_1 \in \mathcal{K}$, satisfy for all $t \in [\![1,T]\!]$,*

$$\phi_t - \phi_{t-1} \leqslant \frac{2D}{L3^{3/4}T^{1/4}}\|g_t\|^2 + \frac{L}{D3^{3/4}T^{1/4}}\|x_t - v_t\|^2.$$

Let us remark that the potential used in Lemma 3.2 is inspired by the potential ψ_t used to obtain the optimal upper bound on the regret of FTRL (which gives a reformulation of classical FTRL regret proofs; see Appendix A.2 and, e.g., [Orabona, 2019, Chapters 6–7]), where ψ_t is defined as:

$$\psi_t \triangleq \sum_{s=1}^{t} \langle g_s, y_s - y_{t+1}\rangle - \frac{1}{2\eta}\|y_{t+1} - y_1\|^2. \quad (5)$$

The proof of Lemma 3.2 is quite technical and we defer it to Appendix A.1. In Section 3.3, we discuss how we designed the proof of Lemma 3.2 and jointly obtained the optimal choice of the OFW parameters (4) and of the potential function ϕ_t used in Lemma 3.2.

We now prove Theorem 3.1 which relies on Lemma 3.2.

Proof of Theorem 3.1. First, using convexity of the cost functions ℓ_t (recall $g_t = \nabla \ell_t(x_t)$), we get:

$$R_T \leqslant \sum_{t=1}^{T} \langle g_t, x_t - x_\star\rangle.$$

Then, summing the inequalities from Lemma 3.2 applied to the linearized cost functions $\tilde{\ell}_t : x \mapsto \langle g_t, x\rangle$, and observing that $\phi_0 = 0$ as $y_1 = x_1$, we get:

$$\phi_T \leqslant \frac{2D}{L3^{3/4}T^{1/4}} \sum_{t=1}^{T} \|g_t\|^2 + \frac{L}{D3^{3/4}T^{1/4}} \sum_{t=1}^{T} \|x_t - v_t\|^2.$$

Throughout the proof, we will repeatedly use the fact that the optimum of a constrained convex optimization problem $x_* \in \arg\min_{x\in\mathcal{K}} f(x)$ satisfies $\langle \nabla f(x_*), x_* - x\rangle \leqslant 0$ for all $x \in \mathcal{K}$. Using the optimality from the definition of y_{T+1}, we get:

$$\sum_{t=1}^{T} \langle g_t, x_t - x_\star\rangle - \frac{1}{2\eta}\|x_\star - x_1\|^2 \leqslant \sum_{t=1}^{T} \langle g_t, x_t - y_{T+1}\rangle - \frac{1}{2\eta}\|y_{T+1} - x_1\|^2 \leqslant \phi_T.$$

Combining those three inequalities (recall $\eta = \frac{3^{3/4}D}{2LT^{3/4}}$), we get:

$$R_T \leqslant \frac{2D}{L3^{3/4}T^{1/4}} \sum_{t=1}^{T} \|g_t\|^2 + \frac{L}{D3^{3/4}T^{1/4}} \sum_{t=1}^{T} \|x_t - v_t\|^2 + \frac{LT^{3/4}}{D3^{3/4}}\|x_\star - x_1\|^2.$$

Hence, as $\|g_t\| \leqslant L$ and $\|x_t - v_t\| \leqslant D$ for all $t \in [\![1, T]\!]$ and $\|x_\star - x_1\| \leqslant D$, we get that $R_T \leqslant \frac{4}{3^{3/4}} LDT^{3/4}$. This concludes the proof of Theorem 3.1. □

3.2 Extension to an anytime OFW tuning

The proof of Theorem 3.1 can also be adapted to an anytime version of the online Frank–Wolfe algorithm 1 shown in Algorithm 3 where the parameters η_t and σ_t for each time round t are obtained by replacing the horizon T by t in the tuning (4), that is:

$$\eta_t = \frac{D}{2L}\left(\frac{3}{t}\right)^{3/4} \quad \text{and} \quad \sigma_t = \min\left(1, \sqrt{\frac{3}{t}}\right). \quad (6)$$

Note that parameters η_t and σ_t are anytime: they depend on the time round t but not on the horizon T.

Algorithm 3 Anytime online Frank–Wolfe algorithm

Require: $x_1 \in \mathcal{K}$, $\eta_t \geqslant 0$ and $\sigma_t \in (0, 1)$ for $t \geqslant 2$

for $t = 1$ to T (or ∞) **do**
  Play x_t, pay cost $\ell_t(x_t)$, observe $g_t = \nabla \ell_t(x_t)$.
  $\mathrm{dir}_t \leftarrow \eta_{t+1} \sum_{s=1}^{t} g_s + (x_t - x_1)$
  $v_t \leftarrow \arg\min_{v\in\mathcal{K}} \langle \mathrm{dir}_t, v\rangle$
  $x_{t+1} \leftarrow (1 - \sigma_{t+1})x_t + \sigma_{t+1}v_t$
end for

Adapting the proof of Theorem 3.1 to the anytime setting of Algorithm 3, we get the following theorem, whose detailed proof can be found in Appendix A.3.

Theorem 3.3. *Assume that the cost functions ℓ_t are convex and L-Lipschitz, and that the convex closed domain $\mathcal{K}$ of feasible points has a diameter bounded by D. Then, for any $x_\star \in \mathcal{K}$, the following upper bound on the regret of the anytime online Frank–Wolfe Algorithm 3, with parameters defined in* (6)*, holds (simultaneously for all t):*

$$\forall t \geqslant 1, \quad R_t \leqslant \frac{5}{3^{3/4}} LDt^{3/4} < 2.20LDt^{3/4}.$$

3.3 How we got the proof of Theorem 3.1

As explained in Remark 2.1, (2) can be used to obtain rigorous regret upper bounds and their proofs for Algorithm 1. However, the algebraic structure of the problem to be solved proved itself quite challenging. In particular, we did not manage to extract a simple solution from (2). Therefore, we restricted ourselves to search for optimized algorithms within the set of algorithms with simple structured proofs. A classical template for such proofs is that of relying on the construction of *potential* (or Lyapunov) functions (see, e.g., [Bansal and Gupta, 2019] for a nice introduction). We thereby adapted (2) to help searching for appropriate potential functions.

The idea of a potential-based proof is to define a sequence of potentials $(\phi_t)_{0\leqslant t\leqslant T}$ such that (i) $R_T \leqslant \phi_T + A$ and (ii) we can prove inequalities $\phi_t - \phi_{t-1} \leqslant B_t$ for all t, where A and B_t are constants which can only depend on L, D, t and T. Then, we immediately obtain an upper bound on the regret R_T as $R_T \leqslant \phi_0 + A + \sum_{t=1}^{T} \phi_t - \phi_{t-1} \leqslant \phi_0 + A + \sum_{t=1}^{T} B_t$. Under some good choice of potentials ϕ_t, each potential difference $\phi_t - \phi_{t-1}$ depends only on a small number of vectors, gradients and sum of past gradients (for instance, we will use x_t, x_{t+1}, y_t, y_{t+1}, g_t and $G_{t-1} = \sum_{s=1}^{t-1} g_s$). In other words, (ii) consists in studying a single iteration of the procedure while (i) ensures that we can combine those one-iteration analyses to form the global bound.

In such proofs, the first step is therefore to understand what information needs to be summarized in ϕ_t. Here, motivated by the fact that Algorithm 1 is seen as an approximation of FTRL (see [Hazan, 2016, Chapter 7]), and by the potential functions used for upper bounding FTRL's regret (see (5), which is a reformulation of classical FTRL regret proofs; see, e.g., [Orabona, 2019, Chapters 6–7]), we tried the family of potentials parametrised by a and b as

$$\begin{aligned}\phi_t =& \sum_{s=1}^{t} \langle g_s, x_s - y_{t+1}\rangle - \frac{1}{2\eta}\|y_{t+1} - x_1\|^2 \\ &+ a\,\|x_{t+1} - y_{t+1}\|^2 + b\,\eta\,\langle G_t, x_{t+1} - y_{t+1}\rangle \\ &+ \frac{b}{2}\left(\|x_{t+1} - x_1\|^2 - \|y_{t+1} - x_1\|^2\right).\end{aligned} \tag{7}$$

Note that $\phi_t - \phi_{t-1}$ is *autonomous*: it is a function of t only through its dependence in $x_t, x_{t+1}, y_t, y_{t+1}, g_t, G_{t-1}, x_1$. Because of this fact, and as x_{t+1}, y_t and y_{t+1} are obtained as autonomous functions of x_t, g_t, G_{t-1} and x_1 using Algorithm 1 and FTRL, it is possible to use the same proof for upper bounding $\phi_t - \phi_{t-1}$ for all iteration times t (then, we get that B_t is also autonomous, which imposes that $B_t = B$ is a constant that depends only on L, D and T). Thus, we can use only 1 iteration of upper bounding $\phi_t - \phi_{t-1} \leqslant B$ for an abstract t instead of doing an upper bound for each t individually. We also note that $\phi_0 = 0$ and $R_T - \frac{1}{2\eta}\|x_\star - x_1\|^2 \leqslant \phi_T$ for $a \geqslant 0$ and $b \geqslant 0$ (using optimality of y_{T+1}).

To jointly design the proof of Algorithm 1 (with fixed η and σ) and the potential (7), we use the following 1-iteration variant of (2):

$$\begin{aligned}\inf_{a\geqslant 0, b\geqslant 0}\ \sup_{\substack{\mathcal{K},\, d\in\mathbb{N}\\ g_t,\, G_{t-1}\\ x_1, x_t, v_t, x_{t+1}\\ y_t,\, y_{t+1}}} &\ \phi_t - \phi_{t-1} \\ \text{s.t. } &\phi_t - \phi_{t-1} \text{ is generated by (7) from} \\ &\qquad \{x_t, x_{t+1}, y_t, y_{t+1}, g_t, G_{t-1}, x_1\}, \\ &\|g_t\| \leqslant L, \\ &\mathcal{K} \subset \mathbb{R}^d \text{ is non-empty, closed, convex,} \\ &\mathrm{Diam}(\{x_1, x_t, v_t, x_{t+1}, y_t, y_{t+1}\}) \leqslant D, \\ &(x_{t+1}, v_t) \text{ are generated from} \\ &\qquad \{x_1, x_t, g_t, G_{t-1}\} \text{ by Algorithm 1,} \\ &y_t \text{ and } y_{t+1} \text{ are generated from} \\ &\qquad \{x_1, g_t, G_{t-1}\} \text{ by FTRL.}\end{aligned} \tag{8}$$

Note that (8) can be rewritten as a semidefinite program whose size does not depend on T, which allows efficient numerical solving even for large values of T.

Then, numerically solving (8) for fixed η and σ gives $b = 0$ in (7) and the values of Lagrange multipliers of the constraints gives the proof structure of Lemma 3.2 (with literal values for η, σ, a, and for two Lagrange multipliers) up to the final step of verifying that the remaining terms form a sum of squares. Hence, we are left with finding the optimal η and σ minimizing the regret upper bound given by (8) with the constraint that the remaining terms form a sum of squares, which is a non-convex problem. We then relax this non-convex problem by keeping only the leading order terms in the sum of squares constraint, which we can then solve algebraically, resulting in the choices

$$\eta = \frac{D3^{3/4}}{2LT^{3/4}}, \quad \sigma = \frac{\sqrt{3}}{\sqrt{T}}, \quad \text{and} \quad a = \frac{1}{6\eta},$$

thereby concluding the construction of the result of Lemma 3.2. The details for rewriting (8) as a semidefinite program, numerically solving it, and then for obtaining the optimal parameters above can be found in Appendix B.3.

4 NUMERICAL RESULTS: TIGHT REGRET BOUNDS, DIRECT STEPSIZE OPTIMIZATION

In this section, we leverage the techniques of Section 2 to provide strong numerical evidence on the regret behaviors of different variations of online Frank–Wolfe-type algorithms.

The numerical experiments of this section rely on the CVXPY [Diamond and Boyd, 2016] modeling language used in combination with the MOSEK semidefinite solver [ApS, 2019] for solving (3). For computing (2) we directly implemented the online algorithms within the PEPit software [Goujaud et al., 2024]. In all those numerical experiments, we used $L = D = 1$. Those numerical experiments were performed on a MacBook Pro 14" with M3 Pro SoC and 36GB of RAM in a few tens of minutes for the largest time horizon and in a few minutes for other values (see Appendix B.4 for more details on the computational times), note that RAM size was the main limiting factor. (Our code used for those numerical experiments is available at `https://github.com/JulienWeibel/Optimized-projection-free-algorithms-for-online-learning-construction-and-worst-case-analysis`.) Those numerical experiments strongly support the following claims, see Figure 1. In Figure 1, we also compare the tight numerical bounds obtained from PEPs to known upper bounds for OFW, referring to their theorems in the caption of Figure 1.

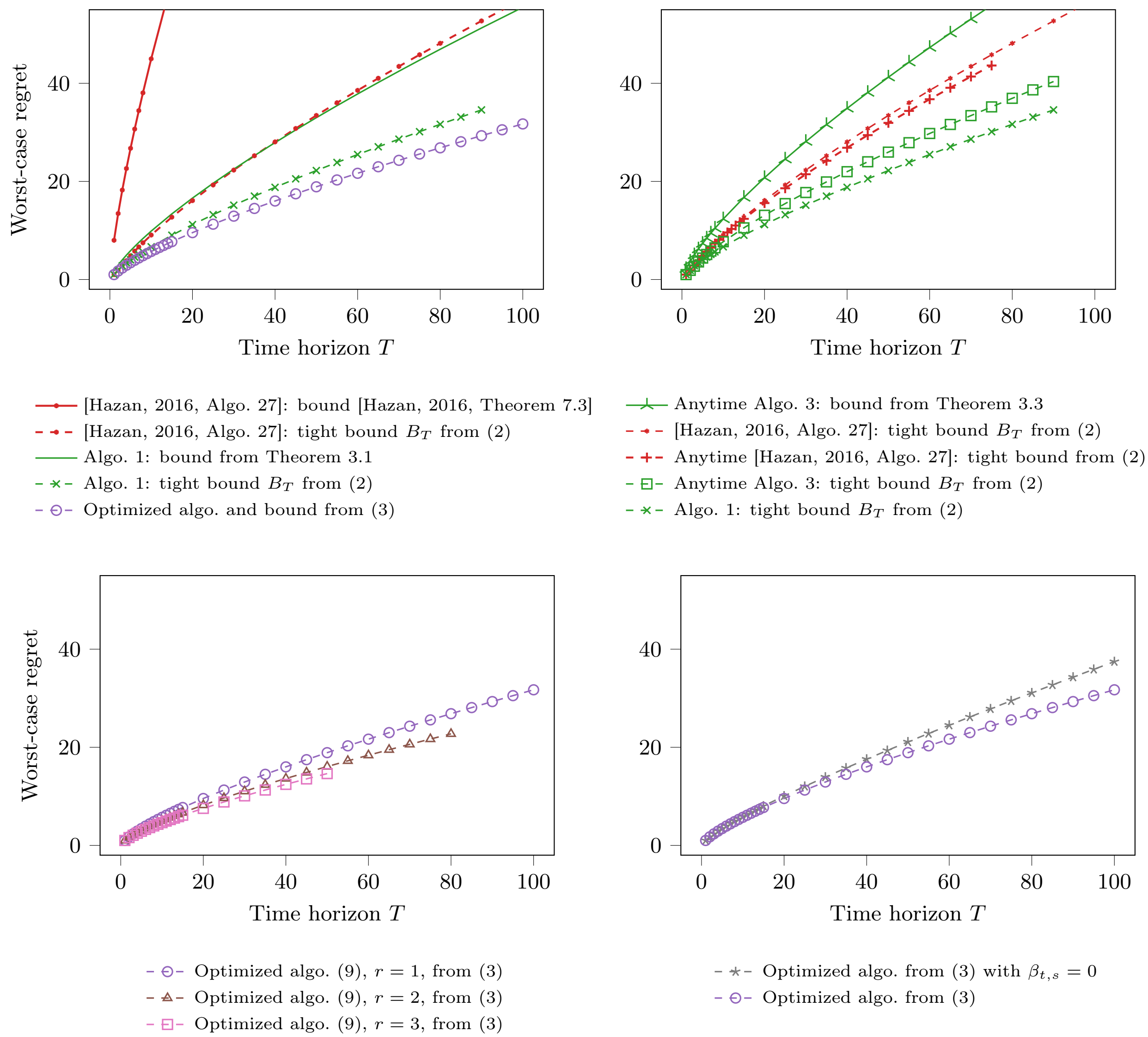

Figure 1: (Top left) Comparison of known upper bounds (respectively from [Hazan, 2016, Theorem 7.3] and Theorem 3.1) against tight numerical bounds (worst-case regrets) obtained from (2), for [Hazan, 2016, Algorithm 27] and Algorithm 1 (parameters from Theorem 3.1). (Top right) Tight numerical regret bounds for [Hazan, 2016, Algorithm 27] and Algorithm 1 (parameters from Theorem 3.1) against their anytime versions. (Bottom left) Tight numerical bounds for optimized online Frank–Wolfe with respectively $r \in \{1, 2, 3\}$ linear optimization steps per time round (where (41) is a variant of (3) with (1) replaced by (9), which we detail in Appendix B.5). (Bottom right) Tight numerical regret bounds for optimized online Frank–Wolfe with and without regularization (i.e., (3) with and without $\beta_{t,s} = 0$).

Comparison of known upper bounds to tight numerical bounds. (Top left of Figure 1.) Algorithm 1, instantiated with parameters in Equation (4), enjoys a regret guarantee that improves upon the classical bound $8LDT$ from [Hazan, 2016, Theorem 7.3] (corresponding to Algorithm 27 therein). This improvement also holds when comparing the tight bounds of both algorithms, with a gain of approximately 34% (i.e., a factor of about 0.66). Although the upper bound provided in Theorem 3.1 is not tight, it remains within a constant factor of approximately 1.5 of the numerically computed tight worst-case bound using (2). Note that the difference between those two bounds is due to Theorem 3.1 being optimal among simple potential-based proofs as described in Section 3.3, while the tight numerical bound is optimal among all possible proofs (including non-potential-based ones). Moreover, Algorithm 1 (with the parameters from Theorem 3.1) is near-optimal among online Frank–Wolfe algorithms of the form (1), up to a constant multiplicative factor of roughly 1.18. Note that the difference between those two tight numerical bounds is due to them being optimal for different classes of algorithms: tuning (4) is optimal among the simpler variants of OFW defined by Algorithm 1, while (3) gives optimal tuning among the wider class of algorithms defined by (1), which includes Algorithm 1 as a special case.

Anytime variants. (Top right of Figure 1.) The anytime algorithmic variants presented in Figure 1 correspond to respectively [Hazan, 2016, Algorithm 27] and Algorithm 1 (with parameter choices from Theorem 3.1) where the time-horizon T dependence of algorithm parameters is replaced by the current time t. We observe that the anytime variant of Algorithm 1 (that is, Algorithm 3) has a guarantee close to that of the original Algorithm 1 within a constant multiplicative factor of about 1.17. The anytime variant of [Hazan, 2016, Algorithm 27] has a better guarantee than the original [Hazan, 2016, Algorithm 27], but still worse than the anytime variant of Algorithm 1 (with a constant multiplicative factor of about 1.24). Although the upper bound provided in Theorem 3.3 is not tight, it remains within a constant factor of approximately 1.6 of the numerically computed tight worst-case bound.

Multiple linear optimization rounds per iteration. (Bottom left of Figure 1.) A natural extension of (1) consists in performing a fixed number ($r > 1$) of linear optimization steps per iteration by defining r search directions $\mathrm{dir}_{t,1}, \ldots, \mathrm{dir}_{t,r}$ (sequentially) per time step, together with the corresponding atoms $v_{t,1}, \ldots, v_{t,k}$:

$$\begin{aligned} \mathrm{dir}_{t,k} &= \sum_{s=1}^{t} \eta_{t,k,s}\, g_s + \sum_{s=1}^{t-1} \sum_{j=1}^{r} \beta_{t,k,s,j}\, (v_{s,j} - x_1) \\ &\qquad + \sum_{j=1}^{k-1} \beta_{t,k,t,j}\, (v_{t,j} - x_1) \qquad (9) \\ v_{t,k} &= \operatorname*{arg\,min}_{v \in \mathcal{K}} \langle \mathrm{dir}_{t,k}, v \rangle \end{aligned}$$

for choosing the next query point $x_{t+1} = x_1 + \sum_{s=1}^{t} \sum_{k=1}^{r} \gamma_{t+1,s,k}\, (v_{s,k} - x_1)$. We observe that multiple rounds (fixed in advance and not a function of the time horizon) of linear optimization oracles do not help improve the regret rates of online Frank–Wolfe algorithms. In all cases, the regret scales as $T^{3/4}$. (For a clearer view of this rate, log–log plot variants of Figure 1 are provided in Appendix B.7.) Hence, this gives a strong numerical conjecture that all variants of OFW within our model class have a regret rate of $\Omega(T^{3/4})$. The numerical regret bounds for optimized online Frank–Wolfe algorithms with fixed number $r > 1$ of linear optimization oracle calls per time round were obtained in a similar way to the method outlined in Section 2.2, see Appendix B.6 for details.

Unregularized online Frank–Wolfe. (Bottom right of Figure 1.) A desirable feature of setting $\beta_{t,s} = 0$ in (1) is that the remaining parameters $\eta_{t,s}$ and $\gamma_{t,s}$ are naturally dimension-independent, that is, they cannot depend on L and D in a meaningful way (as there is no other external quantities, there is no way to have $\eta_{t,s}$ and $\gamma_{t,s}$ being dimension-independent while depending on L and D). Unfortunately, this nice feature is counter-balanced by an apparent worse regret rate. More precisely, the optimized Frank–Wolfe algorithms with fixed $\beta_{t,s} = 0$ appear to have their worst-case regrets scaling as $O(T^{\alpha})$ for $\alpha \approx 7/8$.

5 CONCLUSION

In this work, we have studied and developed projection-free algorithms for online learning that rely on linear optimization oracles (a.k.a. Frank–Wolfe) for handling the constraint set, and for convex loss functions. More precisely, the contributions of this work are (i) to formulate the problem of analyzing and designing online Frank–Wolfe-type algorithms as semidefinite problems (SDPs) that can be solved numerically in a variety of settings, (ii) how to use those design methods to propose an improved (optimized) variant of an online Frank–Wolfe algorithm (Algorithm 1), along with its conceptually simple potential-based proof, and (iii) its anytime version, which benefits from a similar $O(T^{3/4})$ regret rate without requiring knowledge of the time horizon T in advance. This SDP methodology provides a constructive and principled approach to regret bounds and their corresponding worst-case instances. Algorithms with optimal regret guarantees can then be designed by jointly optimising the algorithm parameters and the regret bound. We then leveraged those techniques to perform rigorous numerical experiments strongly supporting (a) near-optimality claims of the proposed parameter choices, and (b) that all OFW-type algorithms have a regret rate of $\Omega(T^{3/4})$. Finally, we explained how the presented SDP approach can be used to obtain conceptually simple proofs with optimal regret bounds, taking as an example our potential-based proof from part (ii).

The findings of our numerical experiments motivate the following future work opportunities: (1) Can we find a tighter analysis of the OFW algorithm? (2) Can we find a closed-form for the optimal OFW-type methods? (3) Can we find a closed-form solution and the corresponding regret bound for the optimal unregularized OFW-type method?

Acknowledgments

This work took place in the context of the associate team 4TUNE within CWI-Inria international lab. Julien Weibel and Adrien Taylor are supported by the European Union (ERC grant CASPER 101162889). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. The French government also partly funded this work under the management of Agence Nationale de la Recherche as part of the "France 2030" program, reference ANR-23-IACL-0008 (PR[AI]RIE-PSAI) and reference ANR-23-IACL-0006 (MIAI Cluster).

The author would like to thank Shuvomoy Das Gupta and anonymous referees for helpful comments which helped improved the presentation of this article.

References

[ApS, 2019] ApS, M. (2019). Mosek optimization suite.

[Bansal and Gupta, 2019] Bansal, N. and Gupta, A. (2019). Potential-function proofs for gradient methods. *Theory of Computing*, 15(1):1–32.

[Basu et al., 2006] Basu, S., Pollack, R., and Roy, M.-F. (2006). *Algorithms in Real Algebraic Geometry*. Algorithms and Computation in Mathematics. Springer-Verlag.

[Bobadilla et al., 2013] Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-based systems*, 46:109–132.

[Boyd and Vandenberghe, 2004] Boyd, S. P. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[Cesa-Bianchi and Lugosi, 2006] Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

[Das Gupta et al., 2024] Das Gupta, S., Van Parys, B. P. G., and Ryu, E. K. (2024). Branch-and-bound performance estimation programming: A unified methodology for constructing optimal optimization methods. *Mathematical Programming*, 204(1-2):567–639.

[Devaine et al., 2013] Devaine, M., Gaillard, P., Goude, Y., and Stoltz, G. (2013). Forecasting electricity consumption by aggregating specialized experts: A review of the sequential aggregation of specialized experts, with an application to slovakian and french country-wide one-day-ahead (half-) hourly predictions. *Machine Learning*, 90:231–260.

[Diamond and Boyd, 2016] Diamond, S. and Boyd, S. P. (2016). Cvxpy: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research (JMLR)*, 17(83):1–5.

[Drori and Teboulle, 2014] Drori, Y. and Teboulle, M. (2014). Performance of first-order methods for smooth convex minimization: A novel approach. *Mathematical Programming*, 145(1-2):451–482.

[Frank and Wolfe, 1956] Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110.

[Goujaud et al., 2023] Goujaud, B., Dieuleveut, A., and Taylor, A. (2023). On fundamental proof structures in first-order optimization. In *Conference on Decision and Control (CDC)*.

[Goujaud et al., 2024] Goujaud, B., Moucer, C., Glineur, F., Hendrickx, J. M., Taylor, A. B., and Dieuleveut, A. (2024). PEPit: Computer-assisted worst-case analyses of first-order optimization methods in Python. *Mathematical Programming Computation*, 16(3):337–367.

[Hazan, 2016] Hazan, E. (2016). Introduction to Online Convex Optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325.

[Hazan and Kale, 2012] Hazan, E. and Kale, S. (2012). Projection-free online learning. In *International Conference on Machine Learning (ICML)*.

[Hazan and Minasyan, 2020] Hazan, E. and Minasyan, E. (2020). Faster projection-free online learning. In *Conference on Learning Theory (COLT)*.

[Kaelbling et al., 1996] Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

[Kalai and Vempala, 2005] Kalai, A. and Vempala, S. (2005). Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307.

[Kalhan et al., 2021] Kalhan, D. S., Singh Bedi, A., Koppel, A., Rajawat, K., Hassani, H., Gupta, A. K., and Banerjee, A. (2021). Dynamic online learning via frank-wolfe algorithm. *IEEE Transactions on Signal Processing*, 69:932–947.

[Karimi and Vavasis, 2017] Karimi, S. and Vavasis, S. (2017). A single potential governing convergence of conjugate gradient, accelerated gradient and geometric descent. *arXiv preprint arXiv:1712.09498*.

[Kodali et al., 2017] Kodali, N., Abernethy, J., Hays, J., and Kira, Z. (2017). On convergence and stability of gans. *preprint arXiv:1705.07215.*

[Levy and Krause, 2019] Levy, K. and Krause, A. (2019). Projection free online learning over smooth sets. In *International Conference on Artificial Intelligence and Statistics (AISTATS).*

[Li and Hoi, 2014] Li, B. and Hoi, S. C. (2014). Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)*, 46(3):1–36.

[Mhammedi, 2022] Mhammedi, Z. (2022). Efficient projection-free online convex optimization with membership oracle. In *Conference on Learning Theory (COLT).*

[Moondra et al., 2021] Moondra, J., Mortagy, H., and Gupta, S. (2021). Reusing combinatorial structure: faster iterative projections over submodular base polytopes. *Advances in Neural Information Processing Systems*, 34:25386–25399.

[Naldi et al., 2025] Naldi, S., Din, M. S. E., Taylor, A., and Wang, W. (2025). Solving generic parametric linear matrix inequalities. *preprint arXiv:2503.01487.*

[Orabona, 2019] Orabona, F. (2019). A modern introduction to online learning. *preprint arXiv:1912.13213.*

[Park et al., 2024] Park, C., Liu, X., Ozdaglar, A., and Zhang, K. (2024). Do LLM agents have regret? a case study in online learning and games. *preprint arXiv:2403.16843.*

[Taylor and Bach, 2019] Taylor, A. and Bach, F. (2019). Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions. In *Conference on Learning Theory (COLT).*

[Taylor et al., 2017] Taylor, A. B., Hendrickx, J. M., and Glineur, F. (2017). Exact worst-case performance of first-order methods for composite convex optimization. *SIAM Journal on Optimization*, 27(3):1283–1313.

[Toker and Ozbay, 1995] Toker, O. and Ozbay, H. (1995). On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. In *American Control Conference (ACC).*

[Vandenberghe and Boyd, 1996] Vandenberghe, L. and Boyd, S. P. (1996). Semidefinite programming. *SIAM review*, 38(1):49–95.

[Wan et al., 2022] Wan, Y., Tu, W.-W., and Zhang, L. (2022). Online frank-wolfe with arbitrary delays. *Advances in Neural Information Processing Systems*, 35:19703–19715.

[Wan and Zhang, 2021] Wan, Y. and Zhang, L. (2021). Projection-free online learning over strongly convex sets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(11):10076–10084.

[Wang et al., 2022] Wang, X., Wang, S., Liang, X., Zhao, D., Huang, J., Xu, X., Dai, B., and Miao, Q. (2022). Deep reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):5064–5078.

[Xie et al., 2020] Xie, J., Shen, Z., Zhang, C., Wang, B., and Qian, H. (2020). Efficient projection-free online methods with stochastic recursive gradient. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6446–6453.

[Zhang et al., 2022] Zhang, L., Wang, G., Yi, J., and Yang, T. (2022). A simple yet universal strategy for online convex optimization. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26605–26623. PMLR.

[Zinkevich, 2003] Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML).*

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**] Mathematical settings and assumptions are represented in the introduction section and restated in theorems. Algorithms are presented in dedicated display environments.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Not Applicable**] We only consider algorithms from an OFW class that all perform one linear optimization step per time round (same time complexity), and that all have access to exactly one gradient per time step (same sample size complexity). The class contains algorithms with complete memory, but the algorithms studied in the theorems all work with constant space.

   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries.
   [**No**] Open-source and user-friendly code relying on standard external software packages will be released upon acceptance of this work at the conference.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results.
   [**Yes**] All the theoretical results of this work can be found in Section 3. Those theoretical results include their full set of assumptions.
   (b) Complete proofs of all theoretical results.
   [**Yes**] Complete and correct proofs of all theoretical results can be found in Section 3, except for the more technical proofs which can be found in Appendix A.1 (with comments in Section 3 pointing to Appendix A.1).
   (c) Clear explanations of any assumptions.
   [**Yes**] The assumptions used—which are standard in the literature—are presented and explained in the introduction section.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL).
   [**Yes**] The techniques used for reproducing all the numerical results presented in this work are explained in Section 2—and all omitted details are provided in Appendix B. Furthermore, the Python code will be released on an open repository and will be easily executed with standard software suites.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen).
   [**Yes**] All numerical results of this work were obtained via classical convex optimization packages [ApS, 2019, Diamond and Boyd, 2016], but other ones could have been used instead (deterministic convex optimization).
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times).
   [**Not Applicable**] The numerical experiments in this work are deterministic (they do not involve any probability distributions or dataset), thus the notions of statistical significance and error bars are not relevant here.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider).
   [**Yes**] Numerical experiments are cheap and can be run on any modern laptop. We included a comment with the model of modern laptop used to run those numerical experiments, an upper bound on computation time, and a remark about RAM size being a limiting factor for running larger experiments.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**]
   (b) The license information of the assets, if applicable. [**Not Applicable**]
   (c) New assets either in the supplemental material or as a URL, if applicable. [**Not Applicable**]
   (d) Information about consent from data providers/curators. [**Not Applicable**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [**Not Applicable**]

   The classical convex optimization packages [ApS, 2019, Diamond and Boyd, 2016] used to run the numerical experiments of this work are properly cited in the paper. No other asset is used in this work. Our paper does not introduce any new asset.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [**Not Applicable**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [**Not Applicable**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [**Not Applicable**]

   We did not use crowdsourcing nor conduct research with human subjects.

Instructions for Paper Submissions to AISTATS 2026: Supplementary Materials

Contents

A Omitted proofs

A.1 Detailed proof of Lemma 3.2

We first discuss the main difference between our proof scheme for upper bounding the regret of OFW and that of Hazan [Hazan, 2016, Theorem 7.3]. Hazan's proof scheme is based on proving by induction that $F_{t+1}(x_t) - F_{t+1}(y_{t+1}) \leqslant 2D\sigma_t$ where for all t, the function $F_t : x \mapsto \eta \sum_{s=1}^{t-1} \langle g_s, x\rangle + \frac{1}{2}\|x - x_1\|^2$ is the intermediate objective function of the FTRL iterate y_t. Using the strong convexity of the functions F_t, this bound can be transformed into a bound on the norm distance between the OFW iterates x_t and the anticipative[1] FTRL iterates y_{t+1}, which then gives a regret upper bound for OFW involving the regret upper bound of FTRL and the norm distance errors $\|x_t - y_{t+1}\|^2$. On the other hand, our proof scheme for OFW is potential-based and stores information on the closeness of the OFW iterates x_{t+1} and the FTRL iterates y_{t+1} through the norm term $\frac{1}{6\eta}\|x_{t+1} - y_{t+1}\|^2$ in the potential ϕ_t rather than through the FTRL intermediate functions with $F_{t+1}(x_t) - F_{t+1}(y_{t+1})$ in Hazan's proof scheme. A benefit of our more direct approach is that when upper bounding the potential increase $\phi_t - \phi_{t-1}$, we get that either the regret or the norm term $\|x_t - y_{t+1}\|^2$ can be "large" but not both simultaneously, whereas with Hazan's proof scheme both could be "large" simultaneously. This allows our proof scheme to give tighter regret upper bounds for OFW than Hazan's proof scheme. Another improvement of our proof scheme is to work with linearized cost functions, which allows the OFW and FTRL iterates to use the same gradients (in Hazan's proof scheme, the mismatch between gradients for OFW and FTRL creates an extra error term).

Proof of Lemma 3.2. Let $t \in [\![1, T]\!]$ be fixed. We are going to prove the upper bound on $\phi_t - \phi_{t-1}$. Denote by $G_{t-1} = \sum_{s=1}^{t-1} g_s$ the sum of past gradients, so that $G_t = G_{t-1} + g_t$. Note that we have:

$$\begin{aligned}\phi_t - \phi_{t-1} &= \langle g_t, x_t - y_{t+1}\rangle + \langle G_{t-1}, y_t - y_{t+1}\rangle \\ &\qquad + \frac{1}{6\eta}(\|x_{t+1} - y_{t+1}\|^2 - \|x_t - y_t\|^2) + \frac{1}{2\eta}(\|y_t - x_1\|^2 - \|y_{t+1} - x_1\|^2).\end{aligned}$$

Using the optimality in the definition of y_t, we have that $\langle G_{t-1}, y_t - y_{t+1}\rangle \leqslant \frac{1}{\eta}\langle y_t - x_1, y_{t+1} - y_t\rangle$, which gives us:

$$\phi_t - \phi_{t-1} \leqslant \langle g_t, x_t - y_{t+1}\rangle + \frac{1}{6\eta}(\|x_{t+1} - y_{t+1}\|^2 - \|x_t - y_t\|^2) - \frac{1}{2\eta}\|y_t - y_{t+1}\|^2. \tag{10}$$

From the optimality in the definitions of v_t and y_{t+1}, we have:

$$\begin{aligned} &\langle G_t, v_t - y_{t+1}\rangle + \frac{1}{\eta}\langle x_t - x_1, v_t - y_{t+1}\rangle \leqslant 0 \\ \text{and} \quad &\langle G_t, y_{t+1} - v_t\rangle + \frac{1}{\eta}\langle y_{t+1} - x_1, y_{t+1} - v_t\rangle \leqslant 0,\end{aligned}$$

which imply that $\langle x_t - y_{t+1}, v_t - y_{t+1}\rangle \leqslant 0$. Thus, since $x_{t+1} = (1-\sigma)x_t + \sigma v_t$, we get:

$$\begin{aligned}\|x_{t+1} - y_{t+1}\|^2 &= (1-\sigma)^2\|x_t - y_{t+1}\|^2 + 2\sigma(1-\sigma)\langle x_t - y_{t+1}, v_t - y_{t+1}\rangle + \sigma^2\|v_t - y_{t+1}\|^2 \\ &\leqslant (1-\sigma)^2\|x_t - y_{t+1}\|^2 - 2\sigma^2\langle x_t - y_{t+1}, v_t - y_{t+1}\rangle + \sigma^2\|v_t - y_{t+1}\|^2 \\ &= (1-2\sigma)\|x_t - y_{t+1}\|^2 + \sigma^2\|x_t - v_t\|^2.\end{aligned}$$

Combining this inequality with Equation (10), we get:

$$\phi_t - \phi_{t-1} \leqslant \frac{\sigma^2}{6\eta}\|x_t - v_t\|^2 + \langle g_t, x_t - y_{t+1}\rangle - \frac{1}{2\eta}\|y_t - y_{t+1}\|^2 + \frac{1}{6\eta}((1-2\sigma)\|x_t - y_{t+1}\|^2 - \|x_t - y_t\|^2). \tag{11}$$

[1] There is a typo in the proof of Hazan [Hazan, 2016, Theorem 7.3]. The proof is written for bounding $F_{t+1}(x_t) - F_{t+1}(y_t)$ (indeed it seems more natural to compare x_t to y_t as OFW can be seen as an approximation of FTRL), but after correction it indeed needs to be $F_{t+1}(x_t) - F_{t+1}(y_{t+1})$.

Let $\lambda \geqslant 0$ whose value will be chosen later. From the optimality in the definitions of y_t and y_{t+1}, we have:

$$\langle G_{t-1}, y_t - y_{t+1}\rangle + \frac{1}{\eta}\langle y_t - x_1, y_t - y_{t+1}\rangle \leqslant 0$$
$$\text{and}\quad \langle G_t, y_{t+1} - y_t\rangle + \frac{1}{\eta}\langle y_{t+1} - x_1, y_{t+1} - y_t\rangle \leqslant 0,$$

which imply that $\langle g_t, y_{t+1} - y_t\rangle \leqslant -\frac{1}{\eta}\|y_t - y_{t+1}\|^2$. From this inequality, we obtain:

$$\begin{aligned}\langle g_t, x_t - y_{t+1}\rangle &= \langle g_t, x_t - y_{t+1} + \lambda(y_t - y_{t+1})\rangle + \lambda\langle g_t, y_{t+1} - y_t\rangle\\ &\leqslant \langle g_t, x_t - y_{t+1} + \lambda(y_t - y_{t+1})\rangle - \frac{\lambda}{\eta}\|y_t - y_{t+1}\|^2\\ &\leqslant \lambda^g\|g_t\|^2 + \frac{1}{4\lambda^g}\|x_t - y_{t+1} + \lambda(y_t - y_{t+1})\|^2 - \frac{\lambda}{\eta}\|y_t - y_{t+1}\|^2,\end{aligned}$$

for any $\lambda^g > 0$ to be chosen latter. Combining this inequality with Equation (11), we get:

$$\begin{aligned}\phi_t - \phi_{t-1} \leqslant \frac{\sigma^2}{6\eta}\|x_t - v_t\|^2 + \lambda^g\|g_t\|^2 + \frac{1}{4\lambda^g}\|x_t - y_{t+1} + \lambda(y_t - y_{t+1})\|^2\\ - \frac{1+2\lambda}{2\eta}\|y_t - y_{t+1}\|^2 + \frac{1}{6\eta}((1-2\sigma)\|x_t - y_{t+1}\|^2 - \|x_t - y_t\|^2).\end{aligned}$$

Hence, to complete the proof for the upper bound on $\phi_t - \phi_{t-1}$, it suffices to show that there exists $\lambda \geqslant 0$ such that the following expression is non-negative, which can be done by rewriting it as a sum of squares:

$$\begin{aligned}Q(\lambda) \triangleq\ & \frac{1+2\lambda}{2\eta}\|y_t - y_{t+1}\|^2 - \frac{1}{4\lambda^g}\|x_t - y_{t+1} + \lambda(y_t - y_{t+1})\|^2\\ & - \frac{1}{6\eta}((1-2\sigma)\|x_t - y_{t+1}\|^2 - \|x_t - y_t\|^2).\\ =\ & \frac{2+3\lambda}{3\eta}\|y_t - y_{t+1}\|^2 + \frac{\sigma}{3\eta}\|x_t - y_{t+1}\|^2 + \frac{1}{3\eta}\langle x_t - y_{t+1}, y_{t+1} - y_t\rangle\\ & - \frac{1}{4\lambda^g}\|x_t - y_{t+1} + \lambda(y_t - y_{t+1})\|^2\end{aligned}$$

Indeed, substituting our choices of $\eta = \frac{D3^{3/4}}{2LT^{3/4}}$, $\sigma = \min(1, \frac{\sqrt{3}}{\sqrt{T}})$ and $\lambda^g = \frac{\sigma^2 D^2}{3\eta L^2}$, whose derivations are detailed in Section 3, and defining $\tilde{T} = T/3$ for conciseness, we obtain:

$$\begin{aligned}\frac{D}{L}Q(\lambda) =\ & \frac{4+6\lambda}{3}\tilde{T}^{3/4}\|y_t - y_{t+1}\|^2 + \frac{2\tilde{T}^{1/4}}{3}\|x_t - y_{t+1}\|^2 + \frac{2\tilde{T}^{3/4}}{3}\langle x_t - y_{t+1}, y_{t+1} - y_t\rangle\\ & - \frac{3\tilde{T}^{1/4}}{8}\|x_t - y_{t+1} + \lambda(y_t - y_{t+1})\|^2\\ =\ & \Big(\frac{4+6\lambda}{3}\tilde{T}^{3/4} - \frac{3\tilde{T}^{1/4}}{8}\lambda^2\Big)\|y_t - y_{t+1}\|^2 + \frac{7\tilde{T}^{1/4}}{24}\|x_t - y_{t+1}\|^2\\ & + 2\times\Big(\frac{\tilde{T}^{3/4}}{3} + \frac{3\tilde{T}^{1/4}}{8}\lambda\Big)\langle x_t - y_{t+1}, y_{t+1} - y_t\rangle.\end{aligned}$$

Thus, using a Schur complement argument, $Q(\lambda)$ is a sum of squares if and only if:

$$\Big(\frac{1}{3}\tilde{T}^{3/4} + \frac{3}{8}\tilde{T}^{1/4}\lambda\Big)^2 \leqslant \frac{7}{24}\tilde{T}^{1/4}\Big(\frac{4+6\lambda}{3}\tilde{T}^{3/4} - \frac{3}{8}\tilde{T}^{1/4}\lambda^2\Big),$$

which is equivalent to:

$$\frac{\sqrt{\tilde{T}}}{4}\lambda^2 - \frac{\tilde{T}}{3}\lambda + \Big(\frac{\tilde{T}^{3/2}}{9} - \frac{7\tilde{T}}{18}\Big) \leqslant 0.$$

Hence, there exists a non-negative λ solving this second order equation if and only if $\Delta \geqslant 0$, where:

$$\Delta \triangleq \frac{\tilde{T}^2}{9} - 4\frac{\sqrt{\tilde{T}}}{4}\Big(\frac{\tilde{T}^{3/2}}{9} - \frac{7\tilde{T}}{18}\Big) = \frac{7\tilde{T}^{3/2}}{18} \geqslant 0.$$

As a consequence, there exists a choice of $\lambda \geqslant 0$ such that $Q(\lambda)$ is a sum of squares, which concludes the proof of the upper bound on $\phi_t - \phi_{t-1}$, and thus also concludes the proof of Lemma 3.2. $\square$

A.2 Potential-based proof for FTRL

We present here the potential-based proof of the optimal upper bound on the regret of FTRL (see Algorithm 2) using the potential ψ_t defined in (5). This potential-based proof is fundamentally a reformulation of classical FTRL regret proofs (see , e.g., [Orabona, 2019, Chapters 6–7]). As the proof for the single (horizon-dependent) parameter η version of FTRL immediately generalizes to the round-dependent parameters $\{\eta_t\}_{t\in[\![2,T]\!]}$ version of FTRL, we prove the result for the latter version.

We first define the anytime version of FTRL with round-dependent parameters, where the values of η_t can be different between the rounds.

Algorithm 4 Follow The Regularized Leader (FTRL)

Require: $T \geqslant 1$, $y_1 \in \mathcal{K}$, $\eta_t \geqslant 0$ for $t \in [\![2,T]\!]$
1: **for** $t = 1$ to T **do**
2: Play y_t, pay cost $\ell_t(y_t)$, and observe $g_t = \nabla \ell_t(y_t)$.
3: $y_{t+1} \leftarrow \arg\min_{y\in\mathcal{K}} \eta_{t+1}\langle \sum_{s=1}^t g_s, y\rangle + \frac{1}{2}\|y - x_1\|^2$
4: **end for**

We define the potential ψ_t for the anytime version of FTRL (which is the anytime version of the definition in (5)):

$$\psi_t = \sum_{s=1}^{t} \langle g_s, y_s - y_{t+1}\rangle - \frac{1}{2\eta_{t+1}}\|y_{t+1} - y_1\|^2. \tag{12}$$

We can now state and prove the regret upper bound for the anytime version of FTRL with non-increasing parameters $\{\eta_t\}_{t\in[\![2,T]\!]}$. This gives the optimal regret upper bound in the single-parameter case (i.e., $\eta_t = \eta$ for all t).

Lemma A.1. *Let $T \geqslant 1$. Assume that the cost functions ℓ_t are convex and L-Lipschitz for all $t \in [\![1,T]\!]$, and that the convex closed domain $\mathcal{K}$ of feasible points has a diameter bounded by D. Assume that the FTRL parameters $\{\eta_t\}_{t\in[\![2,T]\!]}$ are non-increasing in t, and set $\eta_{T+1} = \eta_T$. Then, for any $y_\star \in \mathcal{K}$, the following upper bound on the regret of the FTRL Algorithm 4 holds:*

$$R_T \triangleq \sum_{t=1}^{T} \ell_t(y_t) - \ell_t(y_\star) \leqslant \frac{1}{2}\sum_{t=1}^{T} \eta_t \|g_t\|^2 + \frac{1}{2\eta_T}\|y_\star - y_1\|^2.$$

In particular,

(i) for $\eta = \frac{D}{L\sqrt{T}}$ we get that $R_T \leqslant LD\sqrt{T}$,

(ii) for $\eta_t = \frac{D}{L\sqrt{t}}$ we get that $R_T \leqslant \frac{3}{2}LD\sqrt{T}$,

(iii) for $\eta_t = \frac{D}{L\sqrt{2t}}$ we get that $R_T \leqslant \sqrt{2}LD\sqrt{T}$.

Note that we can impose $\eta_T = \eta_{T+1}$ as η_{T+1} is only used for computing y_{T+1} which comes after the horizon T and is thus not played and only used as a virtual point in the analysis. In particular, as η_{T+1} are y_{T+1} is a virtual value only used in the analysis, this does not change the anytime property of the tunings (ii) and (iii). This "trick" allows us to obtain a slightly tighter and nicer-looking regret upper bound: indeed, for $\eta_t = \Theta(\frac{1}{\sqrt{t}})$, this allows us to save an extra factor of $O(\frac{1}{\sqrt{T}})$ and does not change the leading order term in $\Theta(\sqrt{T})$.

Proof. We present a potential-based proof based on the potential ψ_t defined in (12).

We start by upper bounding $\psi_t - \psi_{t-1}$ for fixed $t \in [\![1,T]\!]$. Using the optimality in the definition of y_t gives:

$$\langle G_{t-1}, y_t - y_{t+1}\rangle + \frac{1}{\eta_t}\langle y_t - y_1, y_t - y_{t+1}\rangle \leqslant 0.$$

Then, using this equation and the inequality $\langle u, v\rangle \leqslant \frac{\eta_t}{2}\|u\|^2 + \frac{1}{2\eta_t}\|v\|^2$, we get the upper bound:

$$\begin{aligned}
\psi_t - \psi_{t-1} &= \langle G_t, y_t - y_{t+1}\rangle + \frac{1}{2\eta_t}\|y_t - y_1\|^2 - \frac{1}{2\eta_{t+1}}\|y_{t+1} - y_1\|^2 \\
&\leqslant \langle g_t, y_t - y_{t+1}\rangle + \frac{1}{2\eta_t}\left(\|y_t - y_1\|^2 + 2\langle y_t - y_1, y_{t+1} - y_t\rangle - \|y_{t+1} - y_1\|^2\right) \\
&\qquad - \left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t}\right)\|y_{t+1} - y_1\|^2 \\
&= \langle g_t, y_t - y_{t+1}\rangle - \frac{1}{2\eta_t}\|y_t - y_{t+1}\|^2 - \left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t}\right)\|y_{t+1} - y_1\|^2 \\
&\leqslant \frac{\eta_t}{2}\|g_t\|^2.
\end{aligned}$$

Now, combining those potential differences, the sum telescopes (remark that $\psi_0 = 0$) and we get:

$$\psi_T = \sum_{t=1}^{T} \psi_t - \psi_{t-1} \leqslant \frac{1}{2}\sum_{t=1}^{T}\eta_t\|g_t\|^2.$$

Using the optimality in the definition of y_{t+1} gives:

$$R_T - \frac{1}{2\eta_{T+1}}\|y_\star - y_1\|^2 \leqslant \psi_T,$$

which concludes the proof of the main upper bound of R_T (recall that $\eta_T = \eta_{T+1}$). The second statement follows directly by using the Lipschitz and diameter bounds. □

A.3 Detailed proof of Theorem 3.3

The proof of Theorem 3.3 is an adaptation of that of Theorem 3.1. Thus, to prove Theorem 3.3, we start by proving the following lemma upper bounding the potential increase $\phi_t - \phi_{t-1}$, which is an adaptation of Lemma 3.2.

Lemma A.2. *Denote by $\ell_t : x \mapsto \langle g_t, x\rangle$ the linear cost function at time $t \geqslant 1$. Assume that $\|g_t\| \leqslant L$ for all $t \geqslant 1$ and that the convex closed domain $\mathcal{K}$ of feasible points has a diameter bounded by D. Define the sequence of potentials $(\phi_t)_{t\geqslant 0}$ as:*

$$\phi_t \triangleq \sum_{s=1}^{t}\langle g_s, x_s - y_{t+1}\rangle - \frac{1}{2\eta_{t+1}}\|y_{t+1} - x_1\|^2 + \frac{1}{6\eta_{t+1}}\|x_{t+1} - y_{t+1}\|^2.$$

Then, Algorithm 3 and 4, run with parameters in (6) *and initialized in $y_1 = x_1 \in \mathcal{K}$, satisfy,*

$$\phi_t - \phi_{t-1} \leqslant \frac{2D}{L3^{3/4}t^{1/4}}\|g_t\|^2 + \frac{L}{D3^{3/4}(t+1)^{1/4}}\|x_t - v_t\|^2\,, \qquad \forall t \geqslant 2\,.$$

The proof of Lemma A.2 is an adaptation of that of Lemma 3.2 where the difference between η_{t+1} and η_t creates extra error terms.

Proof of Lemma A.2. Let $t \geqslant 2$ be fixed. We are going to prove the upper bound on $\phi_t - \phi_{t-1}$. Denote by $G_{t-1} = \sum_{s=1}^{t-1} g_s$ the sum of past gradients, so that $G_t = G_{t-1} + g_t$. Note that we have:

$$\begin{aligned}
\phi_t - \phi_{t-1} &= \langle g_t, x_t - y_{t+1}\rangle + \langle G_{t-1}, y_t - y_{t+1}\rangle \\
&\quad + \frac{1}{6\eta_{t+1}}\|x_{t+1} - y_{t+1}\|^2 - \frac{1}{6\eta_t}\|x_t - y_t\|^2 + \frac{1}{2\eta_t}\|y_t - x_1\|^2 - \frac{1}{2\eta_{t+1}}\|y_{t+1} - x_1\|^2.
\end{aligned}$$

Using the optimality in the definition of y_t, we have that $\langle G_{t-1}, y_t - y_{t+1}\rangle \leqslant \frac{1}{\eta_t}\langle y_t - x_1, y_{t+1} - y_t\rangle$, which gives us:

$$\begin{aligned}
\phi_t - \phi_{t-1} &\leqslant \langle g_t, x_t - y_{t+1}\rangle - \frac{1}{2\eta_t}\|y_t - y_{t+1}\|^2 - \frac{1}{2}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\|y_{t+1} - x_1\|^2 \\
&\quad + \frac{1}{6\eta_{t+1}}\|x_{t+1} - y_{t+1}\|^2 - \frac{1}{6\eta_t}\|x_t - y_t\|^2. \qquad (13)
\end{aligned}$$

From the optimality in the definitions of v_t and y_{t+1}, we have:

$$\langle G_t, v_t - y_{t+1}\rangle + \frac{1}{\eta_{t+1}}\langle x_t - x_1, v_t - y_{t+1}\rangle \leqslant 0$$
$$\text{and}\quad \langle G_t, y_{t+1} - v_t\rangle + \frac{1}{\eta_{t+1}}\langle y_{t+1} - x_1, y_{t+1} - v_t\rangle \leqslant 0,$$

which imply that $\langle x_t - y_{t+1}, v_t - y_{t+1}\rangle \leqslant 0$. Thus, since $x_{t+1} = (1-\sigma_{t+1})x_t + \sigma_{t+1}v_t$, we get:

$$\begin{aligned}\|x_{t+1} - y_{t+1}\|^2 &= (1-\sigma_{t+1})^2\|x_t - y_{t+1}\|^2 + 2\sigma_{t+1}(1-\sigma_{t+1})\langle x_t - y_{t+1}, v_t - y_{t+1}\rangle + \sigma_{t+1}^2\|v_t - y_{t+1}\|^2 \\ &\leqslant (1-\sigma_{t+1})^2\|x_t - y_{t+1}\|^2 - 2\sigma_{t+1}^2\langle x_t - y_{t+1}, v_t - y_{t+1}\rangle + \sigma_{t+1}^2\|v_t - y_{t+1}\|^2 \\ &= (1-2\sigma_{t+1})\|x_t - y_{t+1}\|^2 + \sigma_{t+1}^2\|x_t - v_t\|^2.\end{aligned}$$

Combining this inequality with Equation (13), we get:

$$\begin{aligned}\phi_t - \phi_{t-1} \leqslant{}& \frac{\sigma_{t+1}^2}{6\eta_{t+1}}\|x_t - v_t\|^2 + \langle g_t, x_t - y_{t+1}\rangle - \frac{1}{2\eta_t}\|y_t - y_{t+1}\|^2 - \frac{1}{2}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\|y_{t+1} - x_1\|^2 \\ &+ \frac{1}{6\eta_{t+1}}(1-2\sigma_{t+1})\|x_t - y_{t+1}\|^2 - \frac{1}{6\eta_t}\|x_t - y_t\|^2. \qquad (14)\end{aligned}$$

Let $\lambda \geqslant 0$ whose value will be chosen later. From the optimality in the definitions of y_t and y_{t+1}, we have:

$$\langle G_{t-1}, y_t - y_{t+1}\rangle + \frac{1}{\eta_t}\langle y_t - x_1, y_t - y_{t+1}\rangle \leqslant 0$$
$$\text{and}\quad \langle G_t, y_{t+1} - y_t\rangle + \frac{1}{\eta_{t+1}}\langle y_{t+1} - x_1, y_{t+1} - y_t\rangle \leqslant 0,$$

which imply that $\langle g_t, y_{t+1} - y_t\rangle \leqslant -\frac{1}{\eta_t}\|y_t - y_{t+1}\|^2 - (\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t})\langle y_{t+1} - x_1, y_{t+1} - y_t\rangle$. From this inequality, we obtain:

$$\begin{aligned}\langle g_t, x_t - y_{t+1}\rangle &= \langle g_t, x_t - y_{t+1} + \lambda(y_t - y_{t+1})\rangle + \lambda\langle g_t, y_{t+1} - y_t\rangle \\ &\leqslant \langle g_t, x_t - y_{t+1} + \lambda(y_t - y_{t+1})\rangle - \frac{\lambda}{\eta_t}\|y_t - y_{t+1}\|^2 \\ &\quad - \lambda\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\langle y_{t+1} - x_1, y_{t+1} - y_t\rangle \\ &\leqslant \lambda^g\|g_t\|^2 + \frac{1}{4\lambda^g}\|x_t - y_{t+1} + \lambda(y_t - y_{t+1})\|^2 - \frac{\lambda}{\eta_t}\|y_t - y_{t+1}\|^2 \\ &\quad - \lambda\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\langle y_{t+1} - x_1, y_{t+1} - y_t\rangle,\end{aligned}$$

for any $\lambda^g > 0$ to be chosen latter. Combining this inequality with Equation (14), we get:

$$\begin{aligned}\phi_t - \phi_{t-1} \leqslant{}& \frac{\sigma_{t+1}^2}{6\eta_{t+1}}\|x_t - v_t\|^2 + \lambda^g\|g_t\|^2 + \frac{1}{4\lambda^g}\|x_t - y_{t+1} + \lambda(y_t - y_{t+1})\|^2 \\ &- \frac{1}{2}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\|y_{t+1} - x_1\|^2 - \lambda\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\langle y_{t+1} - x_1, y_{t+1} - y_t\rangle \\ &- \frac{1+2\lambda}{2\eta_t}\|y_t - y_{t+1}\|^2 + \frac{1}{6\eta_{t+1}}(1-2\sigma_{t+1})\|x_t - y_{t+1}\|^2 - \frac{1}{6\eta_t}\|x_t - y_t\|^2.\end{aligned}$$

Recombining terms around the scalar product term, this upper bound can be rewritten as:

$$\begin{aligned}\phi_t - \phi_{t-1} \leqslant{}& \frac{\sigma_{t+1}^2}{6\eta_{t+1}}\|x_t - v_t\|^2 + \lambda^g\|g_t\|^2 - \frac{1}{2}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\|y_{t+1} - x_1 + \lambda(y_{t+1} - y_t)\|^2 \\ &+ \frac{1}{4\lambda^g}\|x_t - y_{t+1} + \lambda(y_t - y_{t+1})\|^2 + \frac{\lambda^2}{2}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\|y_t - y_{t+1}\|^2 \\ &- \frac{1+2\lambda}{2\eta_t}\|y_t - y_{t+1}\|^2 + \frac{1}{6\eta_{t+1}}(1-2\sigma_{t+1})\|x_t - y_{t+1}\|^2 - \frac{1}{6\eta_t}\|x_t - y_t\|^2.\end{aligned}$$

Using the definition of $\eta_t = \frac{D3^{3/4}}{2Lt^{3/4}}$ and that $(1+x)^{3/4} \leqslant 1 + \frac{3}{4}x$ for $x \geqslant 0$, we get that

$$0 \leqslant \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right) \leqslant \frac{3}{4t\eta_t}.$$

Combining those two inequalities, we get:

$$\begin{aligned}\phi_t - \phi_{t-1} \leqslant{}& \frac{\sigma_{t+1}^2}{6\eta_{t+1}}\|x_t - v_t\|^2 + \lambda^g\|g_t\|^2 \\ &+ \frac{1}{4\lambda^g}\|x_t - y_{t+1} + \lambda(y_t - y_{t+1})\|^2 - \left(\frac{1+2\lambda}{2\eta_t} - \frac{3\lambda^2}{8t\eta_t}\right)\|y_t - y_{t+1}\|^2 \\ &+ \frac{1}{6\eta_{t+1}}(1 - 2\sigma_{t+1})\|x_t - y_{t+1}\|^2 - \frac{1}{6\eta_t}\|x_t - y_t\|^2.\end{aligned}$$

Hence, to complete the proof for the upper bound on $\phi_t - \phi_{t-1}$, it suffices to show that there exists $\lambda > 0$ such that the following expression is non-negative:

$$\begin{aligned}Q(\lambda) \triangleq{}& \left(\frac{1+2\lambda}{2\eta_t} - \frac{3\lambda^2}{8t\eta_t}\right)\|y_t - y_{t+1}\|^2 - \frac{1}{4\lambda^g}\|x_t - y_{t+1} + \lambda(y_t - y_{t+1})\|^2 \\ &- \frac{1}{6\eta_{t+1}}(1 - 2\sigma_{t+1})\|x_t - y_{t+1}\|^2 + \frac{1}{6\eta_t}\|x_t - y_t\|^2 \\ ={}& \left(\frac{2+3\lambda}{3\eta_t} - \frac{3\lambda^2}{8t\eta_t}\right)\|y_t - y_{t+1}\|^2 + \frac{1}{3\eta_t}\langle x_t - y_{t+1}, y_{t+1} - y_t\rangle \\ &+ \left(\frac{\sigma_{t+1}}{3\eta_{t+1}} - \frac{1}{6}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\right)\|x_t - y_{t+1}\|^2 - \frac{1}{4\lambda^g}\|x_t - y_{t+1} + \lambda(y_t - y_{t+1})\|^2 \\ \geqslant{}& \tilde{Q}(\lambda),\end{aligned}$$

where

$$\begin{aligned}\tilde{Q}(\lambda) \triangleq{}& \left(\frac{2+3\lambda}{3\eta_t} - \frac{3\lambda^2}{8t\eta_t}\right)\|y_t - y_{t+1}\|^2 + \frac{1}{3\eta_t}\langle x_t - y_{t+1}, y_{t+1} - y_t\rangle \\ &+ \left(\frac{\sigma_{t+1}}{3\eta_{t+1}} - \frac{1}{8t\eta_t}\right)\|x_t - y_{t+1}\|^2 - \frac{1}{4\lambda^g}\|x_t - y_{t+1} + \lambda(y_t - y_{t+1})\|^2 \\ ={}& \left(\frac{2+3\lambda}{3\eta_t} - \frac{3\lambda^2 t^{-1}}{8\eta_t} - \frac{\lambda^2}{4\lambda^g}\right)\|y_t - y_{t+1}\|^2 + 2\times\left(\frac{1}{6\eta_t} + \frac{\lambda}{4\lambda^g}\right)\langle x_t - y_{t+1}, y_{t+1} - y_t\rangle \\ &+ \left(\frac{\sigma_{t+1}}{3\eta_{t+1}} - \frac{1}{8t\eta_t} - \frac{1}{4\lambda^g}\right)\|x_t - y_{t+1}\|^2.\end{aligned}$$

Hence, to complete the proof for the upper bound on $\phi_t - \phi_{t-1}$, it suffices to show that there exists $\lambda \geqslant 0$ such that $\tilde{Q}(\lambda)$ is non-negative, which can be done by rewriting it as a sum of squares. Using a Schur complement argument, we get that $\tilde{Q}(\lambda)$ is a sum of square if and only if:

$$\frac{\sigma_{t+1}}{3\eta_{t+1}} - \frac{1}{4\lambda^g} - \frac{1}{8t\eta_t} \geqslant 0 \tag{15}$$

and

$$\left(\frac{1}{6\eta_t} + \frac{\lambda}{4\lambda^g}\right)^2 \leqslant \left(\frac{\sigma_{t+1}}{3\eta_{t+1}} - \frac{1}{4\lambda^g} - \frac{1}{8t\eta_t}\right)\left(\frac{2+3\lambda}{3\eta_t} - \frac{\lambda^2}{4\lambda^g} - \frac{3\lambda^2}{8t\eta_t}\right). \tag{16}$$

We now substitute our choices of $\eta_t = \frac{D3^{3/4}}{2Lt^{3/4}}$, $\sigma_t = \min(1, \frac{\sqrt{3}}{\sqrt{t}})$ and $\lambda^g = \frac{\sigma_t^2 D^2}{3\eta_t L^2}$, whose derivations for the fixed parameter setting (i.e., non-anytime setting) are detailed in Section 3 (note that those parameter choices are still leading-term optimal in the anytime setting). In particular, note that, as $t \geqslant 2$, we have that $\frac{\sigma_{t+1}}{\eta_{t+1}} \geqslant \frac{2L}{D}\left(\frac{t}{3}\right)^{1/4}$. Thus, defining $\tilde{t} = t/3$ for conciseness, we obtain for $t \geqslant 1$ that $\tilde{Q}(\lambda) \geqslant 0$ is a sum of square if:

$$\frac{7\tilde{t}^{1/2}}{24} \geqslant \frac{1}{12\tilde{t}^{1/4}} \tag{17}$$

and

$$\left(\frac{\tilde{t}^{3/4}}{3} + \frac{3\tilde{t}^{1/4}\lambda}{8}\right)^2 \leqslant \left(\frac{7\tilde{t}^{1/4}}{24} - \frac{1}{12\tilde{t}^{1/4}}\right)\left(\frac{4\tilde{t}^{3/4}}{3} + 2\tilde{t}^{3/4}\lambda - \frac{3\tilde{t}^{1/4}\lambda^2}{8} - \frac{\lambda^2}{4\tilde{t}^{1/4}}\right). \tag{18}$$

The first line is always satisfied for $t \geqslant 1$. The second line is equivalent to:

$$\left(\frac{\sqrt{\tilde{t}}}{4} + \frac{1}{24} - \frac{1}{48\sqrt{\tilde{t}}}\right)\lambda^2 + \left(-\frac{\tilde{t}}{3} + \frac{\sqrt{\tilde{t}}}{6}\right)\lambda + \left(\frac{\tilde{t}^{3/2}}{9} - \frac{7\tilde{t}}{18} + \frac{\sqrt{\tilde{t}}}{9}\right) \leqslant 0. \tag{19}$$

Note that $-\frac{\tilde{t}}{3}+\frac{\sqrt{\tilde{t}}}{6}\leqslant 0$ is equivalent to $t\geqslant 3/4$, which is always true. Also note that $\frac{\sqrt{\tilde{t}}}{4}-\frac{10}{96}+\frac{1}{48\sqrt{\tilde{t}}}\geqslant 0$ is true for any $t\geqslant 1$. Hence, there exists a non-negative λ that solves (19) if and only if $\Delta(\tilde{t})\geqslant 0$ where:

$$\Delta(\tilde{t})\triangleq\left(-\frac{\tilde{t}}{3}+\frac{\sqrt{\tilde{t}}}{6}\right)^2-4\left(\frac{\sqrt{\tilde{t}}}{4}+\frac{1}{24}-\frac{1}{48\sqrt{\tilde{t}}}\right)\left(\frac{\tilde{t}^{3/2}}{9}-\frac{7\tilde{t}}{18}+\frac{\sqrt{\tilde{t}}}{9}\right). \tag{20}$$

Indeed, we have:

$$\Delta(\tilde{t})=\frac{7\tilde{t}^{3/2}}{27}-\frac{\tilde{t}}{108}-\frac{11\sqrt{\tilde{t}}}{216}+\frac{1}{108}\geqslant\Delta(1/3)>0. \tag{21}$$

As a consequence, there exists a choice of $\lambda\geqslant 0$ such that $\tilde{Q}(\lambda)$ is a sum of squares, which concludes the proof of the upper bound on $\phi_t-\phi_{t-1}$ for $t\geqslant 2$, and thus also concludes the proof of Lemma A.2. □

Proof of Theorem 3.3. Let $t\geqslant 1$ be fixed. We prove the regret bound on R_t in the theorem for this value of t. Through this proof, we will denote by s some time index ranging from 1 to t. First, using convexity of the cost functions ℓ_s (recall $g_s=\nabla\ell_s(x_s)$), we get:

$$R_t\leqslant\sum_{s=1}^{t}\langle g_s,x_s-x_\star\rangle.$$

We upper bound $\phi_s-\phi_{s-1}$ by applying Lemma A.2 for $s\in[\![2,t]\!]$ to the linearized cost functions $\tilde{\ell}_s:x\mapsto\langle g_s,x\rangle$, and then summing those inequalities, we get

$$\phi_t-\phi_1\leqslant\frac{2D}{L3^{3/4}}\sum_{s=2}^{t}\frac{1}{s^{1/4}}\|g_s\|^2+\frac{L}{D3^{3/4}}\sum_{s=2}^{t}\frac{1}{(s+1)^{3/4}}\|x_s-v_s\|^2.$$

Throughout the proof, we will repeatedly use the fact that the optimum of a constrained convex optimization problem $x_*\in\arg\min_{x\in\mathcal{K}}f(x)$ satisfies $\langle\nabla f(x_*),x_*-x\rangle\leqslant 0$ for all $x\in\mathcal{K}$. Using the optimality from the definition of y_{t+1}, we get:

$$\sum_{s=1}^{t}\langle g_s,x_s-x_\star\rangle-\frac{1}{2\eta_{t+1}}\|x_\star-x_1\|^2\leqslant\sum_{s=1}^{t}\langle g_s,x_s-y_{t+1}\rangle-\frac{1}{2\eta_{t+1}}\|y_{t+1}-x_1\|^2\leqslant\phi_t.$$

Combining those four inequalities (recall $\eta_{t+1}=\frac{3^{3/4}D}{2L(t+1)^{3/4}}$), we get:

$$R_t\leqslant\frac{2D}{L3^{3/4}}\sum_{s=2}^{t}\frac{1}{s^{1/4}}\|g_s\|^2+\frac{L}{D3^{3/4}}\sum_{s=2}^{t}\frac{1}{(s+1)^{1/4}}\|x_s-v_s\|^2+\frac{L(t+1)^{3/4}}{D3^{3/4}}\|x_\star-x_1\|^2+\phi_1.$$

Using Cauchy-Schwarz inequality, and the bounds on the gradients and on the diameter, we get $\phi_1\leqslant(1+\frac{1}{6\eta_2})LD\leqslant 1.12LD$. Moreover, as $\|g_s\|\leqslant L$ and $\|x_s-v_s\|\leqslant D$ for all $s\in[\![1,t]\!]$ and $\|x_\star-x_1\|\leqslant D$, we get:

$$R_t\leqslant\frac{2LD}{3^{3/4}}\sum_{s=2}^{t}\frac{1}{s^{1/4}}+\frac{LD}{3^{3/4}}\sum_{s=2}^{t}\frac{1}{(s+1)^{1/4}}+\frac{LD(t+1)^{3/4}}{3^{3/4}}+1.12LD. \tag{22}$$

Using series-integral comparison, we get for any integers $1\leqslant a\leqslant b$:

$$\sum_{t=a}^{b}\frac{1}{t^{1/4}}\leqslant\int_{a-1}^{b}u^{-1/4}\,\mathrm{d}u=\frac{4}{3}(b^{3/4}-(a-1)^{3/4}).$$

Combining those inequalities, and using that $(1+x)^{3/4}\leqslant 1+\frac{3}{4}x$ for $x\geqslant 0$, we get:

$$\begin{aligned}R_t&\leqslant\frac{2LD}{3^{3/4}}\frac{4}{3}(t^{3/4}-1)+\frac{LD}{3^{3/4}}\frac{4}{3}((t+1)^{3/4}-2^{3/4})+\frac{LD(t+1)^{3/4}}{3^{3/4}}+1.12LD\\&\leqslant\frac{5}{3^{3/4}}LDt^{3/4}+\left(1.12-\frac{8}{3}\frac{1}{3^{3/4}}-\frac{4}{3}\left(\frac{2}{3}\right)^{3/4}\right)LD+\frac{LD}{3^{3/4}}\frac{1}{t^{1/4}}\\&\leqslant\frac{5}{3^{3/4}}LDt^{3/4}-1.03LD+\frac{LD}{3^{3/4}}\frac{1}{t^{1/4}}\\&\leqslant\frac{5}{3^{3/4}}LDt^{3/4},\end{aligned}$$

where in the last inequality we use that $t\geqslant 1$. This being true for all values of $t\geqslant 1$, this concludes the proof of Theorem 3.3. □

B Detailed semidefinite formulations

B.1 Tractable formulation of (2)

As explained in Section 2.2, (2) is a priori an infinite-dimensional problem, as it includes functional variables which are the losses ℓ_t and the indicator function of the feasible set $\mathcal{K}$. However, (2) can be reformulated as finite dimensional linear semidefinite program. In this section, we detail how to obtain this reformulation.

The first step consists in reformulating (2) as a finite-dimensional problem by sampling the losses ℓ_t at the query points x_t and $x_\star$, and to treat only the responses $(\nabla \ell_t(x_t), \ell_t(x_t))$ and $(\nabla \ell_t(x_\star), \ell_t(x_\star))$ as variables. By appropriately constraining these responses, we can force them to be compatible with some losses ℓ_t satisfying the desired assumptions (convexity and Lipschitzness of ℓ_t). Using an interpolation / extension theorem [Taylor et al., 2017, Theorem 3.3 and Equation (7)], it suffices to require that the samples are compatible through the subgradient inequalities $\ell_t(x_t) - \ell_t(x_\star) \leqslant \langle \nabla \ell_t(x_t), x_t - x_\star \rangle$ and $\ell_t(x_\star) - \ell_t(x_t) \leqslant \langle \nabla \ell_t(x_\star), x_\star - x_t \rangle$ and the Lipschitz bounds $\|\nabla \ell_t(x_t)\| \leqslant L$ for $t = 1, \cdots, T$. Indeed, as the subgradients $\nabla \ell_t(x_\star)$ are never used in (2) their values do not matter, and thus we can impose $\nabla \ell_t(x_\star) = \nabla \ell_t(x_t)$ without changing the value of the problem (that is, we get a tight reformulation of (2) and not just an upper bound). This allows us to assume without loss of generality that the cost functions ℓ_t are all linear. This corresponds to upper bound $\ell_t(x_t) - \ell_t(x_\star)$ by $\langle g_t, x_t - x_\star \rangle$ (recall $g_t = \nabla \ell_t(x_t)$) in the objective of (2), and replace the cost functions variables ℓ_t by the gradient variables g_t. Note that we get a tight reformulation of (2) and not just an upper bound as any point (with linear cost functions) in this new reformulation is still feasible with the same objective value in the initial problem (2).

As for handling the convex domain $\mathcal{K}$, one possible approach is to sample its indicator function at the query points v_t and x_t for $t = 1, \ldots, T$ and $x_\star$ and impose similar compatibility constraints. Indeed, denoting by $\iota_\mathcal{K}$ the indicator function of the feasible set $\mathcal{K}$, and using [Taylor et al., 2017, Theorem 3.6], those compatibility constraints are: for all $u, v \in \{x_1, \ldots, x_T, v_1, \ldots, v_{T-1}, x_\star\}$, and $g \in \nabla \iota_\mathcal{K}(u)$ a subgradient of $\iota_\mathcal{K}$ at u, we have $\langle g, v - u \rangle \leqslant 0$. As the points $\{(x_t, v_t, \mathrm{dir}_t)\}_{t=1,\ldots,T}$ are generated by (1), we get that $-\mathrm{dir}_t$ is a subgradient of $\iota_\mathcal{K}$ at v_t for $t = 1, \ldots, T-1$. However, we have no special choice for the subgradients of $\iota_\mathcal{K}$ at $x_1, \ldots, x_T, x_\star$, and we will choose their subgradients to be 0 (which leads to trivial constraints that we remove from the problem). Those two function sampling steps give us the following finite-dimensional reformulation of (2):

$$
\begin{aligned}
B_T(\{(\eta_{t,s}, \beta_{t,s}, \gamma_{t,s})\}_{t,s}) = \sup_{\substack{\{g_t\}_{t=1,\ldots,T},\, x_\star \\ \{(x_t, v_t, \mathrm{dir}_t)\}_{t=1,\ldots,T} \\ d \in \mathbb{N}}} \quad & \sum_{t=1}^{T} \langle g_t, x_t - x_\star \rangle \\
\text{subject to: } \quad & \|g_t\| \leqslant L \text{ for } t = 1, \ldots, T, \\
& \langle -\mathrm{dit}_t, u - v_t \rangle \leqslant 0 \text{ for } t = 1, \ldots, T-1 \\
& \qquad \text{and } u \in \{x_1, \ldots, x_T, v_1, \ldots, v_{T-1}, x_\star\}, \\
& \mathrm{Diam}(\{x_1, \ldots, x_T, v_1, \ldots, v_{T-1}, x_\star\}) \leqslant D, \\
& \{(x_t, v_t, \mathrm{dir}_t)\}_{t=1,\ldots,T} \text{ is generated by (1).}
\end{aligned}
$$

As $x_2, \cdots, x_T$ are convex combination of $x_1, v_1, \cdots, v_{T-1}$, we can omit them in the boundary and diameter constraints, leading to the following simpler finite-dimensional reformulation of (2):

$$
\begin{aligned}
B_T(\{(\eta_{t,s}, \beta_{t,s}, \gamma_{t,s})\}_{t,s}) = \sup_{\substack{\{g_t\}_{t=1,\ldots,T},\, x_\star \\ \{(x_t, v_t, \mathrm{dir}_t)\}_{t=1,\ldots,T} \\ d \in \mathbb{N}}} \quad & \sum_{t=1}^{T} \langle g_t, x_t - x_\star \rangle \\
\text{subject to: } \quad & \{(x_t, \mathrm{dir}_t)\}_{t=1,\ldots,T} \text{ compatible with (1),} \\
& \langle -\mathrm{dir}_t, u - v_t \rangle \leqslant 0 \text{ for all } t = 1, \ldots, T-1 \\
& \qquad \text{and } u \in \{x_1, v_1, \cdots, v_{T-1}, x_\star\}, \\
& \mathrm{Diam}(\{x_1, v_1, \cdots, v_{T-1}, x_\star\}) \leqslant D, \\
& \|g_t\| \leqslant L \text{ for } t = 1, \ldots, T.
\end{aligned}
\tag{23}
$$

Finally, this sampled version (23) of (2) can be lifted to a semidefinite program via a standard change of variables: all vectors and gradients appearing in (23) are replaced with their Gram matrix (which, recall, encodes all pairwise inner products between these vectors / gradients). As the problem is invariant by translation of the vectors x_t, v_t and $x_\star$ and of the feasible set $\mathcal{K}$, without loss of generality, we can assume that $x_1 = 0$. Let G be the Gram matrix of the gradients $g_1, \cdots, g_T$ and the vectors $v_1, \cdots, v_{T-1}$ and $x_\star$, that is, $G = P^{\mathrm{T}} P$ with $P = [\, g_1 \,|\, \cdots \,|\, g_T \,|\, v_1 \,|\, \cdots \,|\, v_{T-1} \,|\, x_\star \,]$.

Let $e_1, \ldots, e_{2T}$ be the standard basis vectors of $\mathbb{R}^{2T}$. Define $\bar{g}_t = e_t$ for $t = 1, \ldots, T$, $\bar{v}_t = e_{T+t}$ for $t = 1, \ldots, T-1$, $\bar{x}_\star = e_{2T}$ and $\bar{x}_1 = 0$. Thus, we get that $\bar{g}_s^{\mathrm{T}} G \bar{g}_t = \langle g_s, g_t \rangle$ for all $s, t = 1, \ldots, T$; and similarly with the other vectors $\bar{v}_t$ for $t = 1, \ldots, T-1$, $\bar{x}_\star$ and $\bar{x}_1$. For two vectors u and v in $\mathbb{R}^{2T}$, we define $u \odot v = (uv^{\mathrm{T}} + vu^{\mathrm{T}})/2$ their symmetric outer product. We denote by tr the trace operator for square matrices.

The objective, as well as all constraints, then become linear functions of the entries of the Gram matrix G. Indeed, note that for all $u, v \in \{x_1, v_1, \cdots, v_{T-1}, x_\star\}$ and $t \in [\![1, T]\!]$, we have:

$$
\begin{aligned}
&\|g_t\|^2 = \bar{g}_t^{\mathrm{T}} G \bar{g}_t = \operatorname{tr}(\bar{g}_t \odot \bar{g}_t)\, G), \\
&\|u - v\|^2 = \operatorname{tr}(((\bar{u} - \bar{v}) \odot (\bar{u} - \bar{v}))\, G), \\
&\langle g_t, x_t \rangle = \left\langle g_t, \sum_{s=1}^{t-1} \gamma_{t,s}\, v_s \right\rangle = \sum_{s=1}^{t-1} \operatorname{tr}((\bar{g}_t \odot \bar{v}_s)\, G), \\
&\langle g_t, x_* \rangle = \operatorname{tr}((\bar{g}_t \odot \bar{x}_\star)\, G)\,.
\end{aligned}
$$

This gives us the following linear (in G) semidefinite program reformulation of (23):

$$
\begin{aligned}
B_T(\{(\eta_{t,s}, \beta_{t,s}, \gamma_{t,s})\}_{t,s}) = \sup_{G \succcurlyeq 0} \; & \sum_{t=1}^{T} \sum_{s=1}^{t-1} \gamma_{t,s} \operatorname{tr}((\bar{g}_t \odot \bar{v}_s)\, G) - \sum_{t=1}^{T} \operatorname{tr}((\bar{g}_t \odot \bar{x}_\star)\, G) \\
\text{subject to: } & \bar{\mathrm{dir}}_t = \sum_{s=1}^{t} \eta_{t,s}\, \bar{g}_s + \sum_{s=1}^{t-1} \beta_{t,s}\, \bar{v}_s \text{ for } t = 1, \ldots, T-1, \\
& \operatorname{tr}((\bar{\mathrm{dir}}_t \odot (\bar{v}_t - u))\, G) \leqslant 0 \text{ for all } t = 1, \ldots, T-1 \\
& \qquad \text{and } u \in \{\bar{x}_1, \bar{v}_1, \cdots, \bar{v}_{T-1}, \bar{x}_\star\}, \\
& \operatorname{tr}(((u - v) \odot (u - v))\, G) \leqslant D^2 \\
& \qquad \text{for } u,\, v \in \{\bar{x}_1, \bar{v}_1, \cdots, \bar{v}_{T-1}, \bar{x}_\star\}, \\
& \operatorname{tr}((\bar{g}_t \odot \bar{g}_t)\, G) \leqslant L^2 \text{ for } t = 1, \ldots, T.
\end{aligned}
\tag{24}
$$

Note that the variable $d \in \mathbb{N}$ for the dimension of the feasible set $\mathcal{K}$ (and of all the gradients and vectors) that appears in (2) and (23) induces a constraint imposing that the rank of the matrix G is at most d. However, this constraint disappears when taking the supremum over all values of $d \in \mathbb{N}$.

As explained in Remark 2.1, for given numerical values of T, L, D and the algorithm parameters, solving the tractable convex problem (24) allows us to get worst-case examples giving *algorithm-dependent lower bounds* on the worst-case regret. In order to obtain *algorithm-dependent upper bounds* on the worst-case regret, a natural procedure consists in formulating the *Lagrange dual* of (2) (which is also a semidefinite program; see, e.g., [Vandenberghe and Boyd, 1996, Boyd and Vandenberghe, 2004]), whose feasible points naturally corresponds to upper bounds on the regret. In this context, *finding a proof* consists in finding a feasible point to the dual problem [Goujaud et al., 2023]. In particular, it is useful to reformulate (24) as its Lagrange dual which is the following semidefinite problem:

$$
\begin{aligned}
B_T(\{(\eta_{t,s},\beta_{t,s},\gamma_{t,s})\}_{t,s}) = &\inf_{\substack{\lambda^{\mathrm{Lip}}\geqslant 0\\ \lambda^{\mathrm{Diam}}\geqslant 0\\ \lambda^{\mathrm{Brd}}\geqslant 0}} \sum_{t=1}^{T} \lambda_t^{\mathrm{Lip}}\, L^2 + \frac{1}{2}\sum_{u,\, v\in\{\bar{x}_1,\bar{v}_1,\cdots,\bar{v}_{T-1},\bar{x}_\star\}} \lambda_{\{u,v\}}^{\mathrm{Diam}}\, D^2\\
&\text{subject to: } S(\eta,\beta,\gamma;\lambda)\succcurlyeq 0,\\
\text{where } S(\eta,\beta,\gamma;\lambda) = &\frac{1}{2}\sum_{u,\, v\in\{\bar{x}_1,\bar{v}_1,\cdots,\bar{v}_{T-1},\bar{x}_\star\}} \lambda_{\{u,v\}}^{\mathrm{Diam}}\,((u-v)\odot(u-v))\\
&+\sum_{t=1}^{T}\lambda_t^{\mathrm{Lip}}\,(\bar{g}_t\odot\bar{g}_t) + \sum_{t=1}^{T-1}\sum_{u\in\{\bar{x}_1,\bar{v}_1,\cdots,\bar{v}_{T-1},\bar{x}_\star\}} \lambda_{\bar{v}_t,u}^{\mathrm{Brd}}\,(\bar{\mathrm{dir}}_t\odot(\bar{v}_t-u))\\
&-\sum_{t=1}^{T}\sum_{s=1}^{t-1}\gamma_{t,s}\,(\bar{g}_t\odot\bar{v}_s) + \sum_{t=1}^{T}(\bar{g}_t\odot\bar{x}_\star),\\
\text{and } \lambda = (\lambda^{\mathrm{Lip}},\lambda^{\mathrm{Diam}},\lambda^{\mathrm{Brd}}) &\text{ and } \bar{\mathrm{dir}}_t = \sum_{s=1}^{t}\eta_{t,s}\,\bar{g}_s + \sum_{s=1}^{t-1}\beta_{t,s}\,\bar{v}_s \text{ for } t=1,\ldots,T-1.
\end{aligned}
\tag{25}
$$

B.2 Joint stepsize optimization; semidefinite formulation of (3)

As mentioned in Section 2.2, a natural path forward is to use (2) to obtain worst-case optimal algorithms. That is, by solving

$$
\min_{\{(\eta_{t,s},\beta_{t,s},\gamma_{t,s})\}_{t,s}} \left\{ B_T(\{(\eta_{t,s},\beta_{t,s},\gamma_{t,s})\}_{t,s}) \text{ s.t. } \sum_{s=1}^{t-1}\gamma_{t,s}\leqslant 1,\ \gamma_{t,s}\geqslant 0 \right\}.
$$

Using (25), this problem can be reformulated as a linear optimization problem with a bilinear matrix inequality constraint, which is unfortunately NP-hard in general [Toker and Ozbay, 1995]. The bilinearity in the matrix inequality $S(\eta,\beta,\gamma;\lambda)\succcurlyeq 0$ is due to terms $\lambda_{\bar{v}_t,u}^{\mathrm{Brd}}\,(\bar{\mathrm{dir}}_t\odot(\bar{v}_t-u))$ appearing in the definition of $S(\eta,\beta,\gamma;\lambda)$ which are bilinear in (η,β) and λ^{Brd}. A classical approach (see [Drori and Teboulle, 2014]) to circumvent bilinearity in the matrix inequalities consists in using convex relaxation of the problem, which works well in our case as we explained in Section 2.2 and we detail in this section. Note that another possible approach (see [Das Gupta et al., 2024]) consists in adapting a branch-and-bound algorithm to compute the best possible regret guarantee by: (i) dividing the search space into regions, (ii) computing upper and lower bounds on the best possible regret guarantee for each region via convex relaxations of the problem, (iii) discarding regions whose lower bound is larger than the best (across all regions) current upper bound as those regions cannot contain the optimal point / value, and (iv) repeating steps (i)–(iii) with the remaining regions until convergence. However, this branch-and-bound approach is numerically more costly and is not necessary when the direct convex relaxation method works.

For this reason, we propose a slight relaxation of $B_T(\{(\eta_{t,s},\beta_{t,s},\gamma_{t,s})\}_{t,s})$ which corresponds to removing a few constraints from (2) (numerically observed to be inactive). More precisely, this relaxation is obtained through: (i) we observe that all x_t for $t=2,\cdots,T$ are in the convex hull of $x_1,v_1,\cdots,v_{T-1}$, and thus the domain constraints for $\mathcal{K}$ are imposed only on vectors $x_1,v_1,\cdots,v_{T-1},x_\star$; (ii) we keep only the boundary constraints corresponding to the optimality of v_t compared with $v_{t+1},\cdots,v_{T-1},x_\star$: this leads to the definition of $W_T\big(\{\eta_{t,s},\beta_{t,s},\gamma_{t,s}\}_{t,s}\big)$ in Section 2.2 with $W_T\big(\{\eta_{t,s},\beta_{t,s},\gamma_{t,s}\}_{t,s}\big)\geqslant B_T\big(\{\eta_{t,s},\beta_{t,s},\gamma_{t,s}\}_{t,s}\big)$. In terms of the semidefinite program reformulation (25) of (2), this relaxation corresponds to impose $\lambda_{\bar{v}_t,u}^{\mathrm{Brd}}=0$ for all

$t = 1, \ldots, T-1$ and $u \in \{\bar{v}_{t+1}, \cdots, \bar{v}_{T-1}, \bar{x}_\star\}$, giving us:

$$\begin{aligned}
W_T(\{(\eta_{t,s}, \beta_{t,s}, \gamma_{t,s})\}_{t,s}) = &\inf_{\substack{\lambda^{\text{Lip}} \geqslant 0 \\ \lambda^{\text{Diam}} \geqslant 0 \\ \lambda^{\text{Brd}} \geqslant 0}} \sum_{t=1}^{T} \lambda_t^{\text{Lip}} L^2 + \frac{1}{2} \sum_{u,\, v \in \{\bar{x}_1, \bar{v}_1, \cdots, \bar{v}_{T-1}, \bar{x}_\star\}} \lambda_{\{u,v\}}^{\text{Diam}} D^2 \\
&\text{subject to: } S(\eta, \beta, \gamma; \lambda) \succcurlyeq 0, \\
\text{where } S(\eta, \beta, \gamma; \lambda) = &\frac{1}{2} \sum_{u,\, v \in \{\bar{x}_1, \bar{v}_1, \cdots, \bar{v}_{T-1}, \bar{x}_\star\}} \lambda_{\{u,v\}}^{\text{Diam}} \left((u-v) \odot (u-v)\right) \\
&+ \sum_{t=1}^{T} \lambda_t^{\text{Lip}} \left(\bar{g}_t \odot \bar{g}_t\right) + \sum_{t=1}^{T-1} \sum_{u \in \{\bar{v}_{t+1}, \cdots, \bar{v}_{T-1}, \bar{x}_\star\}} \lambda_{\bar{v}_t, u}^{\text{Brd}} \left(\bar{\text{dir}}_t \odot (\bar{v}_t - u)\right) \\
&- \sum_{t=1}^{T} \sum_{s=1}^{t-1} \gamma_{t,s} \left(\bar{g}_t \odot \bar{v}_s\right) + \sum_{t=1}^{T} (\bar{g}_t \odot \bar{x}_\star), \\
\text{and } \lambda = (\lambda^{\text{Lip}}, \lambda^{\text{Diam}}, \lambda^{\text{Brd}}) &\text{ and } \bar{\text{dir}}_t = \sum_{s=1}^{t} \eta_{t,s}\, \bar{g}_s + \sum_{s=1}^{t-1} \beta_{t,s}\, \bar{v}_s \text{ for } t = 1, \ldots, T-1.
\end{aligned} \tag{26}$$

Using this problem, the joint minimization problem

$$\min_{\{(\eta_{t,s}, \beta_{t,s}, \gamma_{t,s})\}_{t,s}} \left\{ W_T(\{(\eta_{t,s}, \beta_{t,s}, \gamma_{t,s})\}_{t,s}) \text{ s.t. } \sum_{s=1}^{t-1} \gamma_{t,s} \leqslant 1,\ \gamma_{t,s} \geqslant 0 \right\} \tag{3}$$

is still *a priori* a linear optimization problem with a bilinear matrix inequality constraint (which recall are unfortunately NP-hard in general [Toker and Ozbay, 1995]) due to the presence in $S(\eta, \beta, \gamma; \lambda)$ of the bilinear (in (η, β) and λ^{Brd}) term (where abusing notations, we write $\bar{v}_T = \bar{x}_\star$):

$$\begin{aligned}
&\sum_{t=1}^{T-1} \sum_{u \in \{\bar{v}_{t+1}, \cdots, \bar{v}_{T-1}, \bar{x}_\star\}} \lambda_{\bar{v}_t, u}^{\text{Brd}} \left(\bar{\text{dir}}_t \odot (\bar{v}_t - u)\right) \\
&\qquad = \sum_{t=1}^{T-1} \sum_{s=1}^{t} \sum_{j=t+1}^{T} \lambda_{\bar{v}_t, \bar{v}_j}^{\text{Brd}}\, \eta_{t,s} \left(\bar{g}_s \odot (\bar{v}_t - \bar{v}_j)\right) + \sum_{t=1}^{T-1} \sum_{s=1}^{t-1} \sum_{j=t+1}^{T} \lambda_{\bar{v}_t, \bar{v}_j}^{\text{Brd}}\, \beta_{t,s} \left(\bar{v}_s \odot (\bar{v}_t - \bar{v}_j)\right).
\end{aligned}$$

However, we remark that:

$$\begin{aligned}
\sum_{t=1}^{T-1} \sum_{s=1}^{t} \sum_{j=t+1}^{T} \lambda_{\bar{v}_t, \bar{v}_j}^{\text{Brd}}\, \eta_{t,s} \left(\bar{g}_s \odot (\bar{v}_t - \bar{v}_j)\right) &= \sum_{t=1}^{T-1} \sum_{s=1}^{t} \left(\sum_{j=t+1}^{T} \lambda_{\bar{v}_t, \bar{v}_j}^{\text{Brd}}\, \eta_{t,s} \right) (\bar{g}_s \odot \bar{v}_t) - \sum_{j=1}^{T} \sum_{s=1}^{j} \sum_{t=s}^{j-1} \lambda_{\bar{v}_t, \bar{v}_j}^{\text{Brd}}\, \eta_{t,s} \left(\bar{g}_s \odot \bar{v}_j\right) \\
&= \sum_{t=1}^{T-1} \sum_{s=1}^{t} \left(\sum_{j=t+1}^{T} \lambda_{\bar{v}_t, \bar{v}_j}^{\text{Brd}}\, \eta_{t,s} - \sum_{j=s}^{t-1} \lambda_{\bar{v}_j, \bar{v}_t}^{\text{Brd}}\, \eta_{j,s} \right) (\bar{g}_s \odot \bar{v}_t),
\end{aligned}$$

and:

$$\begin{aligned}
\sum_{t=1}^{T-1} \sum_{s=1}^{t-1} \sum_{j=t+1}^{T} \lambda_{\bar{v}_t, \bar{v}_j}^{\text{Brd}}\, \beta_{t,s} \left(\bar{v}_s \odot (\bar{v}_t - \bar{v}_j)\right) &= \sum_{t=1}^{T-1} \sum_{s=1}^{t-1} \left(\sum_{j=t+1}^{T} \lambda_{\bar{v}_t, \bar{v}_j}^{\text{Brd}}\, \beta_{t,s} \right) (\bar{v}_s \odot \bar{v}_t) - \sum_{j=1}^{T} \sum_{s=1}^{j-1} \sum_{t=s+1}^{j-1} \lambda_{\bar{v}_t, \bar{v}_j}^{\text{Brd}}\, \beta_{t,s} \left(\bar{v}_s \odot \bar{v}_j\right) \\
&= \sum_{t=1}^{T-1} \sum_{s=1}^{t-1} \left(\sum_{j=t+1}^{T} \lambda_{\bar{v}_t, \bar{v}_j}^{\text{Brd}}\, \beta_{t,s} - \sum_{j=s+1}^{t-1} \lambda_{\bar{v}_j, \bar{v}_t}^{\text{Brd}}\, \beta_{j,s} \right) (\bar{v}_s \odot \bar{v}_t).
\end{aligned}$$

This motivates the following change of variables in (26) which allows us to recast (3) as a linear convex semidefinite program again: η, β and λ^{Brd} are replaced by:

$$\begin{aligned}
B_{t,s} &= \eta_{t,s} \sum_{j=t+1}^{T} \lambda_{\bar{v}_t, \bar{v}_j}^{\text{Brd}} - \sum_{j=s}^{t-1} \eta_{j,s}\, \lambda_{\bar{v}_j, \bar{v}_t}^{\text{Brd}} \qquad \forall 1 \leqslant s \leqslant t \leqslant T, \\
C_{t,s} &= \beta_{t,s} \sum_{j=t+1}^{T} \lambda_{\bar{v}_t, \bar{v}_j}^{\text{Brd}} - \sum_{j=s+1}^{t-1} \beta_{j,s}\, \lambda_{\bar{v}_j, \bar{v}_t}^{\text{Brd}} \qquad \forall 1 \leqslant s < t \leqslant T.
\end{aligned} \tag{27}$$

Note that the other variables, that is, γ, λ^{Lip} and λ^{Diam}, are left unchanged. Also note that from the definition of $B_{t,s}$ and $C_{t,s}$, we get that they must satisfy the constraints $\sum_{s=t}^{T} B_{t,s} = 0$ and $\sum_{s=t+1}^{T} B_{t,s} = 0$ for all $t = 1, \dots, T$. Thus, (3) can be reformulated as the following linear convex semidefinite program:

$$\begin{aligned}
&\inf_{\{(B_{t,s},C_{t,s},\gamma_{t,s})\}_{t,s}} \inf_{\substack{\lambda^{\mathrm{Lip}}\geqslant 0\\ \lambda^{\mathrm{Diam}}\geqslant 0\\ \lambda^{\mathrm{Brd}}\geqslant 0}} \sum_{t=1}^{T} \lambda_t^{\mathrm{Lip}} L^2 + \frac{1}{2} \sum_{u,\, v\in\{\bar{x}_1,\bar{v}_1,\cdots,\bar{v}_{T-1},\bar{x}_\star\}} \lambda_{\{u,v\}}^{\mathrm{Diam}} D^2 \\
&\qquad \text{subject to: } S(B,C,\gamma;\lambda) \succcurlyeq 0, \\
&\qquad\qquad \sum_{t=s}^{T} B_{t,s} = 0 \text{ and } \sum_{t=s+1}^{T} C_{t,s} = 0 \text{ for } s = 1,\dots,T \\
&\text{where } S(B,C,\gamma;\lambda) = \frac{1}{2} \sum_{u,\, v\in\{\bar{x}_1,\bar{v}_1,\cdots,\bar{v}_{T-1},\bar{x}_\star\}} \lambda_{\{u,v\}}^{\mathrm{Diam}} \left((u-v)\odot(u-v)\right) \\
&\qquad + \sum_{t=1}^{T} \lambda_t^{\mathrm{Lip}} \left(\bar{g}_t \odot \bar{g}_t\right) + \sum_{t=1}^{T}\sum_{s=1}^{t} B_{t,s}\left(\bar{g}_s\odot\bar{v}_t\right) + \sum_{t=1}^{T}\sum_{s=1}^{t-1} C_{t,s}\left(\bar{v}_s\odot\bar{v}_t\right) \\
&\qquad - \sum_{t=1}^{T}\sum_{s=1}^{t-1} \gamma_{t,s}\left(\bar{g}_t\odot\bar{v}_s\right) + \sum_{t=1}^{T}(\bar{g}_t\odot\bar{x}_\star), \\
&\text{and } \lambda = (\lambda^{\mathrm{Lip}},\lambda^{\mathrm{Diam}},\lambda^{\mathrm{Brd}}).
\end{aligned} \tag{28}$$

Note that there are T^2 of the variables $\{B_{t,s}, C_{t,s}\}_{t,s}$, while there was $(T-1)^2 + T(T-1)/2$ of the variables η, β and λ^{Brd}; this is because problem (26) was overparametrized. Thus, when inverting the change of variables, for given values of $\{B_{t,s}, C_{t,s}\}_{t,s}$, there exist several solutions for η, β and λ^{Brd} solving (27). We propose one solution for inverting (27) giving simple algorithms: (i) we choose $\eta_{t,s} = 1$ for all $1 \leqslant s \leqslant t \leqslant T$, (ii) we solve for λ^{Brd} using the first line of (27) substituting in the value of η, (iii) we solve for β using the second line of (27) substituting in the value of λ^{Brd}; that is:

$$\begin{aligned}
&\eta_{t,s} = 1 && \forall 1 \leqslant s \leqslant t \leqslant T, \\
&\lambda_{\bar{v}_s,\bar{v}_t}^{\mathrm{Brd}} = B_{t,s+1} - B_{t,s} && \forall 1 \leqslant s < t \leqslant T, \\
&\beta_{t,s} = \frac{1}{B_{t,t}} \left(C_{t,s} + \sum_{j=s+1}^{t-1} \beta_{j,s}\, \lambda_{\bar{v}_j,\bar{v}_t}^{\mathrm{Brd}} \right) && \forall 1 \leqslant s < t \leqslant T.
\end{aligned} \tag{29}$$

B.3 Design and optimization of the proof of Theorem 3.1

As we explained in Section 3.3, we used variants of (2) to design the potential-based proof of Section 3 and to jointly optimize the algorithm and potential parameters for this proof. We outlined our general method in Section 3.3, and here we present the details for rewriting (8) as a semidefinite program, numerically solving it, and then for obtaining the optimal parameters given at the end of Section 3.3.

We start by restating the 1-iteration inner maximization problem from (8) which upper bounds the potential increase for an abstract t and for given pontential parameters a and b:

$$\begin{aligned}
B_t(\eta,\sigma,a,b) \triangleq &\sup_{\substack{\mathcal{K},\, d\in\mathbb{N}\\ g_t,\, G_{t-1}\\ x_1,\, x_t,\, v_t,\, x_{t+1}\\ y_t,\, y_{t+1}}} \phi_t - \phi_{t-1} \\
&\text{subject to: } \phi_t - \phi_{t-1} \text{ is generated from } \{x_t, x_{t+1}, y_t, y_{t+1}, g_t, G_{t-1}, x_1\} \text{ by (7)}, \\
&\qquad \|g_t\| \leqslant L, \\
&\qquad \mathcal{K} \text{ is a non-empty closed convex set of } \mathbb{R}^d, \\
&\qquad \mathrm{Diam}(\{x_1, x_t, v_t, x_{t+1}, y_t, y_{t+1}\}) \leqslant D, \\
&\qquad (x_{t+1}, v_t) \text{ are generated from } \{x_1, x_t, g_t, G_{t-1}\} \text{ by Algorithm 1}, \\
&\qquad y_t \text{ and } y_{t+1} \text{ are generated from } \{x_1, g_t, G_{t-1}\} \text{ by FTRL}.
\end{aligned} \tag{30}$$

In particular, we have that (8) is equal to the minimum of $B_t(\eta,\sigma,a,b)$ over $a \geqslant 0$ and $b \geqslant 0$.

Note that x_{t+1} can be removed from the boundary conditions as it is combination of v_t and x_t. Substituting in the definition of $\phi_t - \phi_{t-1}$ and the optimality condition of v_t, y_t and y_{t+1}, we can get rid of $\mathcal{K}$, and we obtain the following self-containing reformulation of (30):

$$
\begin{aligned}
B_t(\eta,\sigma,a,b) = \sup_{\substack{d\in\mathbb{N}\\ g_t,\,G_{t-1}\\ x_1,\,x_t,\,v_t,\,x_{t+1}\\ y_t,\,y_{t+1}}} & \left\{\begin{aligned}
&\langle g_t, x_t - y_{t+1}\rangle + \langle G_{t-1}, y_t - y_{t+1}\rangle\\
&+ a(\|x_{t+1}-y_{t+1}\|^2 - \|x_t - y_t\|^2)\\
&+ b\,\eta(\langle G_{t-1}+g_t, x_{t+1}-y_{t+1}\rangle - \langle G_{t-1}, x_t - y_t\rangle)\\
&+ \frac{b}{2}\big(\|x_{t+1}-x_1\|^2 - \|y_{t+1}-x_1\|^2\\
&\qquad -\|x_t - x_1\|^2 + \|y_t - x_1\|^2\big)\\
&+ \frac{1}{2\eta}(\|y_t - x_1\|^2 - \|y_{t+1}-x_1\|^2)
\end{aligned}\right\}\\
\text{subject to: } & \|g_t\| \leqslant L,\\
& \mathrm{Diam}(\{x_1, x_t, v_t, y_t, y_{t+1}\}) \leqslant D,\\
& x_{t+1} = \sigma v_t + (1-\sigma)x_t,\\
& \langle \eta\,G_{t-1} + (y_t - x_1), y_t - u\rangle \leqslant 0 \text{ for } u \in \{x_t, v_t, y_{t+1}\},\\
& \langle \eta\,(G_{t-1}+g_t) + (y_{t+1}-x_1), y_{t+1}-u\rangle \leqslant 0 \text{ for } u \in \{x_t, v_t, y_t\},\\
& \langle \eta\,(G_{t-1}+g_t) + (x_t - x_1), v_t - u\rangle \leqslant 0 \text{ for } u \in \{x_t, y_t, y_{t+1}\}.
\end{aligned}
\tag{31}
$$

Without loss of generality, we assume that $x_1 = 0$. Then, using the same method as detailed in Appendix B.1, we can reformulate this problem as a linear convex semidefinite program. To this end, we do a change of variables in (31), replacing the gradients G_{t-1} and g_t and the vectors x_t, v_t, y_t and y_{t+1} by their Gram matrix H. Let $e_1, \dots, e_6$ be the standard basis vectors of $\mathbb{R}^6$. Define $\bar{g}_t = e_1$, $\bar{G}_{t-1} = e_2$, $\bar{x}_t = e_3$, $\bar{v}_t = e_4$, $\bar{y}_t = e_5$ and $\bar{y}_{t+1} = e_6$. Thus, we have $\bar{g}_t^{\mathrm{T}} H \bar{x}_t = \langle g_t, x_t\rangle$, and similarly with the others gradients / vectors. Define the matrix $A_{\mathrm{obj}}(\eta,\sigma,a,b)$, which is linear in (a,b) but not in (η,σ,a,b), as:

$$
\begin{aligned}
A_{\mathrm{obj}}(\eta,\sigma,a,b) \triangleq\ & \bar{g}_t \odot (\bar{x}_t - \bar{y}_{t+1}) + \bar{G}_{t-1} \odot (\bar{y}_t - \bar{y}_{t+1})\\
&+ a((\bar{x}_{t+1} - \bar{y}_{t+1}) \odot (\bar{x}_{t+1} - \bar{y}_{t+1}) - (\bar{x}_t - \bar{y}_t) \odot (\bar{x}_t - \bar{y}_t))\\
&+ b\,\eta((\bar{G}_{t-1} + \bar{g}_t) \odot (\bar{x}_{t+1} - \bar{y}_{t+1}) - \bar{G}_{t-1} \odot (\bar{x}_t - \bar{y}_t))\\
&+ \frac{b}{2}(\bar{x}_{t+1} \odot \bar{x}_{t+1} - \bar{y}_{t+1} \odot \bar{y}_{t+1} - \bar{x}_t \odot \bar{x}_t + \bar{y}_t \odot \bar{y}_t)\\
&+ \frac{1}{2\eta}(\bar{y}_t \odot \bar{y}_t - \bar{y}_{t+1} \odot \bar{y}_{t+1}),
\end{aligned}
\tag{32}
$$

where $\bar{x}_{t+1} = \sigma \bar{v}_t + (1-\sigma)\bar{x}_t$. Also define:

$$
\begin{aligned}
\mathrm{dir}_{\bar{v}_t} &= \eta\,\bar{G}_{t-1} + \eta\,\bar{g}_t + \bar{x}_t,\\
\mathrm{dir}_{\bar{y}_t} &= \eta\,\bar{G}_{t-1} + \bar{y}_t,\\
\mathrm{dir}_{\bar{y}_{t+1}} &= \eta\,\bar{G}_{t-1} + \eta\,\bar{g}_t + \bar{y}_{t+1}.
\end{aligned}
$$

With this change of variables, (31) can be reformulated as the following linear convex semidefinite program:

$$
\begin{aligned}
B_t(\eta,\sigma,a,b) = \sup_{H\succcurlyeq 0}\ & \mathrm{tr}(A_{\mathrm{obj}}\,H)\\
\text{subject to: } & \mathrm{tr}((\bar{g}_t \odot \bar{g}_t)\,H) \leqslant L^2,\\
& \mathrm{tr}((u \odot v)\,H) \leqslant D^2 \text{ for } u, v \in \{\bar{x}_1, \bar{x}_t, \bar{v}_t, \bar{x}_{t+1}, \bar{y}_t, \bar{y}_{t+1}\},\\
& \mathrm{tr}\big((\mathrm{dir}_u \odot (u - v))\,H\big) \leqslant 0 \text{ for } u \in \{\bar{v}_t, \bar{y}_t, \bar{y}_{t+1}\}\\
&\qquad\qquad \text{and } v \in \{\bar{x}_1, \bar{x}_t, \bar{v}_t, \bar{y}_t, \bar{y}_{t+1}\} \setminus \{u\}.
\end{aligned}
\tag{33}
$$

We then reformulate (33) as its Lagrange dual, giving us:

$$
\begin{aligned}
B_t(\eta,\sigma,a,b) = \inf_{\substack{\lambda^{\mathrm{Lip}}\geqslant 0\\ \lambda^{\mathrm{Diam}}\geqslant 0\\ \lambda^{\mathrm{Brd}}\geqslant 0}} \; & \lambda^{\mathrm{Lip}}\, L^2 + \frac{1}{2}\sum_{u,\, v\in\{\bar{x}_1,\bar{x}_t,\bar{v}_t,\bar{y}_t,\bar{y}_{t+1}\}} \lambda^{\mathrm{Diam}}_{\{u,v\}}\, D^2 \\
\text{subject to: } & S(\eta,\sigma,a,b;\lambda) \succcurlyeq 0, \\
\text{where } S(\eta,\sigma,a,b;\lambda) = \; & \frac{1}{2}\sum_{u,\, v\in\{\bar{x}_1,\bar{x}_t,\bar{v}_t,\bar{y}_t,\bar{y}_{t+1}\}} \lambda^{\mathrm{Diam}}_{\{u,v\}}\,((u-v)\odot(u-v)) \\
& + \sum_{u\in\{\bar{v}_t,\bar{y}_t,\bar{y}_{t+1}\}}\;\sum_{v\in\{\bar{x}_1,\bar{x}_t,\bar{v}_t,\bar{y}_t,\bar{y}_{t+1}\}\setminus\{u\}} \lambda^{\mathrm{Brd}}_{u,v}\,(\bar{\mathrm{dir}}_u\odot(u-v)) \\
& + \lambda^{\mathrm{Lip}}\,(\bar{g}_t\odot\bar{g}_t) - A_{\mathrm{obj}}(\eta,\sigma,a,b), \\
\text{and } \lambda = (\lambda^{\mathrm{Lip}},\lambda^{\mathrm{Diam}},\lambda^{\mathrm{Brd}}). &
\end{aligned}
\tag{34}
$$

Now, jointly minimizing (34) in (a,b) and λ for fixed (η,σ), we get a reformulation of (8) as the following linear convex semidefinite program:

$$
\begin{aligned}
\inf_{a\geqslant 0, b\geqslant 0}\;\inf_{\substack{\lambda^{\mathrm{Lip}}\geqslant 0\\ \lambda^{\mathrm{Diam}}\geqslant 0\\ \lambda^{\mathrm{Brd}}\geqslant 0}} \; & \lambda^{\mathrm{Lip}}\, L^2 + \frac{1}{2}\sum_{u,\, v\in\{\bar{x}_1,\bar{x}_t,\bar{v}_t,\bar{y}_t,\bar{y}_{t+1}\}} \lambda^{\mathrm{Diam}}_{\{u,v\}}\, D^2 \\
\text{subject to: } & S(\eta,\sigma,a,b;\lambda) \succcurlyeq 0, \\
\text{where } S(\eta,\sigma,a,b;\lambda) = \; & \frac{1}{2}\sum_{u,\, v\in\{\bar{x}_1,\bar{x}_t,\bar{v}_t,\bar{y}_t,\bar{y}_{t+1}\}} \lambda^{\mathrm{Diam}}_{\{u,v\}}\,((u-v)\odot(u-v)) \\
& + \sum_{u\in\{\bar{v}_t,\bar{y}_t,\bar{y}_{t+1}\}}\;\sum_{v\in\{\bar{x}_1,\bar{x}_t,\bar{v}_t,\bar{y}_t,\bar{y}_{t+1}\}\setminus\{u\}} \lambda^{\mathrm{Brd}}_{u,v}\,(\bar{\mathrm{dir}}_u\odot(u-v)) \\
& + \lambda^{\mathrm{Lip}}\,(\bar{g}_t\odot\bar{g}_t) - A_{\mathrm{obj}}(\eta,\sigma,a,b), \\
\text{and } \lambda = (\lambda^{\mathrm{Lip}},\lambda^{\mathrm{Diam}},\lambda^{\mathrm{Brd}}). &
\end{aligned}
\tag{35}
$$

Note that the size of (35) does not depend on T, which allows efficient numerical solving even for large values of T. Numerically solving (35) gives us that for jointly optimal (a,b) and λ we have $b=0$, $\lambda^{\mathrm{Diam}}_{\{u,v\}}=0$ except for $\lambda^{\mathrm{Diam}}_{\{\bar{x}_t,\bar{v}_t\}}=a\sigma^2$, $\lambda^{\mathrm{Brd}}_{u,v}=0$ except for $\lambda^{\mathrm{Brd}}_{\bar{v}_t,\bar{y}_{t+1}}=\lambda^{\mathrm{Brd}}_{\bar{y}_{t+1},\bar{v}_t}=2a\sigma$, $\lambda^{\mathrm{Brd}}_{\bar{y}_{t+1},\bar{y}_t}$ and $\lambda^{\mathrm{Brd}}_{\bar{y}_t,\bar{y}_{t+1}}=1+\lambda^{\mathrm{Brd}}_{\bar{y}_{t+1},\bar{y}_t}$. Following our notations from the proof of Lemma 3.2 (see Appendix A.1), we write $\lambda^g=\lambda^{\mathrm{Lip}}$ and $\tilde{\lambda}=\lambda^{\mathrm{Brd}}_{\bar{y}_{t+1},\bar{y}_t}$. Note that the Cholesky decomposition of $S=S(\eta,\sigma,a,b;\lambda)$ encodes for the sum of squares $\mathrm{tr}(S\,H)\geqslant 0$ that must be added during the proof, where some of those squares can be interpreted as inequalities. The Cholesky decomposition of S has 2 eigenvectors: the first eigenvector $\sqrt{\lambda^g}\,\bar{g}_t+\frac{1}{2\sqrt{\lambda^g}}\,(\bar{x}_t-\bar{y}_{t+1}+\tilde{\lambda}(\bar{y}_t-\bar{y}_{t+1}))$ encodes for the inequality

$$
\langle g_t, x_t-y_{t+1}+\tilde{\lambda}(y_t-y_{t+1})\rangle \leqslant \lambda^g\|g_t\|^2+\frac{1}{4\lambda^g}\|x_t-y_{t+1}+\tilde{\lambda}(y_t-y_{t+1})\|^2,
$$

while the second eigenvector indicates that the upper bound we get from the proof will contain minus a sum of squares, which is composed of the vectors x_t, y_t and y_{t+1}. The values of all those Lagrange multipliers in λ and the value of S gives us what inequalities and in which quantity to use in the proof of Lemma 3.2. Thus, from all of this we can deduce the proof up to showing that $Q(\tilde{\lambda})$ is a sum of squares, where:

$$
\begin{aligned}
Q(\tilde{\lambda}) \triangleq \frac{1+2\tilde{\lambda}}{2\eta}\|y_t-y_{t+1}\|^2 - \frac{1}{4\lambda^g}\|x_t-y_{t+1}+\tilde{\lambda}(y_t-y_{t+1})\|^2 \\
- a((1-2\sigma)\|x_t-y_{t+1}\|^2-\|x_t-y_t\|^2).
\end{aligned}
$$

Hence, all that is left to do is finding optimal η, σ, a λ^g such that there exists $\tilde{\lambda}\geqslant 0$ for which $Q(\tilde{\lambda})$ is a sum of squares and that minimizes the upper bound given by the proof on the regret of OFW R_T, which is:

$$
\frac{1}{2\eta}\|x_\star-x_1\| + \sum_{t=1}^{T} B_t(\eta,\sigma,a,0) \leqslant \frac{1}{2\eta}D^2+\lambda^g L^2 T + a\sigma^2 D^2 T.
$$

Then, we rewrite $Q(\tilde{\lambda})$ as:

$$\begin{aligned} Q(\tilde{\lambda}) &= \left(\frac{1+2\tilde{\lambda}}{2\eta} + a\right)\|y_t - y_{t+1}\|^2 + 2a\sigma\|x_t - y_{t+1}\|^2 + 2a\langle x_t - y_{t+1}, y_{t+1} - y_t\rangle \\ &\quad - \frac{1}{4\lambda^g}\|x_t - y_{t+1} + \tilde{\lambda}(y_t - y_{t+1})\|^2 \\ &= \left(\frac{1+2\tilde{\lambda}}{2\eta} + a - \frac{\tilde{\lambda}^2}{4\lambda^g}\right)\|y_t - y_{t+1}\|^2 + \left(2a\sigma - \frac{1}{4\lambda^g}\right)\|x_t - y_{t+1}\|^2 \\ &\quad + \left(2a + \frac{\tilde{\lambda}}{2\lambda^g}\right)\langle x_t - y_{t+1}, y_{t+1} - y_t\rangle. \end{aligned}$$

Thus, we get that $Q(\tilde{\lambda})$ is a sum of squares if and only if:

$$\begin{cases} \left(2a\sigma - \frac{1}{4\lambda^g}\right) \geqslant 0 \\ \left(a + \frac{\tilde{\lambda}}{4\lambda^g}\right)^2 \leqslant \left(2a\sigma - \frac{1}{4\lambda^g}\right)\left(\frac{1+2\tilde{\lambda}}{2\eta} + a - \frac{\tilde{\lambda}^2}{4\lambda^g}\right), \end{cases}$$

where the second line is equivalent to:

$$\Big(\frac{a\sigma}{2\lambda^g}\Big)\tilde{\lambda}^2 - \Big(\frac{1}{\eta}\big(2a\sigma - \frac{1}{4\lambda^g}\big) - \frac{a}{2\lambda^g}\Big)\tilde{\lambda} + \Big((1-2\sigma)a^2 + \frac{a}{4\lambda^g} - \frac{1}{2\eta}\big(2a\sigma - \frac{1}{4\lambda^g}\big)\Big) \leqslant 0. \tag{36}$$

An elementary second order polynomial study indicates that this polynomial equation has no non-negative solution $\tilde{\lambda} \geqslant 0$ when the linear coefficient is positive, that is, we get the necessary condition:

$$\Big(\frac{1}{\eta}\big(2a\sigma - \frac{1}{4\lambda^g}\big) - \frac{a}{2\lambda^g}\Big) \geqslant 0. \tag{37}$$

When (37) holds, we can upper bound the constant coefficient of (36) by its leading order term a^2, giving us the (slightly) more conservative but easier no analyze condition:

$$\Big(\frac{a\sigma}{2\lambda^g}\Big)\tilde{\lambda}^2 - \Big(\frac{1}{\eta}\big(2a\sigma - \frac{1}{4\lambda^g}\big) - \frac{a}{2\lambda^g}\Big)\tilde{\lambda} + a^2 \leqslant 0. \tag{38}$$

Moreover, when (37) holds, we have that (38) has a non-negative solution $\tilde{\lambda} \geqslant 0$ if and only if:

$$\Big(\frac{1}{\eta}\big(2a\sigma - \frac{1}{4\lambda^g}\big) - \frac{a}{2\lambda^g}\Big)^2 - 4\Big(\frac{a\sigma}{2\lambda^g}\Big)a^2 \geqslant 0.$$

Hence, combining all of the above, we get that $Q(\tilde{\lambda})$ is a sum of squares whenever:

$$\begin{cases} \left(2a\sigma - \frac{1}{4\lambda^g}\right) \geqslant 0 \\ \Big(\frac{1}{\eta}\big(2a\sigma - \frac{1}{4\lambda^g}\big) - \frac{a}{2\lambda^g}\Big) \geqslant 0 \\ \Big(\frac{1}{\eta}\big(2a\sigma - \frac{1}{4\lambda^g}\big) - \frac{a}{2\lambda^g}\Big)^2 - 4\Big(\frac{a\sigma}{2\lambda^g}\Big)a^2 \geqslant 0. \end{cases} \tag{39}$$

Hence, to find the values of η, σ, a λ^g giving the optimal upper bound, we must solve the problem:

$$\inf_{\eta,\sigma,a,\lambda^g \geqslant 0} \quad \frac{1}{2\eta}D^2 + \lambda^g L^2 T + a\sigma^2 D^2 T$$
$$\text{subject to: } (\eta,\sigma,a,\lambda^g) \text{ satisfy (39), and } \sigma \leqslant 1.$$

Solving this non-convex problem algebraically via Lagrange relaxation (note that only the last constraint in (39) is saturated at the optimum) we get that the optimal values are:

$$\eta = \frac{D3^{3/4}}{2LT^{3/4}}, \quad \sigma = \frac{\sqrt{3}}{\sqrt{T}}, \quad a = \frac{1}{6\eta}, \quad \text{and,} \quad \lambda^g = 2a\sigma^2\frac{D^2}{L^2} = \frac{D^2}{\eta T L^2}.$$

Plugging those values into $Q(\tilde{\lambda})$ then allows us to conclude the proof of Lemma 3.2. This concludes the design of the optimal proof of Lemma 3.2.

T	10	15	20	25	30	35	40	45	50	55
time (s)	0.27	0.74	1.62	3.35	6.69	14.17	22.97	40.76	62.63	94.35
n	60	65	70	75	80	85	90	95		
time (s)	140.88	221.85	313.78	533.60	773.32	1202.68	1603.33	2126.25		
time (min)	2.35	3.70	5.23	8.89	12.89	20.04	26.72	35.44		

Table 1: Computation time for numerically solving (2) for different values of T with parameters fixed as in (4) (with the appropriate transformation of parameters to go from Algorithm 1 to the definition of (1) used in (2)).

T	10	15	20	25	30	35	40	45	50	55
time (s)	0.15	0.28	0.51	0.88	1.65	2.97	4.23	6.28	11.30	15.82
T	60	65	70	75	80	85	90	95	100	
time (s)	20.94	31.53	42.12	55.64	63.04	98.36	109.48	146.80	194.84	

Table 2: Computation time for numerically solving (3) for different values of T.

B.4 Examples of computation times for computing PEPs (2) and (3)

In this section, we provide some examples of computation times for numerically solving the PEPs (2) and (3) for different values of horizon time T. Table 1 presents the computation times for solving the PEP (2) with parameters fixed as in (4) (with the appropriate transformation of parameters to go from Algorithm 1 to the definition of (1) used in (2)). Table 2 presents the computation times for solving the design PEP (3) which optimizes the parameters of the algorithm, thus giving us the optimal worst-case regret guarantee for any OFW-type algorithm of the form (1). Note that the computation times for solving (3) are significantly smaller than those for solving (2) with fixed parameters, which appears to be due to the fact that, as the design PEP (3) jointly optimizes the algorithm parameters and its regret bound proof, it can find more structured proofs that allow for more efficient numerical solving of the underlying semidefinite program.

B.5 Examples of numerically optimized stepsize patterns

The following list provides numerical examples of optimal parameter values for the online Frank–Wolfe-type algorithm (1) for $T = 2, \dots, 6$, together with their optimal worst-case regret guarantees. (Note that for $T = 1$ the algorithm is necessarily trivial as we can only play the uninformed choice x_1.) Those values were obtained by numerically solving the relaxed linear convex semidefinite program (28) with $L = D = 1$. Note that we chose the parameters given by (29) and thus we do not report the values of $\eta_{t,s} = 1$ for $1 \leqslant s \leqslant t \leqslant T-1$. We present the values of the parameters $\{\gamma_{t,s}\}_{1 \leqslant s < t \leqslant T}$ and $\{\beta_{t,s}\}_{1 \leqslant s < t \leqslant T-1}$ as square $T \times T$ and $(T-1) \times (T-1)$ matrices, respectively, where the values for out of range indices are left blank intentionally.

- For $T = 2$, we have $R_T \leqslant B_2 \leqslant 1.7321$ and:

$$[\gamma_{t,s}] = \begin{bmatrix} & \\ 0.5 & \end{bmatrix}, \quad [\beta_{t,s}] = \begin{bmatrix} \ \end{bmatrix}.$$

- For $T = 3$, we have $R_T \leqslant B_3 \leqslant 2.3421$ and:

$$[\gamma_{t,s}] = \begin{bmatrix} & & \\ 0.5 & & \\ 0.3118 & 0.3764 & \end{bmatrix}, \quad [\beta_{t,s}] = \begin{bmatrix} & \\ -0.1099 & \end{bmatrix}.$$

- For $T = 4$, we have $R_T \leqslant B_4 \leqslant 2.9029$ and:

$$[\gamma_{t,s}] = \begin{bmatrix} & & & \\ 0.5 & & & \\ 0.3133 & 0.3734 & & \\ 0.1843 & 0.2197 & 0.4116 & \end{bmatrix}, \quad [\beta_{t,s}] = \begin{bmatrix} & & \\ 0.1961 & & \\ -0.4249 & 0.1465 & \end{bmatrix}.$$

- For $T = 5$, we have $R_T \leqslant B_5 \leqslant 3.4217$ and:

$$[\gamma_{t,s}] = \begin{bmatrix} 0.5 & & & \\ 0.3067 & 0.3866 & & \\ 0.2124 & 0.2677 & 0.3075 & \\ 0.1201 & 0.1514 & 0.1739 & 0.4345 \end{bmatrix},$$

$$[\beta_{t,s}] = \begin{bmatrix} 0.5649 & & \\ -0.2595 & 0.212 & \\ -0.6282 & -0.253 & 0.3473 \end{bmatrix}.$$

- For $T = 6$, we have $R_T \leqslant B_6 \leqslant 3.917$ and:

$$[\gamma_{t,s}] = \begin{bmatrix} 0.5 & & & & \\ 0.3068 & 0.3863 & & & \\ 0.2101 & 0.2646 & 0.3151 & & \\ 0.1406 & 0.177 & 0.2108 & 0.3309 & \\ 0.0784 & 0.0985 & 0.1174 & 0.1842 & 0.4432 \end{bmatrix},$$

$$[\beta_{t,s}] = \begin{bmatrix} 0.5856 & & & \\ -0.0481 & 0.4808 & & \\ -0.5053 & -0.0949 & 0.3922 & \\ -0.7675 & -0.4251 & -0.001 & 0.3876 \end{bmatrix}.$$

Note that the structure of the values of $\{\gamma_{t,s}\}_{1\leqslant s<t\leqslant T}$ implies that we get the update rule $x_{t+1} = \sigma_t v_t + (1-\sigma_t)x_t$ with $\sigma_t = \gamma_{t+1,t}$ in (1).

B.6 Joint stepsize optimization with multiple linear optimization rounds per iteration

As explained in Section 4, the joint stepsize optimization method for online Frank–Wolfe-type algorithms explained in Section 2.2 and in Appendix B.2 can be easily adapted to the variant algorithm with multiple linear optimization rounds per iteration defined in (9). In this section, we explain the details on how to make this adaptation.

We first define the worst-case regret for (9) with given parameters by simply replacing (1) by (9) in (2):

$$\begin{aligned} \tilde{B}_T(\{(\eta_{t,k,s}, \beta_{t,k,s,j}, \gamma_{t,s,j})\}_{t,k,s,j}) \triangleq \sup_{\substack{\mathcal{K},\{\ell_t\}_{t=1,\ldots,T} \\ x_\star,\{x_t\}_{t=1,\ldots,T} \\ d\in\mathbb{N}}} \quad & R_T(x_1,\ldots,x_T;x_\star) \\ \text{subject to:} \quad & \ell_t \text{ is convex and } L\text{-Lipschitz for } t=1,\ldots,T, \\ & \mathcal{K} \text{ is a non-empty closed convex set of } \mathbb{R}^d, \\ & \mathrm{Diam}(\mathcal{K}) \leqslant D, \\ & \{x_t\}_{t=1,\ldots,T} \text{ is generated by (9).} \end{aligned}$$

To adapt the joint minimization problem (3) to (9), we need to replace $W_T\big(\{\eta_{t,s},\beta_{t,s},\gamma_{t,s}\}_{t,s}\big)$ from Section 4 by some relaxation of the problem above. More precisely, this relaxation is obtained through: (i) we observe that all x_t for $t = 2,\cdots,T$ are in the convex hull of $x_1, v_{1,1},\cdots,v_{1,r},\cdots,v_{T-1,r}$, and thus the domain constraints for $\mathcal{K}$ are imposed only on vectors $x_1, v_{1,1},\cdots,v_{T-1,r},x_\star$; (ii) we only keep the boundary constraints corresponding to the optimality of $v_{t,k}$ compared with $v_{t,k+1},\ldots,v_{t,r},v_{t+1,1},\ldots,v_{T-1,r},x_\star$:

$$\begin{aligned}
&\tilde{W}_T(\{(\eta_{t,k,s},\beta_{t,k,s,j},\gamma_{t,s,j})\}_{t,k,s,j}) \triangleq \\
&\qquad \sup_{\substack{\{g_t\}_{t=1,\ldots,T},\, x_\star \\ \{(x_t,v_{t,k},\mathrm{dir}_{t,k})\}_{t=1,\ldots,T,k=1,\ldots,r} \\ d\in\mathbb{N}}} \quad \sum_{t=1}^T \langle g_t, x_t - x_\star\rangle \\
&\qquad\qquad \text{subject to: } \{(x_t,\mathrm{dir}_{t,1},\ldots,\mathrm{dir}_{t,r})\}_{t=1,\ldots,T} \text{ compatible with (9)}, \\
&\qquad\qquad\qquad \langle -\mathrm{dir}_{t,k}, u - v_{t,k}\rangle \leqslant 0 \text{ for all } t=1,\ldots,T-1 \text{ and } k=1,\ldots,r \\
&\qquad\qquad\qquad\qquad \text{and } u\in\{v_{t,k+1},,\cdots,v_{t,r},v_{t+1,1},\cdots,v_{T-1,r},x_\star\}, \\
&\qquad\qquad\qquad \mathrm{Diam}(\{x_1,v_{1,1},\cdots,v_{1,r},\cdots,v_{T-1,r},x_\star\}) \leqslant D, \\
&\qquad\qquad\qquad \|g_t\| \leqslant L \text{ for } t=1,\ldots,T,
\end{aligned} \tag{40}$$

where the line "$\{(x_t,\mathrm{dir}_{t,1},\ldots,\mathrm{dir}_{t,r})\}_{t=1,\ldots,T}$ *compatible with* (9)" means that x_t and $\mathrm{dir}_{t,k}$ can be substituted by their expressions in (9), leading to $\tilde{W}_T\big(\{(\eta_{t,k,s},\beta_{t,k,s,j},\gamma_{t,s,j})\}_{t,k,s,j}\big) \geqslant \tilde{B}_T\big(\{(\eta_{t,k,s},\beta_{t,k,s,j},\gamma_{t,s,j})\}_{t,k,s,j}\big)$.

We can now formulate the adaptation of the joint minimization problem (3) to (9), giving us:

$$\min_{\{(\eta_{t,k,s},\beta_{t,k,s,j},\gamma_{t,s,j})\}_{t,k,s,j}} \left\{ \tilde{W}_T(\{(\eta_{t,k,s},\beta_{t,k,s,j},\gamma_{t,s,j})\}_{t,k,s,j}) \text{ s.t.} \sum_{s=1}^{t-1}\sum_{j=1}^{r} \gamma_{t,s,j} \leqslant 1,\ \gamma_{t,s,j}\geqslant 0 \right\}. \tag{41}$$

In the remaining of this section, we will need the following notations. For integers $t,s\in\mathbb{N}$ and $k,j\in[\![1,r]\!]$, we write $(t,k)\leqslant_{\mathrm{lex}}(s,j)$ if $t<s$, or $t=s$ and $k<j$. We write $(t,k)<_{\mathrm{lex}}(s,j)$ if $(t,k)\leqslant_{\mathrm{lex}}(s,j)$ and $(t,k)\neq(s,j)$.

Then, the method detailed in Appendix B.1 and B.2 can be immediately adapted to reformulate (40) as a linear convex semidefinite program and then form its Lagrange dual, giving us:

$$\begin{aligned}
\tilde{W}_T(\{(\eta_{t,k,s},\beta_{t,k,s,j},\gamma_{t,s,j})\}_{t,k,s,j}) &= \inf_{\substack{\lambda^{\mathrm{Lip}}\geqslant 0\\ \lambda^{\mathrm{Diam}}\geqslant 0\\ \lambda^{\mathrm{Brd}}\geqslant 0}} \sum_{t=1}^T \lambda_t^{\mathrm{Lip}} L^2 + \frac{1}{2}\sum_{u,\,v\in\{\bar{x}_1,\bar{v}_{1,1},\cdots,\bar{v}_{T-1,r},\bar{x}_\star\}} \lambda^{\mathrm{Diam}}_{\{u,v\}} D^2 \\
&\qquad \text{subject to: } S(\eta,\beta,\gamma;\lambda)\succcurlyeq 0, \\
\text{where } S(\eta,\beta,\gamma;\lambda) = &\sum_{t=1}^T \lambda_t^{\mathrm{Lip}} (\bar{g}_t\odot\bar{g}_t) + \frac{1}{2}\sum_{u,\,v\in\{\bar{x}_1,\bar{v}_{1,1},\cdots,\bar{v}_{T-1,r},\bar{x}_\star\}} \lambda^{\mathrm{Diam}}_{\{u,v\}}((u-v)\odot(u-v)) \\
&+ \sum_{t=1}^{T-1}\sum_{k=1}^{r}\sum_{u\in\{\bar{v}_{t,k+1},\cdots,\bar{v}_{T-1,r},\bar{x}_\star\}} \lambda^{\mathrm{Brd}}_{\bar{v}_{t,k},u}(\bar{\mathrm{dir}}_{t,k}\odot(\bar{v}_{t,k}-u)) \\
&- \sum_{t=1}^{T}\sum_{s=1}^{t-1}\sum_{k=1}^{r}\gamma_{t,s,k}(\bar{g}_t\odot\bar{v}_{s,k}) + \sum_{t=1}^T(\bar{g}_t\odot\bar{x}_\star), \\
\text{and } \lambda = (\lambda^{\mathrm{Lip}},\lambda^{\mathrm{Diam}},\lambda^{\mathrm{Brd}}) &\text{ and } \bar{\mathrm{dir}}_{t,k} = \sum_{s=1}^t \eta_{t,k,s}\,\bar{g}_s + \sum_{(s,j)<_{\mathrm{lex}}(t,k)} \beta_{t,k,s,j}\,\bar{v}_{s,j} \text{ for } t=1,\ldots,T-1.
\end{aligned} \tag{42}$$

Substituting (42) in (41), we get a linear optimization problem with a bilinear matrix inequality constraint, which we reformulate as a linear convex semidefinite program using a change of variables similar to that of (27): $(\eta,\beta,\lambda^{\mathrm{Brd}})$ are replaced by the variables (with the convention $\bar{v}_{T,1}=\bar{x}_\star$):

$$\begin{aligned}
B_{t,k,s} &= \eta_{t,k,s}\sum_{(t,k)<_{\mathrm{lex}}(m,j)\leqslant_{\mathrm{lex}}(T,1)} \lambda^{\mathrm{Brd}}_{\bar{v}_{t,k},\bar{v}_{m,j}} - \sum_{(s,1)\leqslant_{\mathrm{lex}}(m,i)<_{\mathrm{lex}}(t,k)} \eta_{m,i,s}\,\lambda^{\mathrm{Brd}}_{\bar{v}_{m,i},\bar{v}_{t,k}} \\
&\quad \forall (1,1)\leqslant_{\mathrm{lex}}(s,j)\leqslant_{\mathrm{lex}}(t,k)\leqslant_{\mathrm{lex}}(T,1), \\
C_{t,k,s,j} &= \beta_{t,k,s,j}\sum_{(t,k)<_{\mathrm{lex}}(m,i)\leqslant_{\mathrm{lex}}(T,1)} \lambda^{\mathrm{Brd}}_{\bar{v}_{t,k},\bar{v}_{m,i}} - \sum_{(s,j)\leqslant_{\mathrm{lex}}(m,i)<_{\mathrm{lex}}(t,k)} \beta_{m,i,s,j}\,\lambda^{\mathrm{Brd}}_{\bar{v}_{m,i},\bar{v}_{t,k}} \\
&\quad \forall (1,1)\leqslant_{\mathrm{lex}}(s,j)<_{\mathrm{lex}}(t,k)\leqslant_{\mathrm{lex}}(T,1).
\end{aligned} \tag{43}$$

Finally, this change of variables gives us the following linear convex semidefinite program which is the adaptation of (28) for (9):

$$\begin{aligned}
\inf_{\{(B_{t,s,k}, C_{t,s,k}, \gamma_{t,s,k})\}_{t,s,k}} \inf_{\substack{\lambda^{\mathrm{Lip}} \geqslant 0 \\ \lambda^{\mathrm{Diam}} \geqslant 0 \\ \lambda^{\mathrm{Brd}} \geqslant 0}} & \sum_{t=1}^{T} \lambda_t^{\mathrm{Lip}} L^2 + \frac{1}{2} \sum_{u,\, v \in \{\bar{x}_1, \bar{v}_{1,1}, \cdots, \bar{v}_{T-1,r}, \bar{x}_\star\}} \lambda_{\{u,v\}}^{\mathrm{Diam}} D^2 \\
\text{subject to: } & S(B, C, \gamma; \lambda) \succcurlyeq 0, \\
& \sum_{(s,1) \leqslant_{\mathrm{lex}} (t,k) \leqslant_{\mathrm{lex}} (T,1)} B_{t,k,s} = 0 \text{ for } s = 1, \ldots, T, \\
& \sum_{(s,j) <_{\mathrm{lex}} (t,k) \leqslant_{\mathrm{lex}} (T,1)} C_{t,k,s,j} = 0 \text{ for } (1,1) \leqslant_{\mathrm{lex}} (s,j) <_{\mathrm{lex}} (T,1),
\end{aligned}$$

$$\begin{aligned}
\text{where } S(B, C, \gamma; \lambda) = & \frac{1}{2} \sum_{u,\, v \in \{\bar{x}_1, \bar{v}_{1,1}, \cdots, \bar{v}_{T-1,r}, \bar{x}_\star\}} \lambda_{\{u,v\}}^{\mathrm{Diam}} \left((u - v) \odot (u - v)\right) + \sum_{t=1}^{T} \lambda_t^{\mathrm{Lip}} \left(\bar{g}_t \odot \bar{g}_t\right) \\
& + \sum_{\substack{(1,1) \leqslant_{\mathrm{lex}} (t,k) \leqslant_{\mathrm{lex}} (T,1) \\ 1 \leqslant s \leqslant t}} B_{t,s} \left(\bar{g}_s \odot \bar{v}_{t,k}\right) + \sum_{\substack{(1,1) \leqslant_{\mathrm{lex}} (t,k) \leqslant_{\mathrm{lex}} (T,1) \\ (1,1) \leqslant_{\mathrm{lex}} (s,j) <_{\mathrm{lex}} (t,k)}} C_{t,s} \left(\bar{v}_{s,j} \odot \bar{v}_{t,k}\right) \\
& - \sum_{t=1}^{T} \sum_{s=1}^{t-1} \sum_{k=1}^{r} \gamma_{t,s,k} \left(\bar{g}_t \odot \bar{v}_{s,k}\right) + \sum_{t=1}^{T} (\bar{g}_t \odot \bar{x}_\star),
\end{aligned}$$

and $\lambda = (\lambda^{\mathrm{Lip}}, \lambda^{\mathrm{Diam}}, \lambda^{\mathrm{Brd}})$.

Note that, as in Appendix B.2, this change of variables is not one-to-one: for given values of B and C there exist several possible values of $(\eta, \beta, \lambda^{\mathrm{Brd}})$ satisfying (43). Nevertheless, as in Appendix B.2, imposing $\eta_{t,k,s} = 1$ for all t, k, s, we get a unique solution by solving the part of (43) for λ^{Brd}, and then solving the second part of (43) for β.

B.7 Extra PEP numerical plots: log-log plots for regret rate

In this section, we present the log-log plots version of the numerical experiments of Section 4. Recall that those numerical experiments on worst-case regret of different variations of online Frank–Wolfe-type algorithms were obtained by leveraging the semidefinite programming techniques of Section 2, which allow to compute exact worst-case regret values. For convenience, we first present again Figure 1 with linear scales. In Figure 2, we present the same numerical experiments as in Figure 1 (same computed values) but as log-log plots. Note that the slopes of curves in those log-log plots correspond to the exponents of the rates from the regret upper bounds in Figure 1.

In Figure 3, we present an interpolation of those regret rate exponent for the different computed exact worst-case regret bounds from the numerical experiments of Figure 1. (The theoretical upper bounds are in closed forms and their exponent of 3/4 can easily be read.) To interpolate those regret rate exponent we interpolate the slopes from the log-log plots. That is, for a sequence of tight regret upper bounds B_T for different horizon values $T \in \{T_1, \cdots, T_n\}$, we select some value $T_{\mathrm{ref}} = T_{\lfloor n/2 \rfloor}$, and then for all values of $T \in \{T_1, \cdots, T_n\} \setminus \{T_{\mathrm{ref}}\}$ we compute:

$$(\log(B_T) - \log(B_{T_{\mathrm{ref}}}))/(\log(T) - \log(T_{\mathrm{ref}})). \tag{44}$$

We also include in the top right plot of Figure 3 (the plot dedicated to the anytime variants of OFW) the bound (22) from the proof of Theorem 3.3 to compare exponent convergence speed of the bound for anytime tunings of OFW to the exponent convergence speed for the sum of terms in $s^{-1/4}$ in the bound (22).

C The PEP methodology for OGD and FTRL

This section presents how to leverage the semidefinite programming method presented in Section 2 to compute tight worst-case regret bounds for OGD and FTRL. We use the same assumptions as for OFW: the cost functions ℓ_t are convex and L-Lipschitz and the domain of feasible points $\mathcal{K}$ is convex closed with diameter bounded by D.

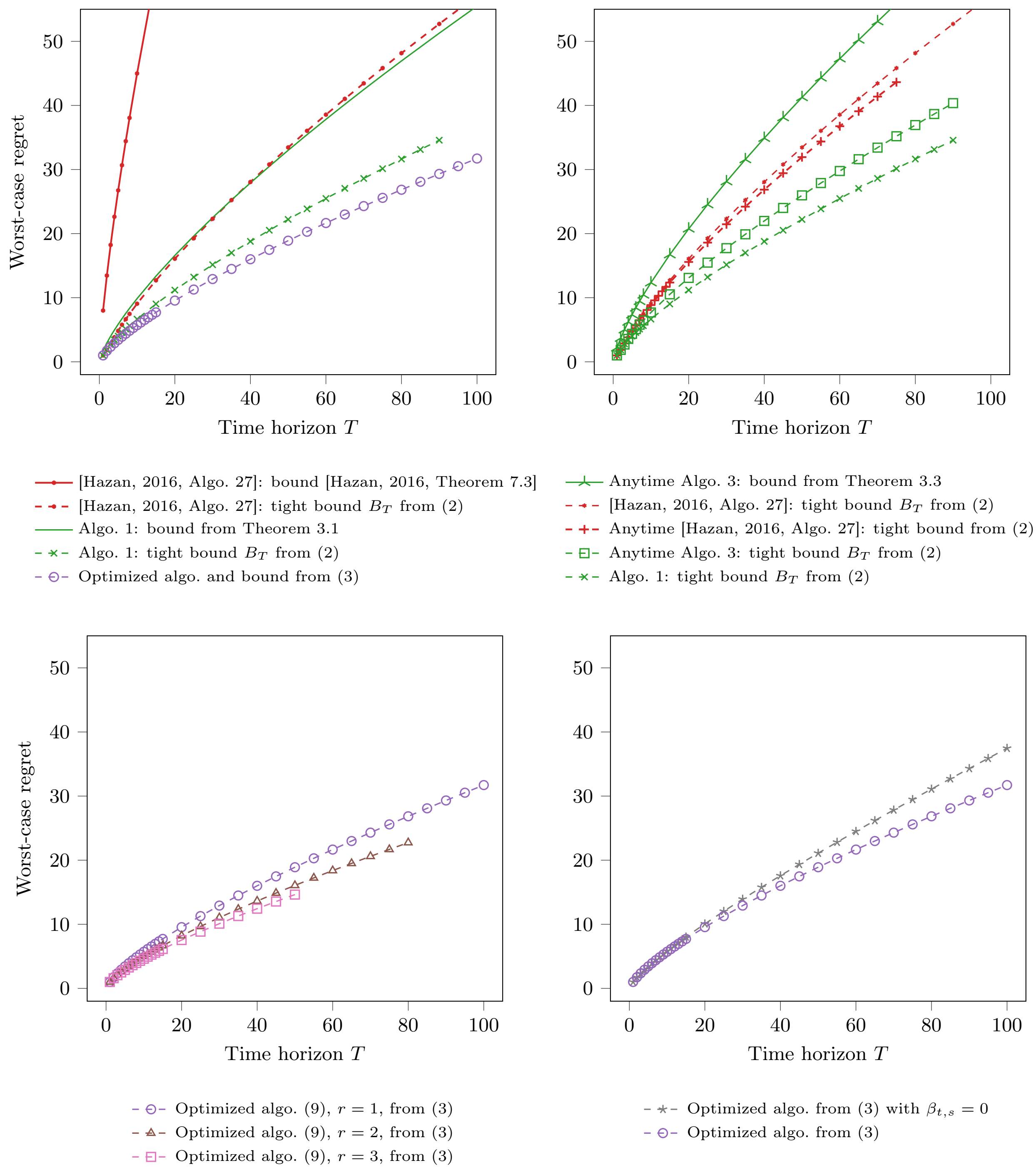

Figure 1: (Top left) Comparison of known upper bounds (respectively from [Hazan, 2016, Theorem 7.3] and Theorem 3.1) against tight numerical bounds (worst-case regrets) obtained from (2), for [Hazan, 2016, Algorithm 27] and Algorithm 1 (parameters from Theorem 3.1). (Top right) Tight numerical regret bounds for [Hazan, 2016, Algorithm 27] and Algorithm 1 (parameters from Theorem 3.1) against their anytime versions. (Bottom left) Tight numerical bounds for optimized online Frank–Wolfe with respectively $r \in \{1, 2, 3\}$ linear optimization steps per time round (where (41) is a variant of (3) with (1) replaced by (9), which we detail in Appendix B.5). (Bottom right) Tight numerical regret bounds for optimized online Frank–Wolfe with and without regularization (i.e., (3) with and without $\beta_{t,s} = 0$). (repeated from page 8)

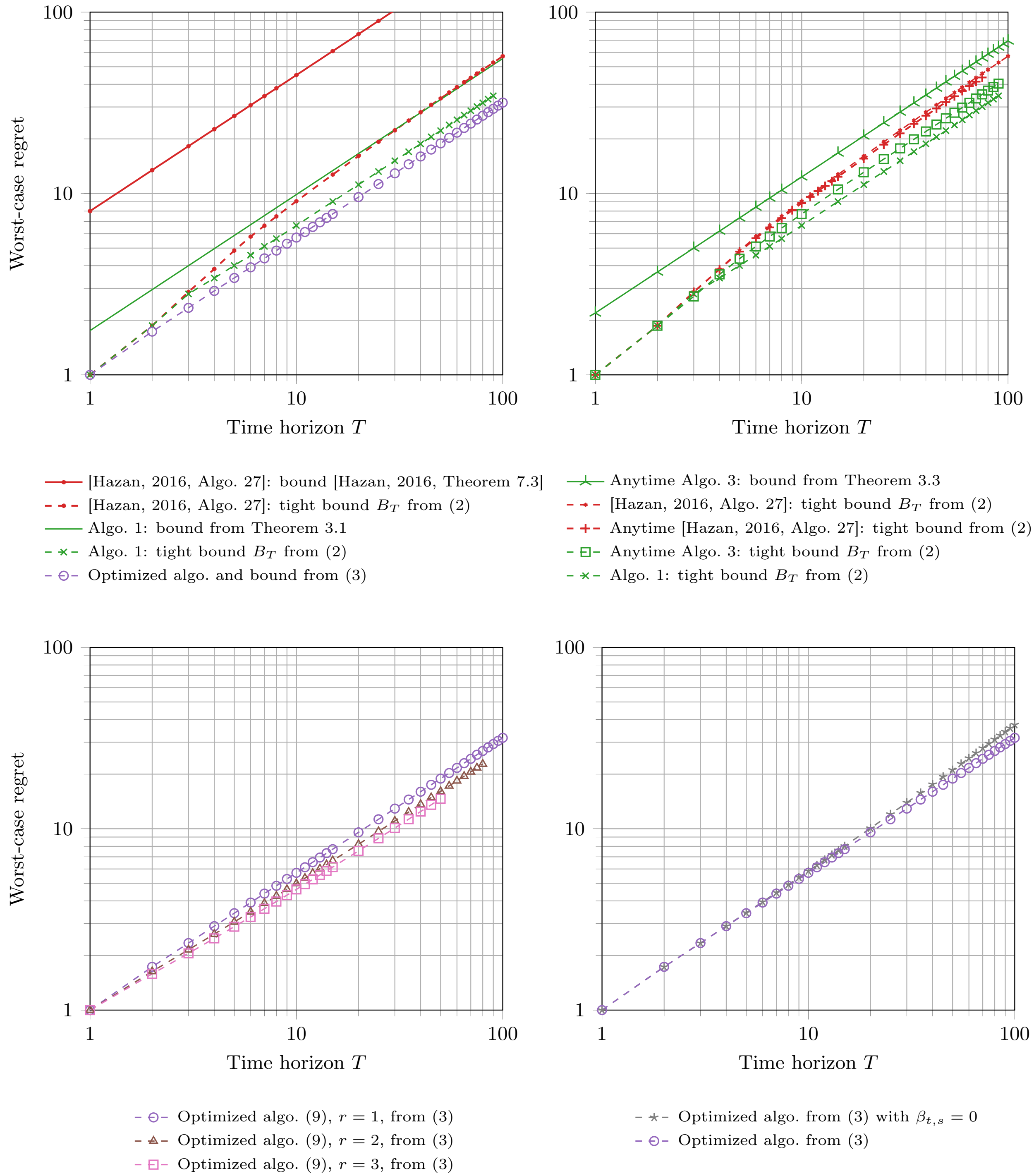

Figure 2: Variant of Figure 1 with log-log plots. (Top left) Comparison of known upper bounds (respectively from [Hazan, 2016, Theorem 7.3] and Theorem 3.1) against tight numerical bounds (worst-case regrets) obtained from (2), for [Hazan, 2016, Algorithm 27] and Algorithm 1 (parameters from Theorem 3.1). (Top right) Tight numerical regret bounds for [Hazan, 2016, Algorithm 27] and Algorithm 1 (parameters from Theorem 3.1) against their anytime versions. (Bottom left) Tight numerical bounds for optimized online Frank–Wolfe with respectively $r \in \{1, 2, 3\}$ linear optimization steps per time round (where (41) is a variant of (3) with (1) replaced by (9), which we detail in Appendix B.5). (Bottom right) Tight numerical regret bounds for optimized online Frank–Wolfe with and without regularization (i.e., (3) with and without $\beta_{t,s} = 0$).

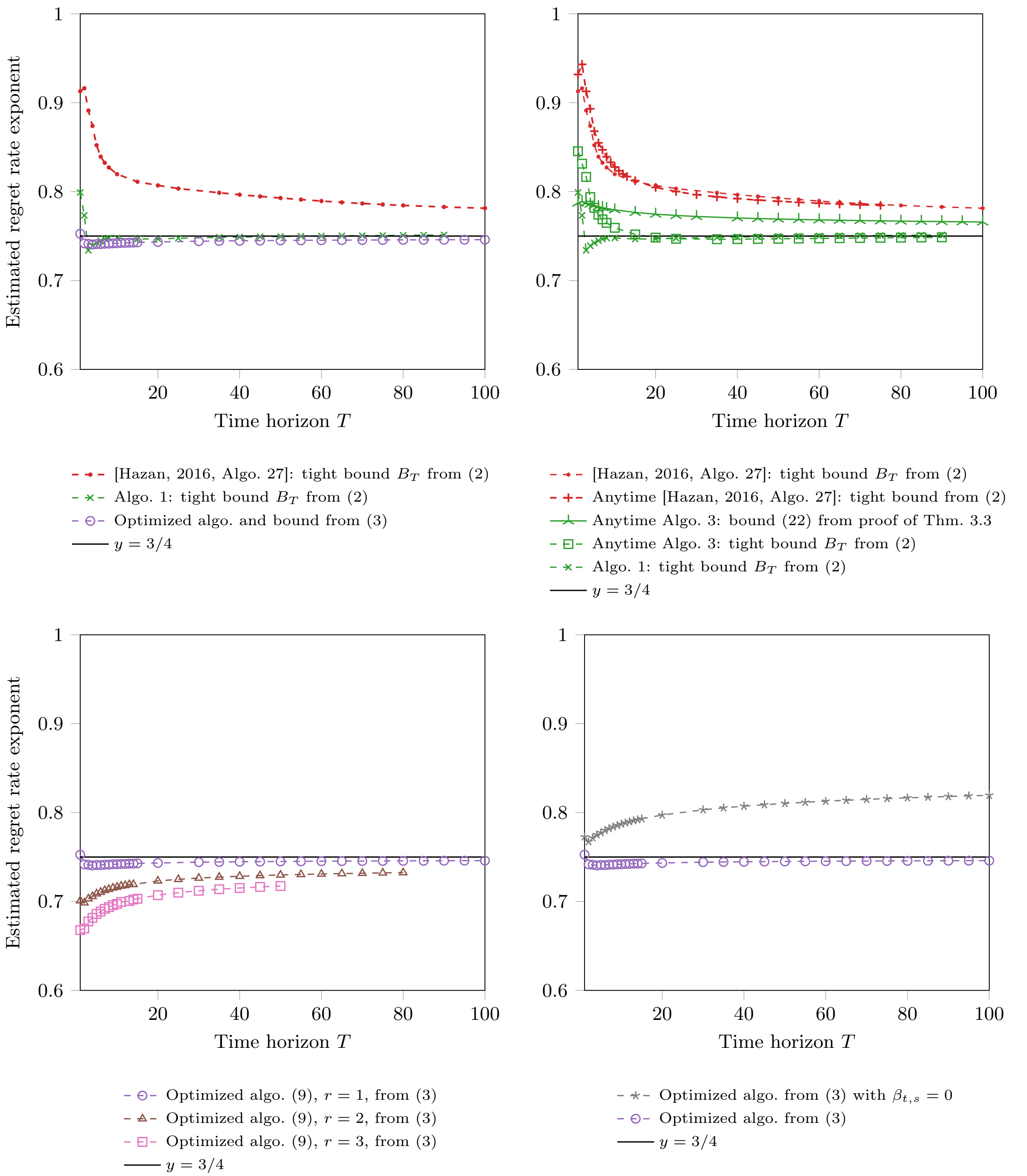

Figure 3: Estimation of regret rate exponents for the tight worst-case regret bounds from Figure 1 by interpolating the slopes of the log-log plots of Figure 2 using (44). (Top left) Tight numerical regret bounds (worst-case regrets) obtained from (2), for [Hazan, 2016, Algorithm 27] and Algorithm 1 (parameters from Theorem 3.1). (Top right) Tight numerical regret bounds for [Hazan, 2016, Algorithm 27] and Algorithm 1 (parameters from Theorem 3.1) against their anytime versions. The bound (22) from the proof of Theorem 3.3 is also included to compare to the speed of convergence for the sum of terms in $s^{-1/4}$ in (22). (Bottom left) Tight numerical bounds for optimized online Frank–Wolfe with respectively $r \in \{1, 2, 3\}$ linear optimization steps per time round (where (41) is a variant of (3) with (1) replaced by (9), which we detail in Appendix B.5). (Bottom right) Tight numerical regret bounds for optimized online Frank–Wolfe with and without regularization (i.e., (3) with and without $\beta_{t,s} = 0$).

C.1 Deriving proof for FTRL from the PEP methodology

We start by studying FTRL with the same regularization parameter for all time steps using the PEP methodology. For convenience, we restate the definition of FTRL with single parameter (note that in this section we will use notations with x for FTRL iterates, whereas we used notations with y for FTRL iterates in the previous sections to distinguish them from OFW iterates).

Algorithm 5 Follow The Regularized Leader (FTRL)

Require: $T \geqslant 1,\ x_1 \in \mathcal{K},\ \eta \geqslant 0$
1: **for** $t = 1$ to T **do**
2: Play x_t, pay cost $\ell_t(x_t)$, observe $g_t = \nabla \ell_t(x_t)$.
3: $x_{t+1} \leftarrow \arg\min_{x\in\mathcal{K}} \eta\langle \sum_{s=1}^{t} g_s, x\rangle + \frac{1}{2}\|x - x_1\|^2$
4: **end for**

Adapting the PEP formulation (2) of OFW to the case of FTRL, we get:

$$\begin{aligned} B_T(\eta) \triangleq \sup_{\substack{\mathcal{K}, \{\ell_t\}_{t\in[\![1,T]\!]} \\ x_\star, \{x_t\}_{t\in[\![1,T]\!]} \\ d\in\mathbb{N}}} \quad & R_T(x_1, \ldots, x_T; x_\star) \\ \text{subject to: } & \ell_t \text{ is convex and } L\text{-Lipschitz for } t \in [\![1, T]\!], \\ & \mathcal{K} \text{ is a non-empty closed convex set of } \mathbb{R}^d, \\ & \mathrm{Diam}(\mathcal{K}) \leqslant D, \\ & \{x_t\}_{t=1,\ldots,T} \text{ is generated by Algorithm 5.} \end{aligned} \tag{45}$$

We now show how to use (45) to derive a proof that $B_T(\eta)$ is an upper bound of the regret for FTRL.

Lemma C.1. *Let $T \geqslant 1$. Assume that the cost functions ℓ_t are convex and L-Lipschitz for all $t \in [\![1, T]\!]$, and that the convex closed domain $\mathcal{K}$ of feasible points has a diameter bounded by D. Then, for any $x_\star \in \mathcal{K}$, the following upper bound on the regret of the FTRL Algorithm 2 holds:*

$$R_T(x_1, \cdots, x_T; x_\star) \leqslant \frac{\eta}{2}\sum_{t=1}^{T} \|g_t\|^2 + \frac{1}{2\eta}\|x_\star - x_1\|^2.$$

In particular, for $\eta = D/(L\sqrt{T})$, we get that the regret is upper bounded by $DL\sqrt{T}$.

Note that the tight regret upper bound from Lemma C.1 is not new, it can be found, e.g., in [Orabona, 2019, Corollary 7.9]. Nevertheless, the proof of Lemma C.1 will allow us to see how to easily find simple clean proofs for regret upper bounds using the PEP methodology.

Remark C.2. Also note that it is important to use the regularization in FTRL with respect to x_1 (or any point in $\mathcal{K}$), as if we were to regularize with respect to an arbitrary point we would have the trivial upper bound LDT for the regret (as FTRL can be equivalently defined as online mirror descent with greedy/lazy updates, this can be seen as having the feasible set $\mathcal{K}$ far from the reference regularization point, say the origin, and thus projection of gradient descent iterates on $\mathcal{K}$ would always land on the same point opposite of the optimum).

We start by explaining how to use (45) to infer the proof of Lemma C.1, and then we will do the proof itself.

We first rewrite (45) as an SDP using a similar argument to that of Section 2.1. We sample function values and gradients for the convex loss functions ℓ_t and the indicator function $i_{\mathcal{K}}$ of the closed convex set $\mathcal{K}$ of feasible points, and then use the interpolation / extension theorems [Taylor et al., 2017, Theorem 3.3 and Equation (7)] and [Taylor et al., 2017, Theorem 3.6], respectively. For all $t \in [\![1, T]\!]$, we denote (f_t, g_t) and $(f_t^\star, g_t^\star)$ by the function values and gradients for the loss function ℓ_t at points x_t and $x_\star$, respectively. We denote by $s_t \in \partial i_{\mathcal{K}}(x_t)$ (resp. $s_\star \in \partial i_{\mathcal{K}}(x_\star)$) a sub-gradient of the indicator function $i_{\mathcal{K}}$ at point x_t for $t = 1, \ldots, T$ (resp. at point $x_\star$). Hence, the upper bound on the regret for FTRL given by (45) can be reformulated as the following finite dimensional program:

$$
\begin{aligned}
\sup_{\substack{x_\star,\{x_t\}_{t=1,\ldots,T}\\ \{(f_t,f_t^\star)\}_{t=1,\ldots,T}\\ \{(g_t,g_t^\star)\}_{t=1,\ldots,T}\\ s_\star,\{s_t\}_{t=1,\ldots,T}\\ d\in\mathbb{N}}} \quad & \sum_{t=1}^{T} f_t - f_t^\star \\
\text{subject to: } & f_t \geqslant f_t^\star + \langle g_t^\star, x_t - x_\star\rangle \text{ for } t=1,\ldots,T, \\
& f_t^\star \geqslant f_t + \langle g_t, x_\star - x_t\rangle \text{ for } t=1,\ldots,T, \\
& \|g_t\| \leqslant L \text{ and } \|g_t^\star\| \leqslant L \text{ for } t=1,\ldots,T, \\
& \|x_t - x_\star\| \leqslant D, \text{ for } t=1,\ldots,T, \\
& \|x_i - x_j\| \leqslant D, \text{ for } i,j=1,\ldots,T, \\
& \langle s_t, x_\star - x_t\rangle \leqslant 0 \text{ and } \langle s_\star, x_t - x_\star\rangle \leqslant 0 \text{ for } t=1,\ldots,T, \\
& \langle s_i, x_j - x_i\rangle \leqslant 0 \text{ for } i,j=1,\ldots,T, \\
& (x_t - x_1) + \eta\sum_{i=1}^{t-1} g_i + s_t = 0 \text{ for } t=1,\ldots,T.
\end{aligned}
\tag{46}
$$

Note that the last constraint corresponds to the optimality condition of FTRL at each time t.

Moreover, as the values of the gradient $g_t^\star$ are never used, we can impose $g_t^\star = g_t$ for all t, which gives a tight relaxation of the problem. Indeed, this corresponds to consider only linear cost functions. In particular, we get that among the worst-case instances, there is a worst-case instance involving only linear cost functions. Hence, we get the following simpler reformulation of (45):

$$
\begin{aligned}
\sup_{\substack{x_\star,\{x_t\}_{t=1,\ldots,T}\\ \{g_t\}_{t=1,\ldots,T}\\ s_\star,\{s_t\}_{t=1,\ldots,T}\\ d\in\mathbb{N}}} \quad & \sum_{t=1}^{T} \langle g_t, x_t - x_\star\rangle \\
\text{subject to: } & \|g_t\| \leqslant L \text{ for } t=1,\ldots,T, \\
& \|x_t - x_\star\| \leqslant D, \text{ for } t=1,\ldots,T, \\
& \|x_i - x_j\| \leqslant D, \text{ for } i,j=1,\ldots,T, \\
& \langle s_t, x_\star - x_t\rangle \leqslant 0 \text{ and } \langle s_\star, x_t - x_\star\rangle \leqslant 0 \text{ for } t=1,\ldots,T, \\
& \langle s_i, x_j - x_i\rangle \leqslant 0 \text{ for } i,j=1,\ldots,T, \\
& (x_t - x_1) + \eta\sum_{i=1}^{t-1} g_i + s_t = 0 \text{ for } t=1,\ldots,T.
\end{aligned}
\tag{47}
$$

We now reformulate this program as an SDP. Define the matrix:

$$P = [x_\star | x_1 | \cdots | x_T | g_1 | \cdots | g_T | s_\star],$$

and let $G = P^{\mathrm{T}}P \succcurlyeq 0$ denote the Gram matrix containing all dots products of those vectors, which is of dimension $(2T+2)\times(2T+2)$. Let $\{\bar{x}_t\}_{t=1,\ldots,T}$ be the vector in $\mathbb{R}^{2T+2}$ such that $\bar{x}_i^{\mathrm{T}} G \bar{x}_j = \langle x_i, x_j\rangle$, and similarly for the other pair of variables. Remark that we do not include the variables $\{s_t\}_{t=1,\ldots,T}$ in P as they are redundant due to the last constraint in (47). Instead, we just define vectors $\{\bar{s}_t\}_{t=1,\ldots,T}$ such that $\bar{s}_t = -(\bar{x}_t - \bar{x}_1) - \eta\sum_{i=1}^{t-1}\bar{g}_t$ for all t. This allows us to redefine (47) as the following equivalent SDP:

$$
\begin{aligned}
\sup_{G\succcurlyeq 0} \quad & \sum_{t=1}^{T} \bar{g}_t^{\mathrm{T}} G(\bar{x}_t - \bar{x}_\star) \\
\text{subject to: } & \bar{g}_t^{\mathrm{T}} G \bar{g}_t \leqslant L \text{ for } t=1,\ldots,T, \\
& (\bar{x}_t - \bar{x}_\star)^{\mathrm{T}} G(\bar{x}_t - \bar{x}_\star) \leqslant D, \text{ for } t=1,\ldots,T, \\
& (\bar{x}_i - \bar{x}_j)^{\mathrm{T}} G(\bar{x}_i - \bar{x}_j) \leqslant D, \text{ for } i,j=1,\ldots,T, \\
& \bar{s}_t^{\mathrm{T}} G(\bar{x}_\star - \bar{x}_t) \leqslant 0 \text{ and } \bar{s}_\star^{\mathrm{T}} G(\bar{x}_t - \bar{x}_\star) \leqslant 0 \text{ for } t=1,\ldots,T, \\
& \bar{s}_i^{\mathrm{T}} G(\bar{x}_j - \bar{x}_i) \leqslant 0 \text{ for } i,j=1,\ldots,T.
\end{aligned}
\tag{48}
$$

Note that the variable $d \in \mathbb{N}$ in (47) imposes that the rank of G is upper bounded by d in (48), but this condition disappear when taking the supremum other $d \in \mathbb{N}$.

The dual variables associated to each constraints in (48) indicate which of those inequalities are used in deriving a proof of the upper bound on the regret of FTRL. Choosing $\eta = D/(L\sqrt{T})$, and running numerical simulations for moderate values of T indicates that some dual variables have constantly small values (several orders smaller than other dual variables). Hence, relaxing the constraints associated to those dual variables with small values gives us a new upper bound on the regret (indeed, with same value) via the following SDP:

$$\begin{aligned} \sup_{G \succcurlyeq 0} \; & \sum_{t=1}^{T} \bar{g}_t^{\mathrm{T}} G(\bar{x}_t - \bar{x}_\star) \\ \text{subject to: } & \bar{g}_t^{\mathrm{T}} G \bar{g}_t \leqslant L \text{ for } t = 1, \ldots, T, \\ & (\bar{x}_1 - \bar{x}_\star)^{\mathrm{T}} G(\bar{x}_1 - \bar{x}_\star) \leqslant D, \\ & \bar{s}_t^{\mathrm{T}} G(\bar{x}_\star - \bar{x}_T) \leqslant 0, \\ & \bar{s}_t^{\mathrm{T}} G(\bar{x}_{t+1} - \bar{x}_t) \leqslant 0 \text{ for } t = 1, \ldots, T-1. \end{aligned} \tag{49}$$

Then, running numerical simulations for (49) with small values of T, we observe that most dual variables take the same values, thus we group them accordingly and relax those constraints by summing them, which gives us a new upper bound on the regret (indeed, with same value) via the following SDP:

$$\begin{aligned} \sup_{G \succcurlyeq 0} \; & \sum_{t=1}^{T} \bar{g}_t^{\mathrm{T}} G(\bar{x}_t - \bar{x}_\star) \\ \text{subject to: } & \sum_{t=1}^{T} \bar{g}_t^{\mathrm{T}} G \bar{g}_t \leqslant LT, \\ & (\bar{x}_1 - \bar{x}_\star)^{\mathrm{T}} G(\bar{x}_1 - \bar{x}_\star) \leqslant D, \\ & \bar{s}_t^{\mathrm{T}} G(\bar{x}_\star - \bar{x}_T) + \sum_{t=1}^{T-1} \bar{s}_t^{\mathrm{T}} G(\bar{x}_{t+1} - \bar{x}_t) \leqslant 0. \end{aligned} \tag{50}$$

We now observe that the values of the dual variables for those three remaining constraints are respectively $\eta/2$, $1/(2\eta)$ and $1/\eta$. Looking at the two constraints involving constants L and D, we can already infer that the upper bound on the regret will be:

$$\frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2 + \frac{1}{2\eta} \|x_\star - x_1\|^2,$$

which is upper bounded by $LD\sqrt{T}$ for $\eta = D/(L\sqrt{T})$. From (50), we now that to prove this upper bound, we only need to use the third constraint inequality (times $1/\eta$) and some scalar product / euclidean norm inequalities (which do not include any information on the relation between vectors). In general, finding which scalar product inequalities to use can be done by studying the Cholesky decomposition of the matrix dual variable for the SDP constraint (see matrix S in (dual-SDP-generic), and see Appendix B.3 for an explanation on how we used the Cholesky decomposition in the case of OFW). In this case, the scalar product inequalities to use are rather simple to guess, and thus we will not need to look at this dual matrix variable.

Hence, we have all the ingredients and we are now ready to do the proof of Lemma C.1.

Proof of Lemma C.1. For simplicity, we write $x_{T+1} = x_\star$. We start by summing the boundary inequalities $\langle s_t, x_{t+1} - x_t \rangle \leqslant 0$ for $t \in [\![1, T]\!]$. As we have:

$$\begin{aligned} \bar{s}_t^{\mathrm{T}} G(\bar{x}_\star - \bar{x}_T) + \sum_{t=1}^{T-1} \bar{s}_t^{\mathrm{T}} G(\bar{x}_{t+1} - \bar{x}_t) &= \sum_{t=1}^{T} \langle s_t, x_{t+1} - x_t \rangle \\ &= -\sum_{t=1}^{T} \langle x_t - x_1, x_{t+1} - x_t \rangle - \eta \sum_{t=1}^{T} \sum_{i=1}^{t-1} \langle g_i, x_{t+1} - x_t \rangle \\ &= -\sum_{t=1}^{T} \langle x_t - x_1, x_{t+1} - x_t \rangle - \eta \sum_{i=1}^{T-1} \langle g_i, x_{T+1} - x_{i+1} \rangle. \end{aligned}$$

this sum of inequalities gives the following inequality (which corresponds to the third inequality in (50)):

$$\eta \sum_{t=1}^{T-1} \langle g_t, x_{t+1} - x_\star \rangle \leqslant \sum_{t=1}^{T-1} \langle x_t - x_1, x_{t+1} - x_t \rangle + \langle x_T - x_1, x_\star - x_T \rangle.$$

Now, we use this inequality and the inequality $\langle u, v \rangle \leqslant \frac{\eta}{2}\|u\|^2 + \frac{1}{2\eta}\|v\|^2$, as well as the convexity of the cost functions ℓ_t, to upper bound the regret for FTRL:

$$\begin{aligned} R_T(x_1, \ldots, x_T; x_\star) &\leqslant \sum_{t=1}^{T} \langle g_t, x_t - x_\star \rangle \\ &= \sum_{t=1}^{T} \langle g_t, x_t - x_{t+1} \rangle + \sum_{t=1}^{T} \langle g_t, x_{t+1} - x_\star \rangle \\ &\leqslant \sum_{t=1}^{T} \langle g_t, x_t - x_{t+1} \rangle + \frac{1}{\eta} \sum_{t=1}^{T} \langle x_t - x_1, x_{t+1} - x_t \rangle \\ &\leqslant \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2 + \frac{1}{2\eta} \sum_{t=1}^{T} \|x_t - x_{t+1}\|^2 + \frac{1}{\eta} \sum_{t=1}^{T} \langle x_t - x_1, x_{t+1} - x_t \rangle \\ &= \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2 + \frac{1}{2\eta} \sum_{t=1}^{T} \left\{ \|x_{t+1} - x_1\|^2 - \|x_t - x_1\|^2 \right\} \\ &= \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2 + \frac{1}{2\eta} \|x_\star - x_1\|^2, \end{aligned}$$

which concludes the proof. □

C.2 Deriving proof for OGD from the PEP methodology

Now, we turn to OGD, and again use (2) to derive a proof that $B_T(\eta)$ is an upper bound of the regret for OGD when using the same step-size for all times (i.e. $\eta_t = \eta$ for all t) and with static regret.

We now turn to study OGD using the PEP methodology. We consider the following version of OGD with a single horizon-dependent step-size (where $\Pi_\mathcal{K}$ denotes the euclidean projection on the convex closed set $\mathcal{K}$).

Algorithm 6 Online Gradient Descent (OGD)

Require: $T \geqslant 1$, $x_1 \in \mathcal{K}$, $\eta \geqslant 0$
1: **for** $t = 1$ to T **do**
2: Play x_t, pay cost $\ell_t(x_t)$, observe $g_t = \nabla \ell_t(x_t)$.
3: $x_{t+1} \leftarrow \Pi_\mathcal{K}(x_t - \eta g_t) = \arg\min_{x \in \mathcal{K}} \|x - (x_t - \eta g_t)\|^2$
4: **end for**

Adapting the PEP formulation (2) of OFW to the case of OGD, we get:

$$\begin{aligned} B_T(\eta) \triangleq \sup_{\substack{\mathcal{K}, \{\ell_t\}_{t \in [\![1,T]\!]} \\ x_\star, \{x_t\}_{t \in [\![1,T]\!]} \\ d \in \mathbb{N}}} \quad & R_T(x_1, \ldots, x_T; x_\star) \\ \text{subject to:} \quad & \ell_t \text{ is convex and } L\text{-Lipschitz for } t \in [\![1, T]\!], \\ & \mathcal{K} \text{ is a non-empty closed convex set of } \mathbb{R}^d, \\ & \mathrm{Diam}(\mathcal{K}) \leqslant D, \\ & \{x_t\}_{t=1,\ldots,T} \text{ is generated by Algorithm 6.} \end{aligned} \tag{51}$$

We now show how to use (51) to derive a proof that $B_T(\eta)$ is an upper bound of the regret for OGD.

Lemma C.3. *Let $T \geqslant 1$. Assume that the cost functions ℓ_t are convex and L-Lipschitz for all $t \in [\![1, T]\!]$, and that the convex closed domain $\mathcal{K}$ of feasible points has a diameter bounded by D. Then, for any $x_\star \in \mathcal{K}$, the* *following upper bound on the regret of the OGD Algorithm 6 holds:*

$$R_T(x_1, \cdots, x_T; x_\star) \leqslant \frac{\eta}{2}\sum_{t=1}^{T} \|g_t\|^2 + \frac{1}{2\eta}\|x_\star - x_1\|^2.$$

In particular, for $\eta = D/(L\sqrt{T})$, we get that the regret is upper bounded by $DL\sqrt{T}$.

We start by explaining how to use (51) to infer the proof of Lemma C.3, and then we will do the proof itself. As the idea is similar to the case of FTRL, we just give the general idea.

Using a similar argument to that of Section 2.1 for OFW and Appendix C.1, the PEP program (51) for OGD can be reformulated as the following finite dimensional program:

$$\begin{aligned}
\sup_{\substack{x_\star, \{x_t\}_{t=1,\dots,T} \\ \{g_t\}_{t=1,\dots,T} \\ s_\star, \{s_t\}_{t=1,\dots,T} \\ d\in\mathbb{N}}} \quad & \sum_{t=1}^{T} \langle g_t, x_t - x_\star\rangle \\
\text{subject to:} \quad & \|g_t\| \leqslant L \text{ for } t = 1, \dots, T, \\
& \|x_t - x_\star\| \leqslant D, \text{ for } t = 1, \dots, T, \\
& \|x_i - x_j\| \leqslant D, \text{ for } i, j = 1, \dots, T, \\
& \langle s_t, x_\star - x_t\rangle \leqslant 0 \text{ and } \langle s_\star, x_t - x_\star\rangle \leqslant 0 \text{ for } t = 1, \dots, T, \\
& \langle s_i, x_j - x_i\rangle \leqslant 0 \text{ for } i, j = 1, \dots, T, \\
& x_t - (x_{t-1} - \eta g_{t-1}) + s_t = 0 \text{ for } t = 2, \dots, T.
\end{aligned}$$

We then turn this PEP into an SDP using the Gram matrix G associated to the matrix P containing all vectors (without redundancy) as before. We define the vectors $\bar{x}_\star$, $\bar{s}_\star$, $\bar{x}_t$ and $\bar{g}_t$ for $t = 1, \dots, T$ as before. As the optimality constraint is different between OGD and FTRL, here we have $\bar{s}_t = -\bar{x}_t + (\bar{x}_{t-1} - \eta\bar{g}_{t-1})$ for $t = 2, \dots, T$. (Note that the value of s_1 can be set freely to zero in the PEP above, so we can simply set $\bar{s}_1 = 0$.) This allows us to redefine (51) as the following equivalent SDP:

$$\begin{aligned}
\sup_{G\succcurlyeq 0} \quad & \sum_{t=1}^{T} \bar{g}_t^{\mathrm{T}} G(\bar{x}_t - \bar{x}_\star) \\
\text{subject to:} \quad & \bar{g}_t^{\mathrm{T}} G\bar{g}_t \leqslant L \text{ for } t = 1, \dots, T, \\
& (\bar{x}_t - \bar{x}_\star)^{\mathrm{T}} G(\bar{x}_t - \bar{x}_\star) \leqslant D, \text{ for } t = 1, \dots, T, \\
& (\bar{x}_i - \bar{x}_j)^{\mathrm{T}} G(\bar{x}_i - \bar{x}_j) \leqslant D, \text{ for } i, j = 1, \dots, T, \\
& \bar{s}_t^{\mathrm{T}} G(\bar{x}_\star - \bar{x}_t) \leqslant 0 \text{ and } \bar{s}_\star^{\mathrm{T}} G(\bar{x}_t - \bar{x}_\star) \leqslant 0 \text{ for } t = 1, \dots, T, \\
& \bar{s}_i^{\mathrm{T}} G(\bar{x}_j - \bar{x}_i) \leqslant 0 \text{ for } i, j = 1, \dots, T.
\end{aligned} \tag{52}$$

Note that (52) for OGD looks identical to (48) for FTRL, the only (implicit) difference is the definition of the vectors $\bar{s}_t$ for $t = 1, \dots, T$.

Choosing $\eta = D/(L\sqrt{T})$, running numerical simulations of 52 for small values of T, relaxing constraints with small dual variable values and further relaxing by grouping constraints with the same dual variables values, we get a new upper bound on the regret of OGD (indeed, with same value) via the following SDP:

$$\begin{aligned}
\sup_{G\succcurlyeq 0} \quad & \sum_{t=1}^{T} \bar{g}_t^{\mathrm{T}} G(\bar{x}_t - \bar{x}_\star) \\
\text{subject to:} \quad & \sum_{t=1}^{T} \bar{g}_t^{\mathrm{T}} G\bar{g}_t \leqslant LT, \\
& (\bar{x}_1 - \bar{x}_\star)^{\mathrm{T}} G(\bar{x}_1 - \bar{x}_\star) \leqslant D, \\
& \sum_{t=2}^{T} \bar{s}_t^{\mathrm{T}} G(\bar{x}_\star - \bar{x}_t) \leqslant 0.
\end{aligned} \tag{53}$$

We now observe that the values of the dual variables for those three remaining constraints are respectively $\eta/2$, $1/(2\eta)$ and $1/\eta$. Looking at the two constraints involving constants L and D, we can already infer that the upper bound on the regret will be:

$$\frac{\eta}{2}\sum_{t=1}^{T}\|g_t\|^2 + \frac{1}{2\eta}\|x_\star - x_1\|^2,$$

which is upper bounded by $LD\sqrt{T}$ for $\eta = D/(L\sqrt{T})$. From (53), we now that to prove this upper bound, we only need to use the third constraint inequality (times $1/\eta$) and some scalar product / euclidean norm inequalities (which we will easily infer as for FTRL).

Hence, we have all the ingredients and we are now ready to do the proof of Lemma C.3.

Proof. We start by summing the boundary inequalities $\langle s_t, x_\star - x_t\rangle \leqslant 0$ where $s_t = -x_t + (x_{t-1} - \eta g_{t-1})$ for $t \in [\![2, T]\!]$. This gives us the following inequality (which corresponds to the third constraint in (53)):

$$0 \geqslant \sum_{t=2}^{T}\langle s_t, x_\star - x_t\rangle = \sum_{t=2}^{T}\langle x_{t-1} - x_t, x_\star - x_t\rangle - \eta\sum_{t=2}^{T}\langle g_{t-1}, x_\star - x_t\rangle,$$

which we more conveniently rewrite as:

$$\sum_{t=2}^{T}\langle g_{t-1}, x_t - x_\star\rangle \leqslant \frac{1}{\eta}\sum_{t=2}^{T}\langle x_{t-1} - x_t, x_t - x_\star\rangle.$$

Now, we use this inequality and the inequality $\langle u, v\rangle \leqslant \frac{\eta}{2}\|u\|^2 + \frac{1}{2\eta}\|v\|^2$, as well as the convexity of the cost functions ℓ_t, to upper bound the regret for OGD:

$$\begin{aligned}
R_T(x_1, \dots, x_T; x_\star) &\leqslant \sum_{t=1}^{T}\langle g_t, x_t - x_\star\rangle \\
&= \sum_{t=1}^{T}\langle g_t, x_t - x_{t+1}\rangle + \sum_{t=1}^{T-1}\langle g_t, x_{t+1} - x_\star\rangle \\
&= \sum_{t=1}^{T}\langle g_t, x_t - x_{t+1}\rangle + \sum_{t=2}^{T}\langle g_{t-1}, x_t - x_\star\rangle \\
&\leqslant \frac{\eta}{2}\sum_{t=1}^{T}\|g_t\|^2 + \frac{1}{2\eta}\sum_{t=1}^{T}\|x_t - x_{t+1}\|^2 + \frac{1}{\eta}\sum_{t=2}^{T}\langle x_{t-1} - x_t, x_t - x_\star\rangle \\
&\leqslant \frac{\eta}{2}\sum_{t=1}^{T}\|g_t\|^2 + \frac{1}{2\eta}\sum_{t=1}^{T}\|x_t - x_{t+1}\|^2 + \frac{1}{\eta}\sum_{t=1}^{T-1}\langle x_t - x_{t+1}, x_{t+1} - x_\star\rangle \\
&= \frac{\eta}{2}\sum_{t=1}^{T}\|g_t\|^2 + \frac{1}{2\eta}\sum_{t=1}^{T-1}\Big\{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2\Big\} + \frac{1}{2\eta}\|x_T - x^*\|^2 \\
&= \frac{\eta}{2}\sum_{t=1}^{T}\|g_t\|^2 + \frac{1}{2\eta}\|x_1 - x^*\|^2,
\end{aligned}$$

which concludes the proof. ☐

C.3 Adapting the proof for FTRL to Bregman divergences

We now adapt the proof for FTRL of Lemma C.1 to the case of general norm and Bregman divergences. This illustrates the fact that the proof we derived from the PEP (45) is simple and clean enough to easily be adapted to a more general setting.

Let $\|\cdot\|$ denote a general norm, and denote by $\|\cdot\|_*$ its associated dual norm defined by $\|g\|_* = \sup_{x\,:\,\|x\|\leqslant 1}\langle g, x\rangle$. In particular, this definition implies the generalised Cauchy-Schwarz inequality $\langle g, x\rangle \leqslant \|g\|_*\|x\|$. Let ψ be a 1-strongly convex function w.r.t. the norm $\|\cdot\|$, that is:

$$\psi(x) \geqslant \psi(y) + \langle\nabla\psi(y), x - y\rangle + \frac{1}{2}\|x - y\|^2.$$

The Bregman divergence associated to ψ is then defined as:

$$B_\psi(x;y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x-y\rangle.$$

The Bregman divergence $B_\psi(x;y)$ will be used as a regularization function to replace the term $\frac{1}{2}\|x-y\|^2$ used before.

Then, we define the iterates of FTRL (on linearized losses) with Bregman divergence regularization for all t as:

$$x_{t+1} = \operatorname*{arg\,min}_{x\in\mathcal{K}} B_\psi(x;x_1) + \eta\sum_{\tau=1}^{t}\langle \nabla\ell_\tau(x_\tau),\, x\rangle. \tag{54}$$

In particular, the sub-gradients $s_t \in \partial i_{\mathcal{K}}(x_t)$ is defined by the optimality condition:

$$\nabla\psi(x_t) - \nabla\psi(x_1) + \eta\sum_{\tau=1}^{t-1} g_\tau + s_t = 0. \tag{55}$$

We can now state the following lemma upper bounding the regret of FTRL with Bregman divergence, and whose proof is an adaptation of the proof of Lemma C.1.

Lemma C.4. *Let $T \geqslant 1$. Assume that the cost functions ℓ_t are convex and L-Lipschitz for all $t \in [\![1,T]\!]$, and that the convex closed domain $\mathcal{K}$ of feasible points is such that $\sup_{x,y\in\mathcal{K}} B_\psi(x;y) \leqslant \frac{1}{2}D^2$ for some D. Then, for any $x_\star \in \mathcal{K}$, the following upper bound on the regret of the FTRL Algorithm defined by (54) holds:*

$$R_T(x_1,\cdots,x_T;x_\star) \leqslant \frac{\eta}{2}\sum_{t=1}^{T}\|g_t\|_*^2 + \frac{1}{\eta}B_\psi(x_\star;x_1).$$

In particular, for $\eta = D/(L\sqrt{T})$, we get that the regret is upper bounded by $DL\sqrt{T}$.

Proof. For simplicity, we write $x_{T+1} = x_\star$. As in the proof of Lemma C.1, we start by summing the boundary inequalities $\langle s_t, x_{t+1}-x_t\rangle \leqslant 0$ for $t \in [\![1,T]\!]$ where s_t is defined in (55). This gives us:

$$\begin{aligned}
0 &\geqslant \sum_{t=1}^{T}\langle s_t, x_{t+1}-x_t\rangle \\
&= -\sum_{t=1}^{T}\langle \nabla\psi(x_t)-\nabla\psi(x_1), x_{t+1}-x_t\rangle - \eta\sum_{t=1}^{T}\sum_{i=1}^{t-1}\langle g_i, x_{t+1}-x_t\rangle \\
&= -\sum_{t=1}^{T}\langle \nabla\psi(x_t)-\nabla\psi(x_1), x_{t+1}-x_t\rangle - \eta\sum_{i=1}^{T-1}\langle g_i, x_{T+1}-x_{i+1}\rangle.
\end{aligned}$$

We then rewrite this inequality as:

$$\sum_{t=1}^{T-1}\langle g_t, x_{t+1}-x_\star\rangle \leqslant \frac{1}{\eta}\sum_{t=1}^{T}\langle \nabla\psi(x_t)-\nabla\psi(x_1), x_{t+1}-x_t\rangle.$$

We will need the following inequality, which comes directly from rewriting the definition of 1-strong convexity of ψ:

$$\frac{1}{2}\|x-y\|^2 \leqslant \psi(x)-\psi(y)-\langle\nabla\psi(y), x-y\rangle.$$

Now, we use those inequalities and the inequality:

$$\langle g, x\rangle \leqslant \|g\|_*\|x\| \leqslant \frac{\eta}{2}\|g\|_*^2 + \frac{1}{2\eta}\|x\|^2,$$

as well as the convexity of the cost functions ℓ_t, to upper bound the regret for FTRL (writing R_T for

$R_T(x_1, \ldots, x_T; x_\star))$:

$$
\begin{aligned}
R_T &\leqslant \sum_{t=1}^{T} \langle g_t, x_t - x_\star \rangle \\
&= \sum_{t=1}^{T} \langle g_t, x_t - x_{t+1} \rangle + \sum_{t=1}^{T} \langle g_t, x_{t+1} - x_\star \rangle \\
&\leqslant \sum_{t=1}^{T} \langle g_t, x_t - x_{t+1} \rangle + \frac{1}{\eta} \sum_{t=1}^{T} \langle \nabla\psi(x_t) - \nabla\psi(x_1), x_{t+1} - x_t \rangle \\
&\leqslant \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|_*^2 + \frac{1}{2\eta} \sum_{t=1}^{T} \|x_t - x_{t+1}\|^2 + \frac{1}{\eta} \sum_{t=1}^{T} \langle \nabla\psi(x_t) - \nabla\psi(x_1), x_{t+1} - x_t \rangle \\
&\leqslant \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|_*^2 + \frac{1}{\eta} \sum_{t=1}^{T} \Big\{ \psi(x_{t+1}) - \psi(x_t) - \langle \nabla\psi(x_t), x_{t+1} - x_t \rangle \Big\} + \frac{1}{\eta} \sum_{t=1}^{T} \langle \nabla\psi(x_t) - \nabla\psi(x_1), x_{t+1} - x_t \rangle \\
&= \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|_*^2 + \frac{1}{\eta} \Big\{ \psi(x_\star) - \psi(x_1) - \langle \nabla\psi(x_1), x_\star - x_1 \rangle \Big\} \\
&= \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|_*^2 + \frac{1}{\eta} B_\psi(x_\star; x_1),
\end{aligned}
$$

which concludes the proof. □

C.4 Generic SDP for PEP

In this section, we present the following generic form of an SDP arising from a PEP.

$$
\begin{aligned}
&\sup_{\substack{G \succcurlyeq 0 \\ F}} \ \mathrm{tr}(A_{\mathrm{obj}} G) + \langle a_{\mathrm{obj}}, F \rangle \\
&\text{subject to: } \ \mathrm{tr}(A_i G) + \langle a_i, F \rangle \leqslant b_i, \ \text{for } i = 1, \ldots, N.
\end{aligned}
\tag{SDP-generic}
$$

The semidefinite positive matrix variable G corresponds to the Gram matrix of the gradients and points, and the vector variable F corresponds to the sampled function values. Note that in our case, the variable F can be removed as we get a tight relaxation of the PEP by considering only linear cost functions ℓ_t.

We then form the (generic form) Lagrangian function associated to the SDP (SDP-generic) by dualizing all the constraints except the semidefinite positive constraint $G \succcurlyeq 0$.

$$
\begin{aligned}
L(G, F; \lambda_{1:N}) &= \mathrm{tr}(A_{\mathrm{obj}} G) + \langle a_{\mathrm{obj}}, F \rangle - \sum_{i=1}^{N} \lambda_i \Big[\mathrm{tr}(A_i G) + \langle a_i, F \rangle - b_i \Big] \\
&= \sum_{i=1}^{N} b_i \lambda_i + \mathrm{tr}\left(\left(A_{\mathrm{obj}} - \sum_{i=1}^{N} \lambda_i A_i \right) G \right) + \left\langle a_{\mathrm{obj}} - \sum_{i=1}^{N} \lambda_i a_i, F \right\rangle
\end{aligned}
$$

By maximizing this Lagrangian function over the primal variables G and F, we can form the Lagrange dual problem to the generic form SDP (SDP-generic) which is also an SDP.

$$
\begin{aligned}
&\inf_{\lambda_i \geqslant 0, i=1,\ldots,N} \ \sum_{i=1}^{N} b_i \lambda_i \\
&\text{subject to: } \ a_{\mathrm{obj}} = \sum_{i=1}^{N} \lambda_i a_i, \\
&\qquad S := A_{\mathrm{obj}} - \sum_{i=1}^{N} \lambda_i A_i \preccurlyeq 0.
\end{aligned}
\tag{dual-SDP-generic}
$$

Note that in this dual SDP, there is a semidefinite positive constraint involving a matrix S which corresponds to the dual variable of the primal semidefinite positive constraint $G \succcurlyeq 0$. By studying the numerical values of the Cholesky decomposition of this matrix S at optimum, we can get numerical insights on the scalar product / euclidean norm inequalities that need to be used in proofs of regret upper bounds.