
An Ellipsoid Algorithm for Online Convex Optimization

Zakaria Mhammedi*
mhammedi@google.com

Abstract

We study the problem of Online Convex Optimization (OCO) over a convex set $\mathcal{K} \subset \mathbb{R}^d$, accessed via a separation oracle. While classical projection-based algorithms such as projected Online Gradient Descent (OGD) achieve the optimal $O(\sqrt{T})$ regret, they require computing Euclidean projections onto \mathcal{K} whenever an iterate falls outside the feasible set. These projections can be computationally expensive, especially for complex or high-dimensional sets. Projection-free algorithms address this by replacing projections with alternative oracle-based procedures, such as separation or linear optimization oracles. However, the regret bounds of existing separation-based methods scale poorly with the set's *asphericity* κ , defined as the ratio between the radii of the smallest enclosing ball and the largest inscribed ball in \mathcal{K} ; for ill-conditioned sets, κ can be arbitrarily large.

We introduce a new separation-based algorithm for OCO that achieves a regret bound of $\tilde{O}(\sqrt{dT} + d^2)$, with only logarithmic dependence on κ . This removes a key limitation of prior work and eliminates the need for costly geometric pre-processing, such as transforming \mathcal{K} into isotropic position. Our algorithm is based on a novel reduction to online optimization over a sequence of dynamically updated ellipsoids, inspired by the classical ellipsoid method for convex optimization. It requires only $\tilde{O}(1)$ separation oracle calls per round, on par with existing separation-based approaches. These advances make our method particularly well suited for online optimization over geometrically complex feasible sets.

1 Introduction

Convex optimization plays a central role in many areas of modern machine learning and related fields. While classical algorithms are designed for settings with a fixed objective function, recent work has increasingly considered the more general framework of Online Convex Optimization (OCO) [10]. OCO models sequential decision-making problems where a possibly different convex function is revealed at each round, potentially chosen in an adversarial manner. This framework captures a broad range of learning scenarios and provides a unified lens for analyzing them. Standard online-to-batch conversion techniques [3, 22, 4] allow regret guarantees in OCO to be translated into convergence rates for offline (fixed-objective) and stochastic convex optimization problems. In contrast, the analysis of traditional offline methods does not always extend to stochastic settings. Even stochastic non-convex optimization has benefited from the OCO framework; [5] showed that stochastic non-convex optimization can be reduced to OCO, and that this reduction leads to the best-known convergence rates for finding stationary points. Given its broad applicability, designing computationally efficient OCO algorithms with strong regret guarantees remains an important direction of research.

In constrained optimization settings where the decision variable must lie in a feasible set \mathcal{K} , the classical (projected) Online Gradient Descent (OGD) [25] requires a Euclidean projection onto \mathcal{K}

*Google Research, NYC.

whenever an iterate falls outside \mathcal{K} . This step can be computationally expensive when \mathcal{K} has a complex structure, and may limit the practicality of OGD. To address this, projection-free algorithms have been proposed. These methods avoid explicit projections; for example, the Frank-Wolfe algorithm [7] uses linear optimization over \mathcal{K} . More recent approaches rely on membership or separation oracles to enforce constraints [18, 8, 17, 9]. In particular, separation oracles can often be more tractable to implement and use in settings where projections are costly.

The aforementioned projection-free algorithms are analyzed in the OCO framework, where at each round t the algorithm selects a point \mathbf{w}_t from a convex feasible set $\mathcal{K} \subset \mathbb{R}^d$. It then incurs a loss $f_t(\mathbf{w}_t)$, where f_t is a convex function that may be chosen adversarially. The goal is to ensure a small regret $\sup_{\mathbf{u} \in \mathcal{K}} \sum_{t=1}^T (f_t(\mathbf{w}_t) - f_t(\mathbf{u}))$. While projected OGD achieves the optimal $O(\sqrt{T})$ regret, projection-free methods based on linear optimization oracles have not improved beyond $O(T^{3/4})$ [12] without additional structural assumptions on the OCO problem. Separation oracle-based methods proposed by [18, 8] achieve the optimal T -dependence with a $O(\kappa\sqrt{T})$ regret bound, where $\kappa := R/r$ denotes the set's *asphericity*, defined as the ratio of the radii of the smallest enclosing and largest inscribed balls in \mathcal{K} . However, the linear dependence on κ is not desirable, as κ can be arbitrarily large for ill-conditioned sets. Although preprocessing \mathcal{K} into isotropic position can ensure $\kappa \leq d$ [6], this transformation is computationally expensive—requiring up to $\tilde{O}(d^4)$ separation oracle calls [16]—and does not fully resolve the issue: the dependence on κ may reappear implicitly through changes in the Lipschitz constant of the loss functions after the isotropic reparametrization. A more recent method [19] improves this by achieving a regret bound of $\tilde{O}(\sqrt{dT} + \kappa d)$, making the κ term independent of T . However, a residual dependence on the potentially large asphericity remains.

Contributions. This paper introduces a new projection-free algorithm for OCO based on separation oracles. Our key contribution is an algorithm that achieves a regret bound with only logarithmic dependence on the feasible set's asphericity κ . Specifically, it guarantees a regret of $\tilde{O}(\sqrt{dT} + d^2)$ without requiring expensive geometric preprocessing of the set \mathcal{K} . Thus, this approach is more practical and robust to the feasible set's ill-conditioning compared to prior separation-based methods.

The algorithm preserves the oracle complexity of existing separation-based approaches, requiring only $\tilde{O}(1)$ separation oracle calls per round. The total computational cost per iteration is $\tilde{O}(C_{\text{sep}} + d^\omega)$, where C_{sep} denotes the cost of a single separation oracle call and ω is the matrix multiplication exponent. Our approach integrates ideas from the classical ellipsoid method with a reduction to online exp-concave optimization over a sequence of dynamically updated ellipsoids. A slight modification of the standard online-to-batch conversion yields a convergence rate of $\tilde{O}(\sigma\sqrt{d/T} + d^2/T)$ for Stochastic Convex Optimization (SCO), where σ^2 denotes the gradient variance, and a rate of $\tilde{O}(d^2/T)$ in the offline setting ($\sigma = 0$). A detailed comparison with prior results is presented in Table 1.

Limitations. We do not provide regret lower bounds or experimental validation for our algorithm.

Table 1: Comparison of projection-free algorithms for OCO and Stochastic Convex Optimization (SCO). $\kappa = R/r$ is the asphericity of the feasible set \mathcal{K} . σ^2 is the variance in SCO.

Papers	Regret bound in OCO	Convergence rate in SCO ($\sigma \geq 0$)	Oracle type	Number of oracle calls per round	Runtime per round
[12]	$O(T^{3/4})$	$O\left(\frac{1}{T^{1/3}}\right)$	Linear optimization	1	$C_{\text{linOpt}} + d$
[18, 8]	$O(\kappa\sqrt{T})$	$O\left(\frac{\kappa}{\sqrt{T}}\right)$	Separation	$O(1)\text{--}\tilde{O}(1)$	$C_{\text{sep}} + d$
[19]	$\tilde{O}(\sqrt{dT} + \kappa d)$	$\tilde{O}\left(\sigma\sqrt{\frac{d}{T}} + \frac{\kappa d}{T}\right)$	Separation	$\tilde{O}(1)$	$C_{\text{sep}} + d^2$
This paper	$\tilde{O}(\sqrt{dT} + d^2)$	$\tilde{O}\left(\sigma\sqrt{\frac{d}{T}} + \frac{d^2}{T}\right)$	Separation	$\tilde{O}(1)$	$C_{\text{sep}} + d^\omega$

Related works. This work builds upon the recent line of research on separation oracle-based projection-free algorithms for OCO. The prior art established methods achieving $O(\kappa\sqrt{T})$ [18, 17, 8] and subsequently $\tilde{O}(\sqrt{dT} + \kappa d)$ regret bounds [19]. A key limitation of these results is the dependence on the asphericity κ . Our work directly addresses this limitation by providing an algorithm whose regret guarantee depends on κ only *logarithmically*.

Our algorithmic approach is inspired by the classical ellipsoid method [14, 23, 2], which we adapt to the online optimization setting. While the classical method is designed for offline problems, our algorithm addresses the sequential nature of OCO by combining ellipsoid updates with a reduction to online optimization over a sequence of changing ellipsoids that always contain the feasible set \mathcal{K} . Our reduction shares conceptual links with the one presented in [19], but crucially, our method updates these ellipsoids adaptively to better “approximate” the shape of the feasible set, unlike prior approaches using a fixed ball. This ability to adapt to the feasible set’s geometry on the fly is a key feature distinguishing our method from previous separation-based approaches.

2 Preliminaries

In Section 2.1, we formally introduce the OCO setup along with the notation used throughout the paper. Section 2.2 presents key concepts we rely on such as Gauge distance and Gauge projections.

2.1 Setup and Notation

Let \mathcal{K} be a closed convex subset of the Euclidean space \mathbb{R}^d , where we assume $d \geq 2$ throughout.² We consider the standard framework of OCO over \mathcal{K} , in which an algorithm generates a sequence of decisions $(\mathbf{w}_t)_{t \geq 1}$ within \mathcal{K} . On each round $t \in [T]$, the algorithm selects a point $\mathbf{w}_t \in \mathcal{K}$ and incurs a loss $f_t(\mathbf{w}_t)$, where $f_t : \mathcal{K} \rightarrow \mathbb{R}$ is a convex loss function that may depend adversarially on past decisions. As is standard in this setting, we assume access to a subgradient $\mathbf{g}_t \in \partial f_t(\mathbf{w}_t)$ rather than the full loss function. The performance of the algorithm is measured by its regret after T rounds, defined as $\text{Reg}_T := \sum_{t=1}^T f_t(\mathbf{w}_t) - \inf_{\mathbf{u} \in \mathcal{K}} \sum_{t=1}^T f_t(\mathbf{u})$. Due to the convexity of each f_t , the regret can be upper bounded by the so-called linearized regret: $\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t \rangle - \inf_{\mathbf{u} \in \mathcal{K}} \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{u} \rangle$. Hence, it suffices to control the linearized regret to bound Reg_T .

Building on recent projection-free methods, this work aims to develop an efficient OCO algorithm that achieves sublinear linearized regret with only logarithmic dependence on the asphericity κ , while requiring a logarithmic number of separation oracle calls per round.

Definition 2.1 (Separation oracle). A Separation oracle $\text{Sep}_{\mathcal{C}}$ for a set \mathcal{C} is an oracle that given $\mathbf{u} \in \mathbb{R}^d$ returns $(b, \mathbf{v}) \in \{0, 1\} \times \mathbb{B}(1)$ (where $\mathbb{B}(1)$ denotes the unit Euclidean ball in \mathbb{R}^d), such that

- $b = 0$ and $\mathbf{v} = \mathbf{0}$, if $\mathbf{u} \in \mathcal{C}$; and otherwise,
- $b = 1$ and $\langle \mathbf{v}, \mathbf{u} \rangle > \langle \mathbf{v}, \mathbf{w} \rangle$, for all $\mathbf{w} \in \mathcal{C}$.

We denote by $C_{\text{sep}}(\mathcal{C})$ the computational cost of one call to this oracle.

In addition to the standard OCO setup, we impose two common assumptions. To present these assumptions, we let $\|\cdot\|$ denote the Euclidean norm and use $\mathbb{B}(\mathbf{c}, \gamma) \subset \mathbb{R}^d$ to denote the ball of radius $\gamma > 0$ centered at $\mathbf{c} \in \mathbb{R}^d$. With this, the first assumption asserts that each loss function f_t is G -Lipschitz with respect to $\|\cdot\|$, for some constant $G > 0$. The second assumption requires that the feasible set \mathcal{K} is bounded and lies between two Euclidean balls.

Assumption 2.1. There is some $G > 0$, such that for all $t \geq 1$, the function $f_t : \mathcal{K} \rightarrow \mathbb{R}$ is convex and for all $\mathbf{w} \in \mathcal{K}$ and $\mathbf{g} \in \partial f_t(\mathbf{w})$, we have $\|\mathbf{g}\| \leq G$.

Assumption 2.2. The set \mathcal{C} satisfies $\mathbb{B}(\mathbf{c}_0, r) \subseteq \mathcal{C} \subseteq \mathbb{B}(\mathbf{R})$ for some $r, R > 0$, and $\mathbf{c}_0 \in \mathbb{B}(R)$.

Prior work [18, 19] assumes the feasible set \mathcal{K} satisfies $\mathbb{B}(r) \subseteq \mathcal{K} \subseteq \mathbb{B}(R)$ for $r, R > 0$. This paper relaxes the condition $\mathbb{B}(r) \subseteq \mathcal{K}$. Requiring \mathcal{K} to contain a positive-radius ball centered at the *origin* is restrictive for general convex sets; such a ball may only exist after translation, its radius r can be small even then, and computing an optimal translation can be computationally expensive. Our algorithm eliminates the need to compute this translation, dynamically adapting to the geometry of \mathcal{K} instead. We also note that both Assumption 2.2 and the assumptions in [18, 19] require \mathcal{K} to have non-empty interior. This is without loss of generality, as any set \mathcal{K} contained in a lower-dimensional affine subspace can be reparametrized as a full-dimensional set with non-empty interior.

Throughout, we let $\kappa := R/r$ denote the asphericity of \mathcal{K} with r, R as in Assumption 2.2.

²In one dimension ($d = 1$), computing Euclidean projections is straightforward, making projected gradient methods both efficient and optimal.

Algorithm 1 GaugeDist($w; \mathcal{C}, H, c, \delta$): Approximate value and subgradient of the Gauge distance.

require: Separation oracle $\text{Sep}_{\mathcal{C}}$, input vector $w \in \mathbb{R}^d$, $H \in \mathbb{R}^{d \times d}$ PSD, $c \in \mathbb{R}^d$, and $\delta > 0$.
returns If $w \notin \mathcal{C}$ and $c \in \mathcal{C}$, then $S \approx S_{\mathcal{C}-c}(w - c)$ and $s \approx \partial S_{\mathcal{C}-c}(w - c)$.
1: Set $(b, v) \leftarrow \text{Sep}_{\mathcal{C}}(w)$. // $b=1$ if $w \in \mathcal{C}$; and 0, otherwise.
2: **if** $b = 1$ **then** // This corresponds to the case where $w \in \mathcal{C}$.
3: Set $(S, s) \leftarrow (0, 0)$.
4: **return** (S, s) .
5: Set $(b, v) \leftarrow \text{Sep}_{\mathcal{C}}(c)$. // $b=1$ if $c \in \mathcal{C}$; and 0, otherwise.
6: **if** $b = 0$ **then**
7: Set $(S, s) \leftarrow (0, 3dv/\sqrt{v^\top H v})$.
8: **return** (S, s) .
9: Set $\alpha \leftarrow 0$, $\beta \leftarrow 1$, and $\mu \leftarrow (\alpha + \beta)/2$.
10: **while** $\beta - \alpha > \frac{\delta}{8d^2}$ **do**
11: Set $(b, v) \leftarrow \text{Sep}_{\mathcal{C}}(\mu(w - c) + c)$. // $b=1$ if $\mu(w - c) + c \in \mathcal{C}$; and 0, otherwise.
12: **if** $\beta \cdot v^\top(w - c) < \frac{1}{2d}\sqrt{v^\top H v}$ **then break**
13: Set $\alpha \leftarrow \mu$ if $b = 1$; and $\beta \leftarrow \mu$ otherwise.
14: Set $\mu \leftarrow (\alpha + \beta)/2$.
15: Set $S \leftarrow \alpha^{-1} - 1$ and $s \leftarrow \frac{v}{\beta \cdot v^\top(w - c)}$.
16: **return** (S, s) .

Additional notation. We denote by $\mathcal{K}^\circ := \{x \in \mathbb{R}^d : \langle x, y \rangle \leq 1, \forall y \in \mathcal{K}\}$ the *polar* set of \mathcal{K} [13]. We let $\mathbb{S}_{>0}^{d \times d}$ denote the set of $d \times d$ positive-definite matrices and $\|x\|_H := \sqrt{x^\top H x}$ for $x \in \mathbb{R}^d$ and $H \in \mathbb{R}^{d \times d}$. For $c \in \mathbb{R}^d$, $H \in \mathbb{S}_{>0}^{d \times d}$, we define the ellipsoid $\mathcal{E}(c, H) := \{u \in \mathbb{R}^d \mid (u - c)^\top H^{-1}(u - c) \leq 1\}$. We use $\tilde{O}(\cdot)$ to hide poly-log factors in parameters appearing in the expression.

2.2 Gauge Function, Distance, and Projection

We now introduce the concepts of the Gauge function, distance, and projection [18, 19], which are central to our algorithm and existing separation-based methods. Throughout this section, these concepts are defined using a convex set $\mathcal{C} \subseteq \mathbb{R}^d$ containing the origin. Here, \mathcal{C} is used for definitional purposes only and does not necessarily correspond to the feasible set \mathcal{K} in our OCO problem.

Definition 2.2. The Gauge function $\gamma_{\mathcal{C}} : \mathbb{R}^d \rightarrow \mathbb{R}$ of \mathcal{C} is defined as $\gamma_{\mathcal{C}}(u) := \inf\{\lambda \in \mathbb{R}_{\geq 0} \mid u \in \lambda\mathcal{C}\}$.

The Gauge function $\gamma_{\mathcal{C}}$ can be viewed as a “pseudo” norm induced by the convex set \mathcal{C} ; it becomes a true norm when \mathcal{C} is centrally symmetric (i.e., $\mathcal{C} = -\mathcal{C}$). With this, we define the Gauge distance.

Definition 2.3 (Gauge distance). The Gauge distance function $S_{\mathcal{C}}$ corresponding to the set \mathcal{C} is

$$S_{\mathcal{C}}(u) := \min(0, \gamma_{\mathcal{C}}(u) - 1), \quad u \in \mathbb{R}^d.$$

When the set \mathcal{C} contains the origin in its interior, [18] showed that $S_{\mathcal{C}}$ has the more distance-like expression $S_{\mathcal{C}}(u) := \inf_{x \in \mathcal{C}} \gamma_{\mathcal{C}}(u - x)$, for all $u \in \mathbb{R}^d$. From this expression, we see that the Gauge distance generalizes the Euclidean distance, which is obtained by replacing $\gamma_{\mathcal{C}}$ [resp. \mathcal{C}] with the Euclidean norm $\|\cdot\|$ [resp. $\mathbb{B}(1)$]. Observe that when $u \in \mathcal{C}$, then $S_{\mathcal{C}}(u) = 0$. Otherwise, $S_{\mathcal{C}}(u) > 0$.

Gauge projection. Similar to [18, 19], our algorithm relies on Gauge projections to ensure feasible iterates. The Gauge projection operator $\Pi_{\mathcal{C}}^{\text{gau}}$ induced by \mathcal{C} is the mapping: $\Pi_{\mathcal{C}}^{\text{gau}}(u) := \arg \min_{x \in \mathcal{C}} \gamma_{\mathcal{C}}(u - x)$, for all $u \in \mathbb{R}^d$. [18] showed that when 0 lies in the interior of \mathcal{C} , the gauge projection admits the closed-form expression $\Pi_{\mathcal{C}}^{\text{gau}}(u) = \frac{u}{1 + S_{\mathcal{C}}(u)}$. This makes the gauge projection particularly useful in our setting, as it can be approximated efficiently (by approximating the gauge distance $S_{\mathcal{C}}$) using a logarithmic number of separation oracle calls.

In this paper, we use Algorithm 1 to approximate both the Gauge distance and its subgradients, which are required by our OCO algorithm. The algorithm, similar to [19, Algorithm 1], uses calls to a separation oracle for \mathcal{C} and binary search to approximate the distance function. In a nutshell, given input (\mathcal{C}, H, w, c) such that $c \in \mathcal{C}$ and $w \notin \mathcal{C}$, Algorithm 1 returns a pair (S, s) , where S is

an approximation of the Gauge distance $\mathcal{S}_{\mathcal{C}-\mathbf{c}}(\mathbf{w} - \mathbf{c})$, and \mathbf{s} is an approximate subgradient of this function at $\mathbf{w} - \mathbf{c}$. Algorithm 1 is slightly more general than [19, Algorithm 1] in that it allows for $\mathbf{c} \neq \mathbf{0}$ and can handle $\mathbf{c} \notin \mathcal{C}$. This feature is important because our projection-free algorithm will operate on a sequence of potentially uncentered ellipsoids. The next lemma is proven in Appendix A.

Lemma 2.1. *Let $\delta \in (0, 1)$, $\mathbf{w}, \mathbf{c} \in \mathbb{R}^d$, $\mathcal{C} \subseteq \mathbb{R}^d$ convex, and $H \in \mathbb{S}_{>0}^{d \times d}$ be given such that $\mathcal{C} \subseteq \mathcal{E} := \{\mathbf{x} \mid (\mathbf{x} - \mathbf{c})^\top H^{-1}(\mathbf{x} - \mathbf{c}) \leq 1\}$ and $\mathbf{w} \in \mathcal{E}$. Consider a call to Algorithm 1 with input $(\mathcal{C}, H, \mathbf{c}, \mathbf{w}, \delta)$ and let (S, \mathbf{s}) be its output. Then, either $\|H^{1/2}\mathbf{s}\| > 2d$; or we have for all $\mathbf{u} \in \mathbb{R}^d$:*

$$\mathbf{c} \in \mathcal{C}, \quad \mathcal{S}_{\mathcal{C}-\mathbf{c}}(\mathbf{w} - \mathbf{c}) \leq S \leq \mathcal{S}_{\mathcal{C}-\mathbf{c}}(\mathbf{w} - \mathbf{c}) + \delta, \text{ and } \mathcal{S}_{\mathcal{C}-\mathbf{c}}(\mathbf{u} - \mathbf{c}) \geq \mathcal{S}_{\mathcal{C}-\mathbf{c}}(\mathbf{w} - \mathbf{c}) + (\mathbf{u} - \mathbf{w})^\top \mathbf{s} - \delta.$$

Further, Algorithm 1 makes at most $2 + \log_2(8d^2/\delta)$ calls to the separation oracle $\text{Sep}_{\mathcal{C}}$.

As we will see in Section 4.1, the case where $\|H^{1/2}\mathbf{s}\| > 2d$ in Lemma 2.1 corresponds to a situation where the set $H^{-1/2}(\mathcal{C} - \mathbf{c})$ is too “thin” in the direction $H^{1/2}\mathbf{s}$. We will later use this test to dynamically update ellipsoids containing the feasible set, adapting them to its geometry.

3 Overview and Limitations of Previous Approaches

In this section, we briefly review existing separation-based approaches that our work builds on and highlight their main limitations that we address in this paper.

[18, 19] reduce OCO over \mathcal{K} to OCO over a ball $\mathbb{B}(R) \supseteq \mathcal{K}$. For this reduction, they rely on an inner OCO algorithm that produces iterates (\mathbf{u}_t) in $\mathbb{B}(R)$ (where Euclidean projections are cheap), from which feasible iterates (\mathbf{w}_t) are extracted using Gauge projections (also cheap when given a separation oracle for \mathcal{K} ; see Section 2.2). The feedback vector fed to the inner algorithm is given by

$$\tilde{\mathbf{g}}_t = \mathbf{g}_t - \mathbb{I}\{\langle \mathbf{g}_t, \mathbf{u}_t \rangle < 0\} \cdot \langle \mathbf{g}_t, \mathbf{w}_t \rangle \cdot \mathbf{s}_t, \quad (1)$$

where $\mathbf{s}_t \in \partial \mathcal{S}_{\mathcal{K}}(\mathbf{w}_t)$. The subgradient \mathbf{s}_t can be computed efficiently using the GaugeDist subroutine in Algorithm 1; see Section 2.2. This choice of $\tilde{\mathbf{g}}_t$ guarantees that

$$\forall t \geq 1, \forall \mathbf{u} \in \mathcal{K}, \quad \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle \leq \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle, \quad (2)$$

which represents a key result in their reduction; see [19, Lemma 4.1]. The inequality in (2) is useful because it allows one to bound the instantaneous regret of their algorithm (left-hand side of (2)) in terms of that of the inner algorithm (right-hand side of (2)). Thus, if the inner algorithm achieves low regret over the ball $\mathbb{B}(R)$, this guarantee essentially carries over to their final algorithm via (2).

The key issue with this approach is that standard OCO algorithms typically have regret bounds that scale with the maximum norm of the feedback vectors. Thus, if the inner algorithm is one such algorithm, its regret will scale linearly with $\max_{t \in [T]} \|\tilde{\mathbf{g}}_t\|$. Due to the presence of \mathbf{s}_t in the definition of $\tilde{\mathbf{g}}_t$ in (1), and because $\|\mathbf{s}_t\|$ can be as large as the asphericity $\kappa = R/r$ (see [19, Lemma 4.1]), the resulting regret bound is of order $O(\kappa\sqrt{T})$ (in our work, we aim for a logarithmic dependence on κ).

In a follow-up work, [19] manages to move the dependence on κ into a lower-order term, achieving a final regret bound of $\tilde{O}(\sqrt{dT} + \kappa d)$ by using a variant of ONS as the inner algorithm. We now sketch why ONS was key to achieving this result; this is relevant as we also use ONS as the inner algorithm.

Advantage of ONS. The regret of ONS with learning rate η can be bounded by (see [19, Thm. 3.1])

$$\frac{\eta}{2} \sum_{t \in [T]} \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle^2 + O(1) \cdot \frac{d \log(BT)}{\eta}, \quad \forall \mathbf{u} \in \mathcal{K}, \text{ where } B := \max_{t \in [T]} \|\tilde{\mathbf{g}}_t\|. \quad (3)$$

On the other hand, as observed by [19], as long as $\eta \leq (\max_{t \in [T]} \|\langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle\|)^{-1}$, Eq. (2) implies

$$\sum_{t=1}^T \left(\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle - \frac{\eta}{2} \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle^2 \right) \leq \sum_{t=1}^T \left(\langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle - \frac{\eta}{2} \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle^2 \right), \quad \forall \mathbf{u} \in \mathcal{K}.$$

Combining this with (3) and rearranging terms implies that the regret $\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle$ is bounded by $\frac{\eta}{2} \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle^2 + O(1) \cdot \frac{d \log(BT)}{\eta}$, for all $\mathbf{u} \in \mathcal{K}$. Tuning η optimally in the interval between 0 and $(\max_{t \in [T]} \|\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle\|)^{-1}$ yields the regret bound

$$O\left(G\sqrt{dT \log(BT)} + \max_{t \in [T]} \|\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle\| \cdot d \log(BT)\right), \quad (4)$$

for all $\mathbf{u} \in \mathcal{K}$. The key takeaways from (4) are two-fold: I) Unlike the regret bound of [18] where the dependence on B is linear, the B term in (4), which grows with the maximum norm of the vectors (\mathbf{s}_t) , appears only inside a logarithmic term; and II) although the scale of the vectors (\mathbf{s}_t) still affects the bound in (4) outside logarithms through $\max_{t \in [T]} |\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle|$, this quantity can be significantly smaller than $\|\mathbf{s}_t\|$ itself, as we will see in the analysis. [19] show that (4) can be bounded from above by $\tilde{O}(\sqrt{dT} + \kappa d)$. However, the κd term, while independent of T , can still be arbitrarily large for ill-conditioned sets, and we would like to remove it.

Limitations of reparametrization. Although any set \mathcal{K} can be put into isotropic position so that its asphericity κ is reduced to at most $O(d)$ [6], computing an approximate isotropic transformation may require up to $\tilde{O}(d^4)$ calls to a separation oracle [16], which can be computationally prohibitive in many practical applications.

4 Algorithm and Guarantees

In Section 4.1, we present our algorithm and outline the core ideas behind its design and how they allow us to overcome the limitations of previous approaches discussed in Section 3. In Section 4.2, we formally state our algorithm guarantees.

Algorithm 2 Ellipsoid algorithm for online convex optimization.

require: Rounds T , a feasible point $\mathbf{c} \in \mathcal{K}$, r, R , and $G, \eta > 0$.

- 1: Set $\kappa \leftarrow \frac{R}{r}$, $\varepsilon \leftarrow \frac{1}{\kappa^{18} T^2}$, $\beta \leftarrow \eta G^2$, $\mathbf{c}_1 \leftarrow \mathbf{c}$, $H_1 \leftarrow R^2 I$, $\Sigma_1 \leftarrow \beta I$, $(\lambda_{\min}, \lambda_{\max}) \leftarrow (\frac{\varepsilon}{24\kappa^2}, \frac{40d^2\kappa}{\varepsilon})$.
- 2: **for** $t = 1, \dots, T$ **do**
- 3: Set $(S_t, \mathbf{s}_t) \leftarrow \text{GaugeDist}(\mathbf{u}_t; \mathcal{K}, H_t, \mathbf{c}_t, \varepsilon)$. // $S_t \approx S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t)$; $\mathbf{s}_t \approx \partial S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t)$.
- 4: **if** $\|H_t^{1/2} \mathbf{s}_t\| > 2d$ **then**
- 5: Play $\mathbf{w}_t = \mathbf{c}$ and observe $\mathbf{g}_t \in \partial f_t(\mathbf{w}_t)$.
- 6: Set $\tilde{\mathbf{g}}_t \leftarrow \mathbf{0}$.
- 7: // Update the ellipsoid containing \mathcal{K} .
- 8: Update $\mathbf{c}_{t+1} \leftarrow \mathbf{c}_t - \frac{1}{2d+2} \cdot \frac{H_t \mathbf{s}_t}{\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t}}$.
- 9: Update $H_{t+1} \leftarrow \frac{4d^2-1}{4d^2-4} \cdot \left(H_t - \frac{2d}{2d^2+d-1} \frac{H_t \mathbf{s}_t \mathbf{s}_t^\top H_t}{\mathbf{s}_t^\top H_t \mathbf{s}_t} \right)$.
- 10: **else**
- 11: Play $\mathbf{w}_t = \frac{\mathbf{u}_t - \mathbf{c}_t}{1 + S_t} + \mathbf{c}_t$ and observe $\mathbf{g}_t \in \partial f_t(\mathbf{w}_t)$. // $\mathbf{w}_t \approx \Pi_{\mathcal{K}-\mathbf{c}_t}^{\text{gau}}(\mathbf{u}_t - \mathbf{c}_t) + \mathbf{c}_t$.
- 12: Set $\tilde{\mathbf{g}}_t = \mathbf{g}_t - \mathbb{I}\{\langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle < 0\} \cdot \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{c}_t \rangle \cdot \mathbf{s}_t$.
- 13: Update $\mathbf{c}_{t+1} \leftarrow \mathbf{c}_t$ and $H_{t+1} \leftarrow H_t$.
- 14: /* Do one ONS in $\mathcal{E}_{t+1} := \{\mathbf{u} \mid (\mathbf{u} - \mathbf{c}_{t+1})^\top H_{t+1}^{-1}(\mathbf{u} - \mathbf{c}_{t+1}) \leq 1\}$ given “loss vector” $\tilde{\mathbf{g}}_t$ */
- 15: Set $\Sigma_{t+1} \leftarrow \Sigma_{t+1} + \eta \tilde{\mathbf{g}}_t \tilde{\mathbf{g}}_t^\top$.
- 16: Set $\mathbf{z}_{t+1} \leftarrow \mathbf{u}_t - \Sigma_{t+1}^{-1} \tilde{\mathbf{g}}_t$.
- 17: // Approximate $\mathbf{u}_{t+1} \leftarrow \arg \min_{\mathbf{u} \in \mathcal{E}_{t+1}} (\mathbf{z}_{t+1} - \mathbf{u})^\top \Sigma_{t+1} (\mathbf{z}_{t+1} - \mathbf{u})$ with PoE (Algorithm 3).
- 18: Set $\mathbf{u}_{t+1} \leftarrow \text{PoE}(\mathbf{z}_{t+1}, \mathbf{c}_{t+1}, \beta R^2 \Sigma_{t+1}^{-1}, H_{t+1}; \lambda_{\min}, \lambda_{\max}, \varepsilon)$.

4.1 Algorithm Overview and Design Rationale

Our main algorithm (Algorithm 2) effectively reduces OCO over the feasible set \mathcal{K} to OCO over a sequence of ellipsoids

$$\mathcal{E}_t = \mathcal{E}(\mathbf{u}_t, H_t) := \{\mathbf{u} \in \mathbb{R}^d \mid (\mathbf{u} - \mathbf{c}_t)^\top H_t^{-1}(\mathbf{u} - \mathbf{c}_t) \leq 1\}, \quad t \geq 1,$$

each containing \mathcal{K} , with centers $(\mathbf{c}_t) \subset \mathbb{R}^d$ and positive definite matrices $(H_t) \subseteq \mathbb{S}_{>0}^{d \times d}$. Intuitively, with each update, the ellipsoid increasingly pulls in around the feasible set, yielding a progressively tighter approximation of \mathcal{K} and allowing us to bypass the need for any expensive pre-processing of \mathcal{K} , even when \mathcal{K} is ill-conditioned. The advantage of working with ellipsoids containing \mathcal{K} instead of \mathcal{K} itself is that projections onto the former can be carried out more efficiently; we will see that the complexity of doing so is almost independent of the geometry of \mathcal{K} . The specifics of how the ellipsoids are constructed and when updates occur will be explained in detail in the sequel.

Inner algorithm. Similar to prior works [18, 19], our reduction relies on an inner OCO algorithm whose iterate at round t lies within the ellipsoid \mathcal{E}_t . For clarity of exposition, we directly instantiate

the inner algorithm with a variant of ONS (see Line 13 and Line 15) that is adequate to achieve our desired regret bound. The version of ONS we use is similar to the original algorithm proposed by [11], with the key difference is that we perform *generalized projections* over a sequence of varying sets, namely the ellipsoids (\mathcal{E}_t) , rather than projecting onto a fixed set; the generalized projection solves the problem $\inf_{\mathbf{u} \in \mathcal{C}} \|\mathbf{u} - \mathbf{c}_t\|_H^2$ for a convex set \mathcal{C} and a matrix $H \in \mathbb{S}_{>0}^{d \times d}$.

Efficient “projections.” In Algorithm 2, we use the PoE subroutine (described in Appendix C) to efficiently perform generalized projections onto the ellipsoids (\mathcal{E}_t) (see Line 15); this costs $\tilde{O}(d^\omega)$. The iterates (\mathbf{u}_t) produced by ONS are then projected onto the set \mathcal{K} using Gauge projections (Line 3 and Line 10), which we showed can be implemented using logarithmically many calls to a separation oracle for \mathcal{K} (see Section 2.2). This yields a sequence of iterates (\mathbf{w}_t) that are guaranteed to be in \mathcal{K} .

Feedback to inner algorithm. The feedback given to the inner ONS algorithm is a modified gradient vector $\tilde{\mathbf{g}}_t$ (reminiscent of the choice of $\tilde{\mathbf{g}}_t$ in prior works displayed in (1)) defined as

$$\tilde{\mathbf{g}}_t = \mathbf{g}_t - \mathbb{I}\{\langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle < 0\} \cdot \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{c}_t \rangle \cdot \mathbf{s}_t, \quad (5)$$

where \mathbf{c}_t is the center of the ellipsoid at iteration t , and \mathbf{s}_t is an approximate subgradient of the Gauge distance function $\mathcal{S}_{\mathcal{K}-\mathbf{c}_t}$ at $\mathbf{u}_t - \mathbf{c}_t$. This subgradient is computed efficiently (see Line 3) using the GaugeDist subroutine from Section 2.2. Similar to (2), this choice of $\tilde{\mathbf{g}}_t$ ensures that

$$\langle \mathbf{w}_t - \mathbf{u}, \mathbf{g}_t \rangle \leq \langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle, \quad \forall t \geq 1, \forall \mathbf{u} \in \mathcal{K}, \quad (6)$$

where $\mathbf{g}_t \in \partial f_t(\mathbf{w}_t)$ is a subgradient of the loss function at the outer iterate \mathbf{w}_t , and \mathbf{u}_t is the iterate of the inner ONS algorithm. This inequality is useful because it allows us to relate the instantaneous regret of Algorithm 2 to that of the inner ONS algorithm. Thus, as long as ONS, run over the sequence of ellipsoids containing \mathcal{K} , guarantees low regret, we get low regret for Algorithm 2.

As mentioned earlier, one advantage of working with ellipsoids that contain \mathcal{K} is that projections onto them can be performed more efficiently than projections onto \mathcal{K} itself. However, this alone does not explain our use of a sequence of varying ellipsoids, rather than a fixed ball as in [18, 19]. The key reason is that varying ellipsoids allow us to dynamically adapt to the geometry of \mathcal{K} , and avoid any linear dependence in κ in our final regret bound.

Mitigating κ -dependence. As discussed in Section 3, prior methods incur a κ term due to either a large norm of $\mathbf{s}_t \in \partial \mathcal{S}_{\mathcal{K}}(\mathbf{w}_t)$ (as in [18]) or a large inner product $|\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle|$ for some t and comparator \mathbf{u} (as in [19]). In the latter case, observe that $|\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle|$ becomes large only if $|\langle \mathbf{u}_t - \mathbf{u}, \mathbf{s}_t \rangle|$ is large, due to the expression for $\tilde{\mathbf{g}}_t$ in (1) (also the case for our choice of $\tilde{\mathbf{g}}_t$ in (5)).

As we will clarify shortly, a large value of $|\langle \mathbf{u}_t - \mathbf{u}, \mathbf{s}_t \rangle|$ indicates that \mathcal{K} is too “thin” in a certain direction. Our approach exploits this fact to adapt to the geometry of \mathcal{K} whenever such thin directions are detected. On the other hand, our analysis shows that these detections can happen at most $\tilde{O}(d^2)$ times, and outside of the “detection rounds”, the inner product $|\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle|$ is independent of κ , and the regret incurred by the inner ONS algorithm is well controlled.

Geometry-driven ellipsoid updates. We now examine the condition that determines when the ellipsoid is updated in Algorithm 2. From Line 4-Line 8, we see that an update is triggered only when $\|H_t^{1/2} \mathbf{s}_t\| > 2d$; in (9) below, we relate $\|H_t^{1/2} \mathbf{s}_t\|$ to the magnitude of $|\langle \mathbf{u}_t - \mathbf{u}, \mathbf{s}_t \rangle|$ in the previous paragraph. Here, \mathbf{s}_t is an approximate subgradient of $\mathcal{S}_{\mathcal{K}-\mathbf{c}_t}$ at $\mathbf{u}_t - \mathbf{c}_t$. The condition $\|H_t^{1/2} \mathbf{s}_t\| > 2d$ can be interpreted as a test for whether \mathcal{K} is thin in some direction relative to the ellipsoid

$$\mathcal{E}_t = \mathcal{E}(\mathbf{c}_t, H_t) := \{\mathbf{u} \in \mathbb{R}^d \mid (\mathbf{u} - \mathbf{c}_t)^\top H_t^{-1} (\mathbf{u} - \mathbf{c}_t) \leq 1\}.$$

We now clarify this claim. Suppose that $\|H_t^{1/2} \mathbf{s}_t\| > 2d$. Since $\mathbf{s}_t \in \partial \mathcal{S}_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t)$ (ignoring any approximations for now), it follows that $\mathbf{s}_t \in (\mathcal{K} - \mathbf{c}_t)^\circ$ (see Lemma G.5). Therefore, by the definition of the polar set $(\mathcal{K} - \mathbf{c}_t)^\circ$, we have

$$\forall \mathbf{u} \in \mathcal{K}, \quad \langle \mathbf{u} - \mathbf{c}_t, \mathbf{s}_t \rangle \leq 1. \quad (7)$$

Define the unit vector $\mathbf{v}_t := H_t^{1/2} \mathbf{s}_t / \|H_t^{1/2} \mathbf{s}_t\|$. Then, by (7) and the assumption that $\|H_t^{1/2} \mathbf{s}_t\| > 2d$, we have for all $\mathbf{u} \in \mathcal{K} - \mathbf{c}_t$, $\langle H_t^{-1/2} \mathbf{u}, \mathbf{v}_t \rangle = \frac{1}{\|H_t^{1/2} \mathbf{s}_t\|} \cdot \langle \mathbf{u}, \mathbf{s}_t \rangle \leq \frac{1}{\|H_t^{1/2} \mathbf{s}_t\|} \leq \frac{1}{2d}$. Equivalently, we have

$$\forall \mathbf{u} \in H_t^{-1/2}(\mathcal{K} - \mathbf{c}_t), \quad \langle \mathbf{u}, \mathbf{v}_t \rangle \leq \frac{1}{2d}, \quad (8)$$

which implies that the reparametrized set $H_t^{-1/2}(\mathcal{K} - \mathbf{c}_t)$ is thin in the direction of \mathbf{v}_t . When this occurs, we work with a new ellipsoid \mathcal{E}_{t+1} that better aligns with the geometry of \mathcal{K} in that direction. Specifically, we set $\mathcal{E}_{t+1} = \mathcal{E}(\mathbf{c}_{t+1}, H_{t+1})$, where \mathbf{c}_{t+1} and H_{t+1} are as in Line 7 and Line 8 of Algorithm 2, respectively. We now motivate these choices for \mathbf{c}_{t+1} and H_{t+1} .

Suppose that $\|H_t^{1/2}\mathbf{s}_t\| > 2d$, and therefore (8), holds. Then, by (8) and the fact that $\mathcal{K} \subseteq \mathcal{E}_t$, we have $H_t^{-1/2}(\mathcal{K} - \mathbf{c}_t) \subseteq \{\mathbf{u} \in \mathbb{B}(1) \mid \mathbf{u}^\top \mathbf{v}_t \leq 1/2d\}$. Thus, by Lemma B.1 (an adapted result from the analysis of the ellipsoid method; see, e.g., [2, Lem. 2.3]), we have:

$$H_t^{-1/2}(\mathcal{K} - \mathbf{c}_t) \subseteq \tilde{\mathcal{E}}_{t+1} := \mathcal{E}\left(-\frac{\mathbf{v}_t}{2(d+1)}, \tilde{H}_{t+1}\right), \quad \text{where} \quad \tilde{H}_{t+1} := \frac{4d^2-1}{4d^2-4} \left(I_d - \frac{2d}{2d^2+d-1} \mathbf{v}_t \mathbf{v}_t^\top\right).$$

It can be verified that $\mathbf{c}_t + H_t^{1/2}\tilde{\mathcal{E}}_{t+1} = \mathcal{E}(\mathbf{c}_{t+1}, H_{t+1})$, with \mathbf{c}_{t+1} and H_{t+1} as in Line 7 and Line 8, which implies that $\mathcal{K} \subseteq \mathcal{E}_{t+1}$. Lemma B.1 also implies that $\text{vol}(\mathcal{E}_{t+1}) \leq \exp(-\frac{1}{8d}) \cdot \text{vol}(\mathcal{E}_t)$; that is, we not only maintain the invariant $\mathcal{K} \subseteq \mathcal{E}_{t+1}$, but we also ensure that the volume of the ellipsoid shrinks with each update, providing a tighter approximation of \mathcal{K} . Since $\mathcal{K} \subseteq \mathcal{E}_t$, for all $t \geq 1$, the volumes ($\text{vol}(\mathcal{E}_t)$) cannot shrink indefinitely, enabling us to bound the number of ellipsoid updates.

Bounding the number of ellipsoid updates. By Assumption 2.2 and that $\text{vol}(\mathcal{E}_{t+1}) \leq e^{-\frac{1}{8d}} \text{vol}(\mathcal{E}_t)$, we can show that the number of updates is bounded by $O(d^2 \log(R/r))$ (recall that $\mathbb{B}(\mathbf{c}_0, r) \subseteq \mathcal{K} \subseteq \mathbb{B}(R)$). In other words, the condition on Line 4 can be satisfied at most $O(d^2 \log(R/r))$ times.

Regret on update rounds. In our analysis, whenever the ellipsoid is updated at round t (i.e., the condition $\|H_t^{1/2}\mathbf{s}_t\| > 2d$ on Line 4 is satisfied), the algorithm simply plays a default vector $\mathbf{c} \in \mathcal{K}$. The cumulative cost of playing \mathbf{c} over these rounds is at most $O(d^2 \log(R/r))$, due to the bound on the number of updates; we are willing to absorb this overhead into the overall regret bound.

Regret on no-update rounds. Now, we argue that when the ellipsoid is not updated at round t (i.e., when $\|H_t^{1/2}\mathbf{s}_t\| \leq 2d$), the inner product $|\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle|$ remains bounded independently of κ for all $\mathbf{u} \in \mathcal{K}$, which ensures that the regret of the inner ONS algorithm on these rounds remains small.

Suppose that $\|H_t^{1/2}\mathbf{s}_t\| \leq 2d$ (i.e., no ellipsoid update). In this case, by Hölder's inequality:

$$|\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle| \leq \|H_t^{-1/2}(\mathbf{u}_t - \mathbf{u})\| \cdot \|H_t^{1/2}\mathbf{s}_t\| \leq 2d, \quad \forall t \in [T], \forall \mathbf{u} \in \mathcal{K}, \quad (9)$$

where the last inequality follows from the assumption that $\|H_t^{1/2}\mathbf{s}_t\| \leq 2d$ and that both \mathbf{u}_t and \mathbf{u} lie in \mathcal{E}_t . Indeed, $\mathbf{u}_t \in \mathcal{E}_t$ because the inner ONS algorithm outputs iterates in \mathcal{E}_t , and $\mathbf{u} \in \mathcal{E}_t$ because our construction maintains the invariant $\mathcal{K} \subseteq \mathcal{E}_t$. The inequality in (9) implies that for all $\mathbf{u} \in \mathcal{K}$, we have $|\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle| \leq O(Gd)$ (by the definition of $\tilde{\mathbf{g}}_t$ in (5) and Assumption 2.1), which, from (4), is what we need to ensure the regret bound of the inner ONS scales only logarithmically in κ .

4.2 Algorithm Guarantees

We now formalize the claims made in Section 4.1 and present the regret guarantee of our algorithm. We start with the invariants that Algorithm 2 maintains. The proof of the lemma is in Appendix E.1.

Lemma 4.1. *Let $T \geq 1$, $\eta > 0$, and $\mathbf{c} \in \mathcal{K}$ be given and suppose that Assumption 2.2 and Assumption 2.1 hold with $0 < r \leq R$ and $G > 0$, respectively. Consider a call to Algorithm 2 with input $(T, \mathbf{c}, r, R, G, \eta)$. Then, for any subgradients (\mathbf{g}_t) , the variables in Algorithm 2 satisfy for all $t \in [T]$:*

1. $\mathcal{K} \subseteq \mathcal{E}_t := \{\mathbf{u} \in \mathbb{R}^d \mid (\mathbf{u} - \mathbf{c}_t)^\top H_t^{-1}(\mathbf{u} - \mathbf{c}_t) \leq 1\}$;
2. If $\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} \leq 2d$, then $\mathbf{c}_t \in \mathcal{K}$;
3. $\text{vol}(\mathcal{E}_t) \leq e^{-\frac{N_t}{8d}} \cdot R^{2d}$, where $N_t := \sum_{\tau=1}^{t-1} \mathbb{I}\{\sqrt{\mathbf{s}_\tau^\top H_\tau \mathbf{s}_\tau} > 2d\}$;
4. $N_t \leq 8d^2 \log(R/r)$;
5. $\sigma_{\min}(H_t) \geq r^2$ and $\sigma_{\max}(H_t) \leq (1 + 2/d^2)^{N_t} R^2 \leq \kappa^{16} R^2$.

Consistent with the claim made in Section 4.1, Item 1 of the lemma establishes that the set \mathcal{K} is always contained within the ellipsoids (\mathcal{E}_t) . Item 2 shows that $\mathbf{c}_t \in \mathcal{K}$ whenever $\|H_t^{1/2}\mathbf{s}_t\| \leq 2d$, ensuring that when the ellipsoid is not updated, the center \mathbf{c}_t of the ellipsoid \mathcal{E}_t lies within \mathcal{K} . This is crucial for the correctness of our reduction, and in particular for the validity of (6). Item 3 shows that the volume of the ellipsoid shrinks by a factor of $e^{-1/(8d)}$ whenever an update occurs. Item 4 bounds the total number of such updates by $O(d^2 \log(R/r))$.

The next lemma formalizes our reduction result in (6), showing that the instantaneous regret of Algorithm 2 is bounded by the instantaneous regret of the inner ONS algorithm. The lemma also bounds key quantities discussed in Section 4.1 that appear in the ONS regret bound.

Lemma 4.2. *Let $T \geq 2$, $\eta > 0$, and $\mathbf{c} \in \mathcal{K}$ be given and suppose that Assumption 2.2 and Assumption 2.1 hold with $0 < r \leq R$ and $G > 0$, respectively. Consider a call to Algorithm 2 with input $(T, \mathbf{c}, r, R, G, \eta)$. Then, for any subgradients (\mathbf{g}_t) , the variables in Algorithm 2 satisfy for all $t \in [T]$:*

1. $\|\tilde{\mathbf{g}}_t\| \leq G(1 + 4\kappa d)$;
2. $\mathbf{w}_t \in \mathcal{K}$;
3. For all $\mathbf{u} \in \mathcal{K}$, $|\langle \mathbf{u} - \mathbf{u}_t, \tilde{\mathbf{g}}_t \rangle| \leq 2RG(1 + 4d)$;
4. For all $\mathbf{u} \in \mathcal{K}$, $\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle \cdot \mathbb{I}\{\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} \leq 2d\} \leq \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle + \frac{3GR}{T}$.

The proof of the lemma is in Appendix E.1. Item 1 bounds the norm of the feedback vectors $(\tilde{\mathbf{g}}_t)$ passed to the inner ONS algorithm. While the norm of $\tilde{\mathbf{g}}_t$ can be as large as $O(G\kappa d)$, this is acceptable because the regret bound for ONS only involves $\|\tilde{\mathbf{g}}_t\|$ inside logarithmic terms (see (4) and recall that $B = \max_{t \in [T]} \|\tilde{\mathbf{g}}_t\|$). This is precisely why the choice of ONS as the inner algorithm is essential to our approach. Item 2 guarantees the feasibility of the iterates of Algorithm 2. Item 3 shows that the inner product $|\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle|$ is at most $O(RGd)$ for all $\mathbf{u} \in \mathcal{K}$, and crucially, this bound is independent of κ . This is key to ensuring that the regret of the inner ONS algorithm does not depend on κ outside of logarithmic terms; see (4). Finally, Item 4, a consequence of our specific choice of $\tilde{\mathbf{g}}_t$ in (5), shows that, on rounds where no thin direction is detected, the instantaneous regret of Algorithm 2 is bounded by that of the inner ONS algorithm (up to a small additive term).

With these results in place, we now proceed to bound the regret of the inner ONS algorithm. The bound in the next lemma should be reminiscent of the ONS bound displayed in (3) (proof in Appendix D).

Lemma 4.3 (ONS Regret). *Let $T \geq 2$, $\eta > 0$, and $\mathbf{c} \in \mathcal{K}$ be given and suppose that Assumption 2.2 and Assumption 2.1 hold with $0 < r \leq R$ and $G > 0$, respectively. Consider a call to Algorithm 2 with input $(T, \mathbf{c}, r, R, G, \eta)$. Then, for any (\mathbf{g}_t) , the vectors $(\tilde{\mathbf{g}}_t, \mathbf{u}_t, \mathbf{c}_t, H_t)$ in Algorithm 2 satisfy*

$$\forall \mathbf{u} \in \mathcal{K}, \quad \sum_{t=1}^T \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle - \frac{\eta}{2} \sum_{t=1}^T \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle^2 \leq 2\eta G^2 R^2 + 240\eta R^2 G^2 d^2 + \frac{d \log(1 + 9d^2 T^2 R^2 / r^2)}{2\eta},$$

and $\mathbf{u}_t \in \mathcal{E}_t := \mathcal{E}(\mathbf{c}_t, H_t)$, $\forall t \geq 1$. Further, given $\tilde{\mathbf{g}}_t$, computing \mathbf{u}_{t+1} costs at most $O(d^\omega \log(TR/r))$.

The proof closely follows the original analysis of the ONS algorithm by [11], despite the fact that we are working with a sequence of varying feasible sets (\mathcal{E}_t) ; what is important here is that we have $\mathcal{K} \subseteq \mathcal{E}_t$, for all $t \geq 1$. One key difference from the analysis of [11] is that we do not assume access to an exact oracle for generalized projection, which involves solving $\inf_{\mathbf{u} \in \mathcal{C}} \|\mathbf{u} - \mathbf{c}_t\|_H^2$ for a convex set \mathcal{C} and matrix $H \in \mathbb{S}_{>0}^{d \times d}$. Instead, we use the PoE subroutine (Algorithm 3), which approximates this projection in $O(d^\omega \log(TR/r))$ time (see Lemma C.1) using binary search when $\mathcal{C} = \mathcal{E}_t$, for $t \geq 1$. Note that this translates into a $\tilde{O}(C_{\text{sep}}(\mathcal{K}) + d^\omega)$ per-round cost for our OCO algorithm (Algorithm 2). We formalize this next and give the main regret guarantee for Algorithm 2 (proof in Appendix E.2).

Theorem 4.1 (Main Regret). *Let $T \geq 2$ and $\mathbf{c} \in \mathcal{K}$ be given and suppose that Assumption 2.2 and Assumption 2.1 hold with $0 < r \leq R$ and $G > 0$, respectively. Consider a call to Algorithm 2 with input $(T, \mathbf{c}, r, R, G, \eta)$, for $\eta \leq \frac{1}{10dGR}$. Then, for any (\mathbf{g}_t) , the iterates (\mathbf{w}_t) of Algorithm 2 satisfy*

$$\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle \leq \frac{\eta}{2} \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle^2 + \frac{d \log(1 + 9d^2 T^2 R^2 / r^2)}{2\eta} + 50d^2 GR(1 + \log(R/r)). \quad (10)$$

for all $\mathbf{u} \in \mathcal{K}$. Further, the per-round cost of the algorithm is at most $O((C_{\text{sep}}(\mathcal{K}) + d^\omega) \cdot \log(TR/r))$.

Regret bound after tuning η . By Assumption 2.1 and the fact that $\mathbf{w}_t \in \mathcal{K} \subseteq \mathbb{B}(R)$, we have that $|\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle| \leq 2RG$, for all $t \in [T]$. Thus, setting $\eta = (GR\sqrt{T \log(T\kappa)})^{-1} \wedge (10dGR)^{-1}$ in Theorem 4.1 gives a $O(GR\sqrt{dT \log(\kappa T)} + d^2 \log(\kappa T))$ regret bound for Algorithm 2 as desired.

Rates for stochastic optimization. By combining this regret bound with a standard online-to-batch conversion technique (e.g., [10]), we obtain a convergence rate of $\tilde{O}(\sqrt{d/T} + d^2/T)$ for stochastic convex optimization. However, due to the presence of the “second-order” term $\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle^2$ in (10), we can achieve an improved rate of $\tilde{O}(\sigma\sqrt{d/T} + d^2/T)$, which simplifies to $\tilde{O}(d^2/T)$ when the gradient noise σ is zero; see Theorem F.1 for a formal statement.

5 Discussion on Computational Efficiency

A key consideration for our algorithm is its computational runtime, particularly in comparison to classical projection-based methods and other projection-free approaches. The per-iteration computational cost of our proposed algorithm is on the order of $\tilde{O}(d^\omega + C_{\text{sep}})$, where $\omega \approx 2.372$ is the exponent of matrix multiplication, and this cost arises from solving a $d \times d$ linear system. The C_{sep} term is the cost of a single call to the separation oracle.

While the d^ω term appears significant (larger than the d^2 in [19], for instance), it is crucial to contextualize this cost. In many large-scale optimization settings, the separation cost C_{sep} is the dominant computational bottleneck. A canonical example is optimization over a polytope defined by m linear constraints in \mathbb{R}^d : $\mathcal{K} = \{x \in \mathbb{R}^d \mid a_i^\top x \leq b_i, \forall i \in [m]\}$.

In this common setup, a call to the separation oracle reduces to membership testing, which requires computing m inner products. The separation cost is therefore $C_{\text{sep}} = O(md)$. The d^ω term in our complexity is dominated by this separation cost as soon as $m \geq d^{\omega-1} \approx d^{1.37}$. In this regime, our per-iteration cost becomes $\tilde{O}(md)$, which is cheaper than Euclidean projection as we discuss next.

5.1 Cost of Euclidean Projection in the Polytope Setting

We compare the projection cost of two standard methods in the canonical polytope setting.

Using an Interior-Point Method (IPM), the cost of a single projection is $\tilde{O}(\sqrt{m} \cdot (d^\omega + md^2))$ (see, e.g., [1]); here, \sqrt{m} is the number of IPM iterations, and $d^\omega + md^2$ is the per-iteration cost, which stems from computing the Hessian of the logarithmic barrier (md^2) and inverting it (d^ω).

Another alternative for projection is to use a cutting-plane method [24]. The complexity of this approach is $\tilde{O}(d \cdot C_{\text{sep}} + d^4)$ per projection. In the polytope setting, this becomes $\tilde{O}(md^2 + d^4)$.

We see that the computational cost of both projection methods is significantly higher than our method's per-iteration cost of $\tilde{O}(md + d^\omega)$ when $m \geq d$.

While per-iteration cost is important, a more holistic measure of efficiency is the overall runtime required to reach a target ε -average regret. We analyze this next.

5.2 Runtimes to Achieve an ε -Average Regret

We compare our method's runtime for achieving an ε -average regret against OGD with projections performed with either IPM or a cutting-plane method in the polytope setting.

Since OGD has a $O(\sqrt{T})$ regret, the total runtime of OGD to achieve an ε -average regret is bounded by $O(C_{\text{Proj}}/\varepsilon^2)$, where C_{Proj} is the cost of projection. Thus, instantiating the projection costs for the IPM and cutting-planes methods in Section 5.1, we get the runtimes:

$$\tilde{O}(\sqrt{m}(d^\omega + md^2)) \cdot \varepsilon^{-2} \text{ (OGD+IPM)}, \quad \text{and} \quad \tilde{O}(md^2 + d^4) \cdot \varepsilon^{-2} \text{ (OGD+Cutline-plane)}$$

The comparable runtime for our proposed approach is $\tilde{O}(md^2 + d^{\omega+1}) \cdot \varepsilon^{-2}$. Thus, our method's runtime is faster than OGD with the projection approaches we discussed, as long as $m \geq d$.

We note, however, that newer cutting-plane methods exist with an improved complexity, running in $\tilde{O}(md^2 + d^3)$ time per projection [15]. Consequently, pairing OGD with these state-of-the-art methods would result in a runtime to achieve an ε -average regret that is comparable to our approach, or potentially slightly better due to an improved lower-order term (d^3 instead of $d^{\omega+1}$). However, these new methods are significantly more involved and often hide large constant factors in the $\tilde{O}(\cdot)$ notation, which may limit their practical applicability.

Frank-Wolfe (FW) Methods. Finally, we position our work relative to projection-free methods based on linear optimization, such as Frank-Wolfe (FW). These methods are indeed very useful and practical, and their dimension-free guarantees are particularly appealing. We view these approaches as complementary to ours. Our algorithm can be more competitive in popular settings where, in the absence of additional structure (like a well-studied combinatorial structure), the linear optimization oracle required by FW becomes computationally expensive; in some cases, comparable to a full projection such as in the canonical polytope case discussed above.

Acknowledgments and Disclosure of Funding

We thank the reviewers for their constructive feedback on the presentation of our computational efficiency analysis.

References

- [1] Stephen P Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge university press, 2004.
- [2] Sébastien Bubeck. *Convex optimization: Algorithms and complexity*, 2015.
- [3] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.
- [4] Ashok Cutkosky. Anytime online-to-batch, optimism and acceleration. In *International Conference on Machine Learning*, pages 1446–1454. PMLR, 2019.
- [5] Ashok Cutkosky, Harsh Mehta, and Francesco Orabona. Optimal stochastic non-smooth non-convex optimization through online-to-non-convex conversion, 2023.
- [6] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394, 2005.
- [7] Marguerite Frank, Philip Wolfe, et al. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.
- [8] Dan Garber and Ben Kretzu. New projection-free algorithms for online convex optimization with adaptive regret guarantees. In *Conference on Learning Theory*, pages 2326–2359. PMLR, 2022.
- [9] Benjamin Grimmer. Radial duality part i: foundations. *Mathematical Programming*, 205(1):33–68, 2024.
- [10] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.
- [11] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.
- [12] Elad Hazan and Satyen Kale. Projection-free online learning. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1843–1850, 2012.
- [13] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of convex analysis*. Springer Science & Business Media, 2004.
- [14] DB Iudin and Arkadi S Nemirovskii. Evaluation of informational complexity of mathematical-programming programs. *Matekon*, 13(2):3–25, 1977.
- [15] Yin Tat Lee, Aaron Sidford, and Santosh S. Vempala. Efficient convex optimization with membership oracles, 2017.
- [16] László Lovász and Santosh Vempala. Simulated annealing in convex bodies and an $o^*(n^4)$ volume algorithm. *Journal of Computer and System Sciences*, 72(2):392–417, 2006. JCSS FOCS 2003 Special Issue.
- [17] Zhou Lu, Nataly Brukhim, Paula Gradu, and Elad Hazan. Projection-free adaptive regret with membership oracles. In *International Conference on Algorithmic Learning Theory*, pages 1055–1073. PMLR, 2023.
- [18] Zakaria Mhammedi. Efficient projection-free online convex optimization with membership oracle. In *Conference on Learning Theory*, pages 5314–5390. PMLR, 2022.

- [19] Zakaria Mhammedi. Online convex optimization with a separation oracle. *arXiv preprint arXiv:2410.02476*, 2024.
- [20] Marco Molinaro. Curvature of feasible sets in offline and online optimization. *arXiv preprint arXiv:2002.03213*, 2020.
- [21] Phan Phien. Some quantitative results on lipschitz inverse and implicit functions theorems. *arXiv preprint arXiv:1204.4916*, 2012.
- [22] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.
- [23] Naum Z Shor. Cut-off method with space extension in convex programming problems. *Cybernetics*, 13(1):94–96, 1977.
- [24] Pravin M Vaidya. A new algorithm for minimizing convex functions over convex sets. *Mathematical programming*, 73(3):291–341, 1996.
- [25] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*, pages 928–936, 2003.

NeurIPS Paper Checklist

1. Claims

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

Answer: [\[Yes\]](#)

Justification: All the claims made in the abstract are formally stated in Section 4 and proven in the appendix.

Guidelines:

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

2. Limitations

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

Answer: [\[Yes\]](#)

Justification: After the contribution paragraph in the introduction, we discuss the limitations of our work; we do not provide regret lower bounds or experimental validation for our algorithm.

Guidelines:

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

3. Theory assumptions and proofs

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

Answer: [Yes]

Justification: The paper contains the full set of assumptions and formal result statements. The proofs will be included in the appendix.

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [NA]

Justification: This is a purely theoretical paper.

Guidelines:

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

5. Open access to data and code

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

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

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

6. Experimental setting/details

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

Answer: [NA] .

Justification: The paper does not include experiments.

Guidelines:

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

7. Experiment statistical significance

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

Answer: [NA] .

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

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

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

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

9. Code of ethics

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

Answer: [Yes]

Justification: Reviewed code of ethics.

Guidelines:

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

10. Broader impacts

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

Answer: [NA]

Justification: No societal impact; purely theoretical paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

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

Answer: [NA]

Justification: No risks. Purely theoretical paper.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [NA]

Justification: No use of existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

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

Answer: [NA]

Justification: No release of new assets.

Guidelines:

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

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

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

Answer: [NA]

Justification: No crowdsourcing involvement.

Guidelines:

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

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

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

Answer: [NA]

Justification: No crowdsourcing or research on human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Contents of Appendix

A	Computing the Gauge Distance (Proofs of Lemma 2.1)	21
B	Ellipsoid Updates	23
C	Generalized Projections onto Ellipsoids	25
D	ONS Analysis (Proof of Lemma 4.3)	30
E	OCO Analysis	33
E.1	Algorithm Invariants (Proofs of Lemma 4.1 and Lemma 4.2)	33
E.2	OCO Regret (Proofs of Theorem 4.1)	37
F	Rates for Stochastic Convex Optimization	38
G	Helper Lemmas	40

A Computing the Gauge Distance (Proofs of Lemma 2.1)

For the proof of Lemma 2.1, we need the next intermediate result showing that the output (S, s) of Algorithm 1 is such that $(u - c)^\top s \leq 1$ for all $u \in \mathcal{C}$ and input vector c .

Lemma A.1. *Let $\delta \in (0, 1)$, $w, c \in \mathbb{R}^d$, $\mathcal{C} \subseteq \mathbb{R}^d$ convex, and $H \in \mathbb{R}^{d \times d}$ PSD be given, and consider a call to Algorithm 1 with input $(\mathcal{C}, H, c, w, \delta)$. Then, the output (S, s) of Algorithm 1 satisfies*

$$\forall u \in \mathcal{C}, \quad (u - c)^\top s \leq 1. \quad (11)$$

Proof. We first establish (11). If $w \in \mathcal{C}$, then the “if” condition on Line 2 evaluates to “true”, and so the algorithm returns the pair $(S, s) = (0, 0)$, which clearly satisfies (11). Now, if $c \notin \mathcal{C}$, then the “if” condition on Line 6 evaluates to “true”, and so the algorithm returns the pair $(S, s) = (0, 3dv/\sqrt{v^\top H v})$, where v is the unit-norm vector corresponding to the hyperplane separating c from \mathcal{C} . Thus, by definition of a separating hyperplane, we have that for all $u \in \mathcal{C}$, $u^\top v \leq c^\top v$, which implies (11) after using that $s = 3dv/\sqrt{v^\top H v}$.

Now, suppose that $w \notin \mathcal{C}$ and $c \in \mathcal{C}$. In this case, Algorithm 1 returns (S, s) , where $s = \frac{v}{\beta \cdot (w - c)^\top v}$ and v is the vector returned by the last call to $\text{Sep}_{\mathcal{C}}(\mu(w - c) + c)$ before the algorithm returns; here, β and μ are as in Algorithm 1. We first verify that $\beta \cdot (w - c)^\top v > 0$ so that s is well defined. Since v is the vector returned by the last call to $\text{Sep}_{\mathcal{C}}(\mu(w - c) + c)$, we have

$$\forall u \in \mathcal{C}, \quad (u - c)^\top v < \mu(w - c)^\top v. \quad (12)$$

Instantiating this with $u = c$ and using that $\mu = \frac{\alpha + \beta}{2} \leq \beta$, we get that

$$0 < \beta(w - c)^\top v,$$

and so s is well-defined in \mathbb{R}^d . Dividing both sides of (12) by $\beta \cdot (w - c)^\top v$ and using that $s = \frac{v}{\beta \cdot (w - c)^\top v}$, we get

$$\forall u \in \mathcal{C}, \quad (u - c)^\top s \leq \mu(w - c)^\top s = \frac{\mu \cdot (w - c)^\top v}{\beta \cdot (w - c)^\top v} \leq 1,$$

where the last inequality follows by the fact that $\mu = \frac{\alpha + \beta}{2} \leq \beta$ (see Algorithm 1). \square

Proof of Lemma 2.1. We consider cases.

Case where $c \notin \mathcal{C}$. If $c \notin \mathcal{C}$, then the “if” condition on Line 6 evaluates to “true”, and so the algorithm returns the pair $(S, s) = (0, 3dv/\sqrt{v^\top H v})$, where v is the vector returned by the call to $\text{Sep}_{\mathcal{C}}(c)$ on Line 5. In this case, we clearly have that $\|H^{1/2}s\| > 2d$.

For the rest of this proof, we assume that $c \in \mathcal{C}$.

Case where $c, w \in \mathcal{C}$. If $w \in \mathcal{C}$, then the “if” condition on Line 2 of Algorithm 1 evaluates to “true”, and so the algorithm returns the pair $(S, s) = (0, 0)$. Since $w \in \mathcal{C}$, we have $\gamma_{\mathcal{C}-c}(w - c) = \inf\{\lambda \geq 0 \mid w - c \in \lambda \cdot (\mathcal{C} - c)\} \leq 1$, and so, we have for all $u \in \mathbb{R}^d$:

$$S_{\mathcal{C}-c}(u - c) = \max(0, \gamma_{\mathcal{C}-c}(u - c) - 1) \geq 0 = \max(0, \gamma_{\mathcal{C}-c}(w - c) - 1) = S_{\mathcal{C}-c}(w - c).$$

This implies the desired claim.

Case where $c \in \mathcal{C}$ and $w \notin \mathcal{C}$: approximate gauge value. For the rest of this proof, we assume that $c \in \mathcal{C}$ and $w \notin \mathcal{C}$, and let α, β, μ, v, S , and s be as in Algorithm 1 when the algorithm returns.

If the condition on Line 12 of Algorithm 1 is satisfied, then by definition of s in Algorithm 1, we have that

$$\|H^{1/2}s\| > 2d.$$

Moving forward, we consider the alternative case where $\|H^{1/2}s\| \leq 2d$, which, by contrapositive, implies that the condition on Line 12 is never satisfied during the call to Algorithm 1. Thus, by design, when Algorithm 1 returns, we have

$$\alpha(w - c) + c \in \mathcal{C}, \quad \beta(w - c) + c \notin \mathcal{C}, \quad \text{and} \quad |\beta - \alpha| \leq \frac{\delta}{8d^2}.$$

Since $\gamma_{C-c}(\mathbf{u}) = \inf\{\lambda > 0 \mid \mathbf{u} \in \lambda \cdot (C - \mathbf{c})\}$, for all $\mathbf{u} \in \mathbb{R}^d$, we have that

$$\frac{1}{\beta} \leq \gamma_{C-c}(\mathbf{w} - \mathbf{c}) \leq \frac{1}{\alpha}, \quad (13)$$

with the convention that $1/0 = +\infty$. By definition of \mathbf{s} in Algorithm 1, we also have that

$$\begin{aligned} \frac{1}{\beta} &= \mathbf{s}^\top (\mathbf{w} - \mathbf{c}), \\ &\leq \|H^{1/2} \mathbf{s}\| \cdot \|H^{-1/2} (\mathbf{w} - \mathbf{c})\|, \quad (\text{H\"older's inequality}) \\ &\leq 2d, \end{aligned} \quad (14)$$

where the last inequality follows by the fact that $\|H^{1/2} \mathbf{s}\| \leq 2d$ and the assumption that $\mathbf{w} \in \mathcal{E}$ in the lemma statement. Using (14), $|\beta - \alpha| \leq \frac{\delta}{8d^2}$, and the fact that $\delta \in (0, 1)$ implies that $\alpha > 0$. Further, (14) with $|\beta - \alpha| \leq \frac{\delta}{8d^2}$ also imply that

$$\begin{aligned} \frac{1}{\alpha} &\leq \frac{1}{\beta - \frac{\delta}{8d^2}}, \\ &\leq \frac{1}{\beta} + \frac{\delta}{4\beta^2 d^2}, \quad (\text{see below}) \end{aligned} \quad (15)$$

$$\leq \frac{1}{\beta} + \delta, \quad (\text{by (14)}) \quad (16)$$

where (15) follows by the fact that $\frac{1}{1-x} \leq 1 + 2x$, for all $x \leq \frac{1}{2}$; we instantiate the latter with $x = \frac{\delta}{8\beta d^2}$ which clearly satisfies $x \leq \frac{1}{2}$ since $\frac{1}{\beta} \leq 2d$ and $\delta \in (0, 1)$. Combining (16) with (13) and using that $S = \alpha^{-1} - 1$ (see Algorithm 1), we get

$$\gamma_{C-c}(\mathbf{w} - \mathbf{c}) \leq S + 1 \leq \gamma_{C-c}(\mathbf{w} - \mathbf{c}) + \delta.$$

This together with the facts that $S_{C-c}(\mathbf{w} - \mathbf{c}) = \max(0, \gamma_{C-c}(\mathbf{w} - \mathbf{c}) - 1)$ and $\gamma_{C-c}(\mathbf{w} - \mathbf{c}) \geq 1$ (since $\mathbf{w} \notin C$ and $\mathbf{c} \in C$) implies that $S_{C-c}(\mathbf{w} - \mathbf{c}) \leq S \leq S_{C-c}(\mathbf{w} - \mathbf{c}) + \delta$, as desired.

Case where $\mathbf{c} \in C$ and $\mathbf{w} \notin C$: approximate gauge subgradient. Still in the same case as the previous paragraph, we now show that the second output \mathbf{s} of Algorithm 1 is an approximate subgradient of the gauge distance function at \mathbf{w} .

By Lemma A.1, we have

$$\forall \mathbf{u} \in C, \quad (\mathbf{u} - \mathbf{c})^\top \mathbf{s} \leq \mu(\mathbf{w} - \mathbf{c})^\top \mathbf{s} \leq 1.$$

This implies that $\mathbf{s} \in (C - \mathbf{c})^\circ$ (by definition of the polar set) and so by Lemma G.5.a, we have

$$\forall \mathbf{u} \in \mathbb{R}^d, \quad \mathbf{s}^\top \mathbf{u} \leq \sup_{\mathbf{x} \in (C - \mathbf{c})^\circ} \mathbf{x}^\top \mathbf{u} = \gamma_{C-c}(\mathbf{u}). \quad (17)$$

On the other hand, combining (16) with (13), we get

$$\gamma_{C-c}(\mathbf{w} - \mathbf{c}) - \delta \leq \frac{1}{\beta} = \mathbf{s}^\top (\mathbf{w} - \mathbf{c}), \quad (18)$$

where the equality uses the expression of \mathbf{s} . Combining (17) and (18) implies that

$$\forall \mathbf{u} \in \mathbb{R}^d, \quad \mathbf{s}^\top (\mathbf{u} - \mathbf{w} + \mathbf{c}) + \gamma_{C-c}(\mathbf{w} - \mathbf{c}) - \delta \leq \gamma_{C-c}(\mathbf{u}),$$

which, in turn, implies (through the change of variable $\mathbf{u} \leftarrow \mathbf{u} - \mathbf{c}$)

$$\forall \mathbf{u} \in \mathbb{R}^d, \quad \mathbf{s}^\top (\mathbf{u} - \mathbf{w}) + \gamma_{C-c}(\mathbf{w} - \mathbf{c}) - \delta \leq \gamma_{C-c}(\mathbf{u} - \mathbf{c}).$$

Thus, subtracting 1 from both sides and using that $S_{C-c}(\mathbf{w} - \mathbf{c}) = \gamma_{C-c}(\mathbf{w} - \mathbf{c}) - 1$ (since $\mathbf{w} \notin C$ and $\mathbf{c} \in C$), we get that

$$\begin{aligned} \forall \mathbf{u} \in \mathbb{R}^d, \quad \mathbf{s}^\top (\mathbf{u} - \mathbf{w}) + S_{C-c}(\mathbf{w} - \mathbf{c}) - \delta &\leq \gamma_{C-c}(\mathbf{u} - \mathbf{c}) - 1, \\ &\leq \max(0, \gamma_{C-c}(\mathbf{u} - \mathbf{c}) - 1), \\ &= S_{C-c}(\mathbf{u}). \end{aligned}$$

Thus, we have shown that when $\|H^{1/2}\mathbf{s}\| \leq 2d$, the outputs (S, \mathbf{s}) of Algorithm 1 are such that S approximates the Gauge distance $\mathcal{S}_{\mathcal{C}-\mathbf{c}}(\mathbf{w}-\mathbf{c})$ and \mathbf{s} is an approximate subgradient of $\mathcal{S}_{\mathcal{C}-\mathbf{c}}$ at $\mathbf{w}-\mathbf{c}$, as desired

Number of oracle calls. The number of oracle calls is bounded by the number of iterations of the “while” loop on Line 10 plus the two calls to $\text{Sep}_{\mathcal{C}}$ before the while loop. Since Algorithm 1 implements a bisection, the number of iterations in the while loop is at most $\log_2(\frac{8d^2}{\delta})$. Thus, the total number of calls to the oracle is at most $\log_2(8d^2\delta^{-1}) + 2$. \square

B Ellipsoid Updates

In this section, we provide the intuition behind the ellipsoid update rule used in Algorithm 2. As discussed in Section 4.1, when the condition for updating the current ellipsoid $\mathcal{E}_t = \mathcal{E}(\mathbf{c}_t, H_t) \supseteq \mathcal{K}$ is satisfied at round t of Algorithm 2 (i.e., when Line 4 is satisfied), we have:

$$H_t^{-1/2}(\mathcal{K} - \mathbf{c}_t) \subseteq \{\mathbf{u} \in \mathbb{B}(1) \mid \mathbf{u}^\top \mathbf{v}_t \leq 1/2d\}. \quad (19)$$

We show that under this condition, an updated ellipsoid \mathcal{E}_{t+1} can be constructed such that it still contains \mathcal{K} and its volume is reduced relative to \mathcal{E}_t .

To achieve this, we consider the more abstract problem of identifying the minimum volume ellipsoid that encloses the set $\{\mathbf{u} \in \mathbb{B}(1) \mid \mathbf{u}^\top \mathbf{v} \leq \varepsilon\}$ for some unit-norm vector \mathbf{v} ; this is the same set as in (19) but with $(\mathbf{v}_t, 1/(2d))$ replaced by $(\mathbf{v}, \varepsilon)$. The construction of this ellipsoid follows a similar approach to that found in the classical ellipsoid method (see, e.g., [2]). A sketch of this construction, which mirrors the steps outlined in [2], is provided below.

Ellipsoid parameterization. Fix $t > 0$ and $\varepsilon \in (0, 1)$, and define $H_{a,b}^{-1} = a\mathbf{v}\mathbf{v}^\top + b \cdot (I_d - \mathbf{v}\mathbf{v}^\top)$, for $a, b > 0$. We will start by solving the intermediate problem of finding parameters $a, b \in \mathbb{R}$ such that the ellipsoid $\mathcal{E}_{t,a,b} := \{\mathbf{x} \in \mathbb{R}^d \mid (\mathbf{x} + t\mathbf{v})^\top H_{a,b}^{-1}(\mathbf{x} + t\mathbf{v}) \leq 1\}$ contains the set given by

$$\{\mathbf{u} \in \mathbb{B}(1) \mid \mathbf{v}^\top \mathbf{u} \leq \varepsilon\}.$$

We will derive expressions for a and b as a function of t , then tune t so that the corresponding ellipsoid has the smallest volume.

Choosing a and b . Observe that to satisfy the requirement that $\{\mathbf{u} \in \mathbb{B}(1) \mid \mathbf{v}^\top \mathbf{u} \leq \varepsilon\} \subseteq \mathcal{E}_{t,a,b}$, it suffices to choose parameters $a, b \in \mathbb{R}$ such that the boundary $\partial\mathcal{E}_{t,a,b}$ of the ellipsoid $\mathcal{E}_{t,a,b}$ contains $-\mathbf{v}$ (which is in $\partial\mathbb{B}(1)$) and the intersection set $\mathbb{B}(1) \cap \{\mathbf{u} \in \mathbb{R}^d \mid \mathbf{v}^\top \mathbf{u} = \varepsilon\}$. This requirement translates to the following constraints:

$$-\mathbf{v} \in \partial\mathcal{E}_{t,a,b} \quad \text{and} \quad \forall \mathbf{z} \perp \mathbf{v}, \quad \varepsilon\mathbf{v} + \sqrt{1 - \varepsilon^2}\mathbf{z} \in \partial\mathcal{E}_{t,a,b}. \quad (20)$$

Solving (20) for a, b , we get

$$a = \frac{1}{(1-t)^2} \quad \text{and} \quad b = \frac{1}{1-\varepsilon^2} \cdot \left(1 - \frac{(t+\varepsilon)^2}{(t-1)^2}\right). \quad (21)$$

Tuning t for the smallest-volume ellipsoid. Now that we have derived expressions for a and b as a function of t such that $\{\mathbf{u} \in \mathbb{B}(1) \mid \mathbf{v}^\top \mathbf{u} \leq \varepsilon\} \subseteq \mathcal{E}_{t,a,b}$, we will tune t to obtain the smallest-volume ellipsoid $\mathcal{E}_{t,a,b}$. Note that the volume of the ellipsoid $\mathcal{E}_{t,a,b}$ satisfies

$$\frac{\text{vol}(\mathcal{E}_{t,a,b})}{\text{vol}(\mathbb{B}(1))} = \frac{1}{\sqrt{a}} \left(\frac{1}{\sqrt{b}}\right)^{d-1}. \quad (22)$$

For $\varepsilon = \frac{1}{2d}$ and (a, b) as in (21), the parameter t that minimizes the right-hand side of (22) is given by

$$t = \frac{1}{2(d+1)}. \quad (23)$$

This leads to $H_{a,b}^{-1} = H^{-1} = \frac{4d^2-4}{4d^2-1} \left(I_d + \frac{2d}{(2d+1)(d-1)}\mathbf{v}\mathbf{v}^\top\right)$ and by Sherman-Morrisson:

$$H_{a,b} = H = \frac{4d^2-1}{4d^2-4} \cdot \left(I_d - \frac{2d}{2d^2+d-1}\mathbf{v}\mathbf{v}^\top\right).$$

Further, by (22), it can be shown that with the choices of (a, b) and t as in (21) and (23), respectively, we have

$$\frac{\text{vol}(\mathcal{E}_{t,a,b})}{\text{vol}(\mathbb{B}(1))} \leq \exp\left(-\frac{1}{8d}\right). \quad (24)$$

We formalize this result in Lemma B.1. Before presenting this lemma, we give a few comments.

The informal sketch presented above shows that when $H_t^{-1/2}(\mathcal{K} - \mathbf{c}_t) \subseteq \{\mathbf{u} \in \mathbb{B}(1) \mid \mathbf{u}^\top \mathbf{v}_t \leq 1/2d\}$ (a condition satisfied whenever the ellipsoid update in Algorithm 2 is triggered), then

$$H_t^{-1/2}(\mathcal{K} - \mathbf{c}_t) \subseteq \tilde{\mathcal{E}}_{t+1} := \mathcal{E}\left(-\frac{\mathbf{v}_t}{2(d+1)}, \tilde{H}_{t+1}\right), \quad \text{where} \quad \tilde{H}_{t+1} := \frac{4d^2-1}{4d^2-4} \cdot \left(I_d - \frac{2d}{2d^2+d-1} \mathbf{v}_t \mathbf{v}_t^\top\right).$$

Further, it can be verified that $\mathbf{c}_t + H_t^{1/2} \tilde{\mathcal{E}}_{t+1} = \mathcal{E}(\mathbf{c}_{t+1}, H_{t+1})$, with \mathbf{c}_{t+1} and H_{t+1} as defined on Line 7 and Line 8 of Algorithm 2. Thus, $\mathcal{K} \subseteq \mathcal{E}_{t+1} := \mathcal{E}(\mathbf{c}_{t+1}, H_{t+1})$ as discussed in Section 4.1.

We also note that the choice of ε scaling inversely with d in the construction just described was necessary to show the volume decrease in (24). This puts a limit on the maximum value ε can take and contributes to the additive $\tilde{O}(d^2)$ term in the final regret bound of Algorithm 2.

We now formalize the guarantee of our construction.

Lemma B.1. *Let $\mathbf{v} \in \mathbb{R}^d$ be such that $\|\mathbf{v}\| = 1$, and define $H := \frac{4d^2-1}{4d^2-4} \cdot \left(I_d - \frac{2d}{2d^2+d-1} \mathbf{v} \mathbf{v}^\top\right)$ and*

$$\mathcal{E} := \left\{ \mathbf{u} \in \mathbb{R}^d \mid \left(\mathbf{u} + \frac{1}{2(d+1)} \mathbf{v} \right)^\top H^{-1} \left(\mathbf{u} + \frac{1}{2(d+1)} \mathbf{v} \right) \leq 1 \right\}.$$

Then, we have

$$\{\mathbf{u} \in \mathbb{B}(1) \mid \mathbf{v}^\top \mathbf{u} \leq \frac{1}{2d}\} \subseteq \mathcal{E}. \quad (25)$$

Furthermore, it holds that $\text{vol}(\mathcal{E}) \leq e^{-\frac{1}{8d}} \cdot \text{vol}(\mathbb{B}(1))$.

Proof. We first prove (25). Let H be as in the lemma statement. By Sherman-Morrisson, we have

$$H^{-1} = \frac{4d^2-4}{4d^2-1} \cdot \left(I_d + \frac{2d}{(2d+1)(d-1)} \mathbf{v} \mathbf{v}^\top \right). \quad (26)$$

Thus, for all $\mathbf{u} \in \mathbb{R}^d$, we have

$$\begin{aligned} & \frac{4d^2-1}{4d^2-4} \cdot \left(\mathbf{u} + \frac{1}{2(d+1)} \mathbf{v} \right)^\top H^{-1} \left(\mathbf{u} + \frac{1}{2(d+1)} \mathbf{v} \right) \\ &= \left(\mathbf{u} + \frac{1}{2(d+1)} \mathbf{v} \right)^\top \left(I_d + \frac{2d}{(2d+1)(d-1)} \mathbf{v} \mathbf{v}^\top \right) \left(\mathbf{u} + \frac{1}{2(d+1)} \mathbf{v} \right), \\ &= 1 + \frac{2d}{(2d+1)(d-1)} (\mathbf{u}^\top \mathbf{v})^2 + \frac{1}{4(d+1)^2} + \frac{d}{2(d+1)^2(2d+1)(d-1)} \\ & \quad + \frac{1}{d+1} \mathbf{u}^\top \mathbf{v} + \frac{2d}{(d+1)(2d+1)(d-1)} \mathbf{u}^\top \mathbf{v}. \end{aligned} \quad (27)$$

We need to prove that the right-hand side is at most 1 for all $\mathbf{u} \in \{\mathbf{x} \in \mathbb{B}(1) \mid \mathbf{v}^\top \mathbf{x} \leq \frac{1}{2d}\}$. Note that for any such \mathbf{u} , we have $\mathbf{u}^\top \mathbf{v} \in [-1, \frac{1}{2d}]$. Therefore, we have that for all $\mathbf{u} \in \{\mathbf{x} \in \mathbb{B}(1) \mid \mathbf{v}^\top \mathbf{x} \leq \frac{1}{2d}\}$:

$$\begin{aligned} & 1 + \frac{2d(\mathbf{u}^\top \mathbf{v})^2}{(2d+1)(d-1)} + \frac{1}{4(d+1)^2} + \frac{d}{2(d+1)^2(2d+1)(d-1)} + \frac{\mathbf{u}^\top \mathbf{v}}{d+1} + \frac{2d\mathbf{u}^\top \mathbf{v}}{(d+1)(2d+1)(d-1)} \\ & \leq \sup_{x \in [-1, \frac{1}{2d}]} 1 + \frac{2dx^2}{(2d+1)(d-1)} + \frac{1}{4(d+1)^2} + \frac{d}{2(d+1)^2(2d+1)(d-1)} + \frac{x}{d+1} + \frac{2dx}{(d+1)(2d+1)(d-1)}. \end{aligned}$$

Now, note that the function

$$x \mapsto 1 + \frac{2d}{(2d+1)(d-1)} x^2 + \frac{1}{4(d+1)^2} + \frac{d}{2(d+1)^2(2d+1)(d-1)} + \frac{1}{d+1} x + \frac{2d}{(d+1)(2d+1)(d-1)} x$$

is convex, and so its maximum in the interval $[-1, \frac{1}{2d}]$ must be attained at the endpoints -1 and $\frac{1}{2d}$. Therefore, we have that for all $\mathbf{u} \in \{\mathbf{x} \in \mathbb{B}(1) \mid \mathbf{v}^\top \mathbf{x} \leq \frac{1}{2d}\}$:

$$\begin{aligned} & 1 + \frac{2d(\mathbf{u}^\top \mathbf{v})^2}{(2d+1)(d-1)} + \frac{1}{4(d+1)^2} + \frac{d}{2(d+1)^2(2d+1)(d-1)} + \frac{\mathbf{u}^\top \mathbf{v}}{d+1} + \frac{2d\mathbf{u}^\top \mathbf{v}}{(d+1)(2d+1)(d-1)} \\ & \leq \max_{x \in \{-1, \frac{1}{2d}\}} 1 + \frac{2dx^2}{(2d+1)(d-1)} + \frac{1}{4(d+1)^2} + \frac{d}{2(d+1)^2(2d+1)(d-1)} + \frac{x}{d+1} + \frac{2dx}{(d+1)(2d+1)(d-1)}, \\ & = \frac{4d^2 - 1}{4d^2 - 4}. \end{aligned}$$

Combining this with (27) shows that $\left(\mathbf{u} + \frac{1}{2(d+1)}\mathbf{v}\right)^\top H^{-1} \left(\mathbf{u} + \frac{1}{2(d+1)}\mathbf{v}\right) \leq 1$ for all $\mathbf{u} \in \{\mathbf{x} \in \mathbb{B}(1) \mid \mathbf{v}^\top \mathbf{x} \leq \frac{1}{2d}\}$, and so

$$\{\mathbf{u} \in \mathbb{B}(1) \mid \mathbf{v}^\top \mathbf{u} \leq \frac{1}{2d}\} \subseteq \mathcal{E}.$$

Volume decrease. It remains to prove that $\text{vol}(\mathcal{E}) \leq e^{-\frac{1}{8d}} \cdot \text{vol}(\mathbb{B}(1))$. The volume of the ellipsoid \mathcal{E} is given by

$$\text{vol}(\mathcal{E}) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} \cdot \frac{1}{\sqrt{\det(H^{-1})}}, \quad (28)$$

where Γ is the *Gamma function*. We now compute the determinant of H^{-1} . By (26), we can write

$$H^{-1} = a\mathbf{v}\mathbf{v}^\top + b(I - \mathbf{v}\mathbf{v}^\top),$$

where $a = \frac{4(1+d)^2}{(1+d)^2}$ and $b = \frac{4d^2-4}{4d^2-1} \cdot \frac{(d-1)(2d+1)}{2d}$. Thus, for any $\mathbf{z} \perp \mathbf{v}$, we have $H^{-1}\mathbf{z} = b\mathbf{z}$. In addition, we have $H^{-1}\mathbf{v} = a\mathbf{v}$ because $\|\mathbf{v}\| = 1$. Therefore, a and b are the only eigenvalues of H^{-1} with b having multiplicity $d-1$, and so $\det(H^{-1}) = a^{d-1}b$. Therefore, by (28)

$$\text{vol}(\mathcal{E}) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} \cdot \frac{1}{\sqrt{a}} \cdot \left(\frac{1}{\sqrt{b}}\right)^{d-1} = \frac{\pi^{d/2}}{\Gamma(d/2+1)} \cdot \frac{2^{-d} \left(\frac{d^2-1}{4d^2-1}\right)^{\frac{1-d}{2}}}{\frac{(d+1)}{(2d+1)}} \leq \frac{\pi^{d/2}}{\Gamma(d/2+1)} \cdot e^{-\frac{1}{8d}}, \quad (29)$$

where the last inequality follows from the fact that

$$\frac{2^{-d} \left(\frac{d^2-1}{4d^2-1}\right)^{\frac{1-d}{2}}}{\frac{(d+1)}{(2d+1)}} \leq e^{-\frac{1}{8d}}, \quad \forall d > 1,$$

and we are assuming that $d \geq 2$ in this paper (see Section 2.1). Combining (29) with the fact that $\text{vol}(\mathbb{B}(1)) = \pi^{d/2}/\Gamma(d/2+1)$ completes the proof. \square

C Generalized Projections onto Ellipsoids

In this section, we consider the generalized projection problem

$$\mathbf{u}_* \in \arg \min_{\mathbf{u} \in \mathcal{E}} (\mathbf{u} - \mathbf{z})^\top Q^{-1}(\mathbf{u} - \mathbf{z}), \quad (30)$$

for $\mathbf{z} \in \mathbb{R}^d$ and $Q \in \mathbb{S}_{>0}^{d \times d}$ in the special case where \mathcal{E} is an ellipsoid; that is:

$$\mathcal{E} = \{\mathbf{x} \in \mathbb{R}^d \mid (\mathbf{x} - \mathbf{c})^\top H^{-1}(\mathbf{x} - \mathbf{c}) \leq 1\},$$

for some $H \in \mathbb{S}_{>0}^{d \times d}$ and $\mathbf{c} \in \mathbb{R}^d$. By Lagrangian duality, the problem in (30) is equivalent to the max-min problem:

$$\sup_{\lambda \geq 0} \inf_{\mathbf{u} \in \mathbb{R}^d} (\mathbf{z} - \mathbf{u})^\top Q^{-1}(\mathbf{z} - \mathbf{u}) + \lambda \cdot ((\mathbf{u} - \mathbf{c})^\top H^{-1}(\mathbf{u} - \mathbf{c}) - 1). \quad (31)$$

Algorithm 3 PoE: Projection onto ellipsoid via binary search.

require: $\mathbf{z}, \mathbf{c} \in \mathbb{R}^d$, $Q, H \in \mathbb{R}^{d \times d}$, and parameters $\lambda_-, \lambda_+, \delta > 0$.
/ If \mathbf{z} is almost in the ellipsoid $\mathcal{E}(\mathbf{c}, H)$, then return a slightly 'scaled down' version of it $(\mathbf{z} - \mathbf{c})/(1 + \delta) + \mathbf{c}$. */*
1: **if** $(\mathbf{z} - \mathbf{c})^\top H^{-1}(\mathbf{z} - \mathbf{c}) \leq 1 + \delta$ **then**
2: **return** $(\mathbf{z} - \mathbf{c})/(1 + \delta) + \mathbf{c}$.
/ Approximate $\mathbf{u}^* \leftarrow \arg \min_{\mathbf{u} \in \mathcal{E}(\mathbf{c}, H)} (\mathbf{z} - \mathbf{u})^\top Q^{-1}(\mathbf{z} - \mathbf{u})$ via binary search */*
3: Set $\alpha \leftarrow \lambda_-$, $\beta \leftarrow \lambda_+$, and $\mu \leftarrow (\alpha + \beta)/2$.
4: Set $\mathbf{u}_\mu^* = (HQ^{-1} + \mu I)^{-1}(HQ^{-1}\mathbf{z} + \mu\mathbf{c})$.
5: Set $D = (\mathbf{u}_\mu^* - \mathbf{c})^\top H^{-1}(\mathbf{u}_\mu^* - \mathbf{c})$.
6: **while** $D \notin [1 - \delta, 1]$ **do**
7: **if** $D < 1 - \delta$ **then**
8: Set $\beta \leftarrow \mu$ and $\mu \leftarrow (\alpha + \beta)/2$.
9: **else**
10: Set $\alpha \leftarrow \mu$ and $\mu \leftarrow (\alpha + \beta)/2$.
11: Set $\mathbf{u}_\mu^* = (HQ^{-1} + \mu I)^{-1}(HQ^{-1}\mathbf{z} + \mu\mathbf{c})$.
12: Set $D = (\mathbf{u}_\mu^* - \mathbf{c})^\top H^{-1}(\mathbf{u}_\mu^* - \mathbf{c})$.
13: **return** \mathbf{u}_μ^* .

For a fixed $\lambda \geq 0$, the inner minimization problem has a closed-form solution:

$$\mathbf{u}_\lambda^* = (HQ^{-1} + \lambda I)^{-1}(HQ^{-1}\mathbf{z} + \lambda\mathbf{c}).$$

By leveraging this structure, we show that the problem in (31) can be solved efficiently via binary search over the dual variable λ . In this paper, we do this using Algorithm 3. This algorithm takes inputs $\mathbf{z}, \mathbf{c}, Q$, and H , and outputs an approximate solution to the problem in (30). We now state the guarantee of the algorithm.

Lemma C.1 (Projection onto ellipsoid). *Let $R > 0$, $\delta > 0$ be given. Let $\mathbf{z}, \mathbf{c} \in \mathbb{B}(R)$, and $Q, H \in \mathbb{S}_{>0}^{d \times d}$ be such that $(\mathbf{z} - \mathbf{c})^\top H^{-1}(\mathbf{z} - \mathbf{c}) > 1 + \delta$ and $\sigma_{\min}(H) \leq R^2$. Further, define $\mathcal{E}(\nu) := \{\mathbf{u} \in \mathbb{R}^d \mid (\mathbf{u} - \mathbf{c})^\top H^{-1}(\mathbf{u} - \mathbf{c}) \leq 1 - \nu\}$, for $\nu \in \mathbb{R}$, and consider a call to Algorithm 3 with input $(\mathbf{z}, \mathbf{c}, Q, H, \lambda_-, \lambda_+, \delta)$, where $0 \leq \lambda_- \leq \delta \cdot \frac{\sigma_{\min}(H)^2}{24R^2 \cdot \sigma_{\max}(Q)}$ and $\lambda_+ \geq \frac{4\sigma_{\max}(H) \cdot R}{\sigma_{\min}(H)^{1/2} \cdot \sigma_{\min}(Q)}$. Then, Algorithm 3 returns $\mathbf{u}^* \in \mathbb{R}^d$ such that:*

$$\exists \mu \in [0, \delta] : \quad \mathbf{u}^* \in \arg \min_{\mathbf{u} \in \mathcal{E}(\mu)} (\mathbf{u} - \mathbf{z})^\top Q^{-1}(\mathbf{u} - \mathbf{z}).$$

The number of iterations in the while loop on Line 6 of Algorithm 3 is at most $O(\log_2(\frac{L\lambda_+}{\delta\lambda_-}))$, where

$$L := 4(B + R) \cdot \frac{\sigma_{\max}(H) \cdot \sigma_{\max}(Q)^2}{\sigma_{\min}(H)^3 \cdot \sigma_{\min}(Q)} \cdot R, \quad \text{and} \quad B := R \cdot \frac{\frac{\sigma_{\max}(H)}{\sigma_{\min}(Q)} + \lambda_+}{\frac{\sigma_{\min}(H)}{\sigma_{\max}(Q)}}.$$

Finally, each iteration of the while loop in Algorithm 3 can be performed in $O(d^\omega)$ time, where ω is the matrix multiplication exponent.

Remark C.1 (Computational cost of PoE). *By Lemma C.1, the number of iterations in the while-loop of Algorithm 3 is bounded by $\log(\frac{L\lambda_+}{\delta\lambda_-})$. We later show that for our application, we have $\frac{L\lambda_+}{\delta\lambda_-} \leq \text{poly}(\frac{R}{r}, T)$. Note also that the cost per iteration of Algorithm 3 is bounded by the cost of solving a linear system (which costs $O(d^\omega)$), and so total cost of running Algorithm 3 is bounded by $O(d^\omega \log \frac{L\lambda_+}{\delta\lambda_-})$. It is possible to implement Algorithm 3 so that the total cost is bounded by $O(d^2 \log \frac{L\lambda_+}{\delta\lambda_-} + d^3)$ instead, where now the dominant term $O(d^3)$ is independent of any logarithmic factor.³ This can be done as follows:*

³Even though $O(d^\omega \log \frac{L\lambda_+}{\delta\lambda_-})$ is technically better than $O(d^2 \log \frac{L\lambda_+}{\delta\lambda_-} + d^3)$ asymptotically (since $\omega < 3$), the $O(\cdot)$ notation in the former typically hides large constants making the new implementation described in the remark more favorable in practice.

- Compute Cholesky decompositions of Q and H (costs $O(d^3)$); that is, compute lower-triangular matrices L_Q and L_H such that $Q = L_Q L_Q^\top$ and $H = L_H L_H^\top$.
- Compute SVD decomposition of $L_H^\top L_Q^{-\top} L_Q^{-1} L_H$ (costs $O(d^3)$); that is, compute (Λ, U) such that $M = U \Lambda U^\top$, $U U^\top = I$, and $\Lambda = \text{diag}(\rho_1, \dots, \rho_d)$.
- Compute inverses L_H^{-1} and Q^{-1} (costs $O(d^3)$).

With this, we have for any $\mu \geq 0$:

$$(H Q^{-1} + \mu I)^{-1} (H Q^{-1} \mathbf{z} + \mu \mathbf{c}) = L_H U \text{diag}\left(\frac{1}{\rho_1 + \mu}, \dots, \frac{1}{\rho_d + \mu}\right) U^\top L_H^{-1} (H Q^{-1} \mathbf{z} + \mu \mathbf{c}).$$

Thus, given $(L_H, L_Q, L_H^{-1}, Q^{-1}, \Lambda, U)$, we can compute $(H Q^{-1} + \mu I)^{-1} (H Q^{-1} \mathbf{z} + \mu \mathbf{c})$ in $O(d^2)$ (matrix-vector multiplication costs) for any $\mu \geq 0$. Thus, the total cost PoE with this implementation is bounded by $O(d^2 \log \frac{L \lambda_+}{\delta \lambda_-} + d^3)$ because the operations described in the bullet points (which cost $O(d^3)$) need to be performed only once.

The computational cost of a Euclidean projection onto an arbitrary set \mathcal{K} can be much worse than that of PoE in Remark C.1. For example, using state-of-the-art ellipsoid methods to project a point onto a set \mathcal{K} specified by a separation oracle can incur a cost of up to $\tilde{O}(d \cdot C_{\text{sep}}(\mathcal{K}) + d^3)$, where C_{sep} denotes the cost of a single separation oracle call [15]. Moreover, the $\tilde{O}(\cdot)$ notation often conceals large constants, which can render these methods impractical. Alternatively, a Euclidean projection can be formulated as a quadratic program and solved using an interior point method. This approach requires a self-concordant barrier for the set \mathcal{K} whose gradients and Hessians are inexpensive to compute. However, even under favorable conditions, the associated cost typically remains higher than the costs detailed in Remark C.1.

Proof of Lemma C.1. Let $\mathbf{z}, \mathbf{c} \in \mathbb{B}(R)$ and $Q, H \in \mathbb{S}_{>0}^{d \times d}$ be given. For $\lambda \geq 1$, let us define

$$\mathbf{u}_\lambda^* \in \arg \min_{\mathbf{u} \in \mathbb{R}^d} (\mathbf{u} - \mathbf{z})^\top Q^{-1} (\mathbf{u} - \mathbf{z}) + \lambda \cdot ((\mathbf{u} - \mathbf{c})^\top H^{-1} (\mathbf{u} - \mathbf{c}) - 1).$$

Setting the gradient of the objective to zero and solving for \mathbf{u} , we obtain

$$\mathbf{u}_\lambda^* = (Q^{-1} + \lambda H^{-1})^{-1} (Q^{-1} \mathbf{z} + \lambda H^{-1} \mathbf{c}). \quad (32)$$

We now study how the “constraint” objective $g(\lambda) := (\mathbf{u}_\lambda^* - \mathbf{c})^\top H^{-1} (\mathbf{u}_\lambda^* - \mathbf{c})$ varies as a function of λ . Taking the derivative of g gives

$$g'(\lambda) = 2(\mathbf{u}_\lambda^* - \mathbf{c})^\top H^{-1} \frac{d\mathbf{u}_\lambda^*}{d\lambda}. \quad (33)$$

On the other hand, by the expression of \mathbf{u}_λ^* in (32), we have

$$\begin{aligned} \frac{d\mathbf{u}_\lambda^*}{d\lambda} &= (Q^{-1} + \lambda H^{-1})^{-1} H^{-1} \mathbf{c} - (Q^{-1} + \lambda H^{-1})^{-1} H^{-1} (Q^{-1} + \lambda H^{-1})^{-1} (Q^{-1} \mathbf{z} + \lambda H^{-1} \mathbf{c}), \\ &= (Q^{-1} + \lambda H^{-1})^{-1} H^{-1} \mathbf{c} - (Q^{-1} + \lambda H^{-1})^{-1} H^{-1} \mathbf{x}_\lambda^*, \\ &= (Q^{-1} + \lambda H^{-1})^{-1} H^{-1} (\mathbf{c} - \mathbf{x}_\lambda^*). \end{aligned} \quad (34)$$

Plugging this into (33), we get that

$$g'(\lambda) = -2(\mathbf{u}_\lambda^* - \mathbf{c})^\top H^{-1} (Q^{-1} + \lambda H^{-1})^{-1} H^{-1} (\mathbf{u}_\lambda^* - \mathbf{c}) \leq 0,$$

where the last inequality follows by the fact that the matrix $H^{-1} (Q^{-1} + \lambda H^{-1})^{-1} H^{-1}$ is positive definite. Therefore, $g(\lambda)$ is non-increasing in λ , and so we can find λ^* using binary search. To show that this can be done efficiently, it remains to identify a reasonably small interval for the values of λ^* .

Upper bound on λ^* . Note that

$$\begin{aligned} \mathbf{u}_\lambda^* &= (Q^{-1} + \lambda H^{-1})^{-1} (Q^{-1} \mathbf{z} + \lambda H^{-1} \mathbf{c}), \\ &= (H Q^{-1} / \lambda + I)^{-1} (H Q^{-1} \mathbf{z} / \lambda + \mathbf{c}), \\ &= (H Q^{-1} / \lambda + I)^{-1} H Q^{-1} \mathbf{z} / \lambda + \mathbf{c} - (I - (H Q^{-1} / \lambda + I)^{-1}) \mathbf{c}, \\ &= \mathbf{c} + (I - (H Q^{-1} / \lambda + I)^{-1}) (\mathbf{z} - \mathbf{c}). \end{aligned} \quad (35)$$

For $\lambda \geq \frac{4\sigma_{\max}(H)R}{\sigma_{\min}(H)^{1/2}\sigma_{\min}(Q)}$, we have by the stability of the inverse operator (Lemma G.1)

$$\|(HQ^{-1}/\lambda + I)^{-1} - I\|_{\text{op}} \leq \frac{4}{3\lambda} \cdot \|HQ^{-1}\|_{\text{op}} \leq \frac{\sigma_{\min}(H)^{1/2}}{3R}, \quad (36)$$

where we used that $\sigma_{\min}(H)^{1/2} \leq R$, by assumption. Using (35) and (36), we get

$$\begin{aligned} g(\lambda) &= (\mathbf{u}_\lambda^* - \mathbf{c})^\top H^{-1}(\mathbf{u}_\lambda^* - \mathbf{c}), \\ &= (\mathbf{z} - \mathbf{c})^\top (I - (HQ^{-1}/\lambda + I)^{-1}) H^{-1} (I - (HQ^{-1}/\lambda + I)^{-1}) (\mathbf{z} - \mathbf{c}), \\ &\leq \|(HQ^{-1}/\lambda + I)^{-1} - I\|_{\text{op}}^2 \cdot \|H^{-1}\|_{\text{op}} \cdot \|\mathbf{z} - \mathbf{c}\|^2, \\ &\leq \frac{1}{9R^2} \cdot \|\mathbf{z} - \mathbf{c}\|^2, \\ &\leq 4/9 < 1, \end{aligned}$$

where the last inequality uses that $\mathbf{z}, \mathbf{c} \in \mathbb{B}(R)$. Thus, for all $\lambda \geq \frac{4\sigma_{\max}(H)R}{\sigma_{\min}(H)^{1/2}\sigma_{\min}(Q)}$, we have $g(\lambda) \leq 1$. Since $g(\lambda)$ is non-increasing, and we know that $g(\lambda^*) = 1$, we must have that

$$\lambda^* \leq \frac{4\sigma_{\max}(H)R}{\sigma_{\min}(H)^{1/2}\sigma_{\min}(Q)} \leq \frac{\|QH^{-1}\|_{\text{op}}^2}{2}.$$

Lower bound on λ^* . For the other direction, we have

$$(HQ^{-1}/\lambda + I)^{-1} = \lambda \cdot (HQ^{-1} + \lambda I)^{-1}.$$

And so, by Lemma G.1, as long as $\lambda \leq \|QH^{-1}\|_{\text{op}}^{-1}/2$, we have

$$\|(HQ^{-1} + \lambda I)^{-1} - QH^{-1}\|_{\text{op}} \leq 2\lambda \|QH^{-1}\|_{\text{op}}^2. \quad (37)$$

Moving forward, we let $E = \lambda QH^{-1} - \lambda(HQ^{-1} + \lambda I)^{-1}$ and assume that $\lambda \leq \lambda^*$ (recall that $\lambda^* \leq \|QH^{-1}\|_{\text{op}}^{-1}/2$). With this and (35), we have that

$$\begin{aligned} g(\lambda) &= (\mathbf{u}_\lambda^* - \mathbf{c})^\top H^{-1}(\mathbf{u}_\lambda^* - \mathbf{c}), \\ &= (\mathbf{z} - \mathbf{c})^\top (I - \lambda QH^{-1} + E) H^{-1} (I - \lambda QH^{-1} + E) (\mathbf{z} - \mathbf{c}). \end{aligned} \quad (38)$$

Now, let $E' := -\lambda QH^{-1} + E$, and note that by the triangle inequality and (37), we have

$$\|E'\|_{\text{op}} \leq \lambda \|QH^{-1}\|_{\text{op}} + 2\lambda^2 \|QH^{-1}\|_{\text{op}}^2 \leq 2\lambda \|QH^{-1}\|_{\text{op}}, \quad (39)$$

where the last inequality follows from the fact that $\lambda \leq \|QH^{-1}\|_{\text{op}}^{-1}/2$. On the other hand, by (38):

$$\begin{aligned} g(\lambda) &= (\mathbf{z} - \mathbf{c})^\top (I + E') H^{-1} (I + E') (\mathbf{z} - \mathbf{c}), \\ &= (\mathbf{z} - \mathbf{c})^\top H^{-1} (\mathbf{z} - \mathbf{c}) + 2(\mathbf{z} - \mathbf{c})^\top E' H^{-1} (\mathbf{z} - \mathbf{c}) + (\mathbf{z} - \mathbf{c})^\top E' H^{-1} E' (\mathbf{z} - \mathbf{c}). \end{aligned}$$

Thus, by (39) and the fact that $\mathbf{z}, \mathbf{c} \in \mathbb{B}(R)$, we have that

$$\begin{aligned} |g(\lambda) - (\mathbf{z} - \mathbf{c})^\top H^{-1} (\mathbf{z} - \mathbf{c})| &\leq \frac{16R^2\lambda \|QH^{-1}\|_{\text{op}}}{\sigma_{\min}(H)} + \frac{16R^2\lambda^2 \|QH^{-1}\|_{\text{op}}^2}{\sigma_{\min}(H)}, \\ &\leq \frac{24R^2\lambda}{\sigma_{\min}(H)} \|QH^{-1}\|_{\text{op}}, \quad (\text{since } \lambda \leq \|QH^{-1}\|_{\text{op}}^{-1}/2) \\ &\leq \frac{24R^2\lambda \cdot \sigma_{\max}(Q)}{\sigma_{\min}(H)^2}. \end{aligned}$$

Therefore, if $\lambda \leq \delta \cdot \frac{\sigma_{\min}(H)^2}{24R^2\sigma_{\max}(Q)}$, we get that

$$|g(\lambda) - (\mathbf{z} - \mathbf{c})^\top H^{-1} (\mathbf{z} - \mathbf{c})| \leq \delta.$$

Thus, we must have

$$\delta \cdot \frac{\sigma_{\min}(H)^2}{24R^2 \cdot \sigma_{\max}(Q)} \leq \lambda^* \leq \frac{4\sigma_{\max}(H) \cdot R}{\sigma_{\min}(H)^{1/2} \cdot \sigma_{\min}(Q)},$$

and so by the assumptions on λ_- and λ_+ it holds that

$$0 < \lambda_- \leq \lambda^* \leq \lambda_+.$$

Now that we have established that λ^* is in between λ_- and λ_+ , we can use binary search to find λ^* such that $g(\lambda^*) \approx 1$. The number of iterations of the binary search will depend on the ratio λ_+/λ_- , the precision δ , and the Lipschitz constant of g .

Lipschitz constant of g . Since $\mathbf{u}_\lambda^* = (HQ^{-1} + \lambda I)^{-1}(HQ^{-1}\mathbf{z} + \lambda \mathbf{c})$ and $\mathbf{z}, \mathbf{c} \in \mathbb{B}(R)$, we have that

$$\begin{aligned}\|\mathbf{u}_\lambda^*\| &\leq \|HQ^{-1} + \lambda I\|_{\text{op}}^{-1} \cdot (R\|HQ^{-1}\|_{\text{op}} + \lambda_+ R), \\ &= \frac{1}{\sigma_{\min}(HQ^{-1} + \lambda I)} \cdot (R\|HQ^{-1}\|_{\text{op}} + \lambda_+ R), \\ &\leq B := R \cdot \frac{\sigma_{\max}(HQ^{-1}) + \lambda_+}{\sigma_{\min}(HQ^{-1})}.\end{aligned}$$

On the other hand, we have that

$$g'(\lambda) = 2(\mathbf{u}_\lambda^* - \mathbf{c})^\top H^{-1} \frac{d\mathbf{u}_\lambda^*}{d\lambda},$$

and from (34)

$$\begin{aligned}\frac{d\mathbf{u}_\lambda^*}{d\lambda} &= (Q^{-1} + \lambda H^{-1})^{-1} H^{-1} (Q^{-1} + \lambda H^{-1})^{-1} Q^{-1} (\mathbf{z} - \mathbf{c}), \\ &= (HQ^{-1} + \lambda I)^{-1} (HQ^{-1} + \lambda I)^{-1} HQ^{-1} (\mathbf{z} - \mathbf{c}), \\ &= (HQ^{-1} + \lambda I)^{-2} HQ^{-1} (\mathbf{z} - \mathbf{c}).\end{aligned}$$

Therefore, by the triangle inequality and the fact that $\mathbf{z}, \mathbf{c} \in \mathbb{B}(R)$, we get

$$\begin{aligned}|g'(\lambda)| &\leq \frac{4(B+R)}{\sigma_{\min}(H)} \cdot \frac{1}{\sigma_{\min}(HQ^{-1} + \lambda_- I)^2} \cdot \|HQ^{-1}\|_{\text{op}} \cdot R, \\ &\leq L := \frac{4(B+R)}{\sigma_{\min}(H)} \cdot \frac{\sigma_{\max}(HQ^{-1})}{\sigma_{\min}(HQ^{-1})^2} \cdot R.\end{aligned}$$

Number of iterations of binary search. Given that

- $0 < \lambda_- \leq \lambda^* \leq \lambda_+$,
- $g(\lambda_+) \leq g(\lambda^*) = 1$, and
- g is non-increasing in λ and L -Lipschitz,

Algorithm 3 finds a $\mu \in [\lambda_-, \lambda_+]$ such that

$$1 - \delta \leq g(\mu) = (\mathbf{u}_\mu^* - \mathbf{c})^\top H^{-1} (\mathbf{u}_\mu^* - \mathbf{c}) \leq 1, \quad (40)$$

after at most $N = O(1) \cdot \log_2(\frac{\lambda_+ L}{\lambda_- \delta})$ iterations of binary search on Line 6. Note that as soon as Algorithm 3 finds such a μ , it returns $\mathbf{u} = \mathbf{u}_\mu^*$ (see Line 6 and Line 13 of Algorithm 3).

Optimality of \mathbf{u}^* . Let $\mu := 1 - (\mathbf{u}^* - \mathbf{c})^\top H^{-1} (\mathbf{u}^* - \mathbf{c})$, where $\mathbf{u}^* = \mathbf{u}_\mu^*$ is the vector returned by the call to Algorithm 3, and note that we have just shown that $\mu \in [0, \delta]$ (see (40)). Now, consider the problem in the lemma statement:

$$\min_{\mathbf{u} \in \mathcal{E}(\nu)} (\mathbf{u} - \mathbf{z})^\top Q^{-1} (\mathbf{u} - \mathbf{z}).$$

This problem can equivalently be written as:

$$\min_{\mathbf{u}: f(\mathbf{u}) \leq 1-\nu} f_0(\mathbf{u}), \quad (41)$$

where $f_0(\mathbf{u}) := (\mathbf{u} - \mathbf{z})^\top Q^{-1} (\mathbf{u} - \mathbf{z})$ and $f(\mathbf{u}) := (\mathbf{u} - \mathbf{c})^\top H^{-1} (\mathbf{u} - \mathbf{c})$. By the definition of \mathbf{u}^* , it can be verified that

$$\nabla f_0(\mathbf{u}^*) + \mu \cdot \nabla f(\mathbf{u}^*) = \mathbf{0}.$$

Furthermore, we have $f(\mathbf{u}^*) = g(\mu) = 1 - \nu$ (by (40) and the definition of ν). Therefore, the primal-dual pair (\mathbf{u}^*, μ) satisfies the KKT conditions for the convex problem in (41), and so we have

$$\mathbf{u}^* \in \arg \min_{\mathbf{u} \in \mathcal{E}(\nu)} (\mathbf{u} - \mathbf{z})^\top Q^{-1} (\mathbf{u} - \mathbf{z}),$$

as desired. This completes the proof of Lemma C.1. \square

D ONS Analysis (Proof of Lemma 4.3)

Proof. Let $(H_t, \mathbf{c}_t, \mathbf{z}_t, \mathbf{u}_t, \Sigma_t, \tilde{\mathbf{g}}_t, \eta)$ be as in Algorithm 2. First, note that for any $t \in [T]$ such that $\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} > 2d$, we have $\tilde{\mathbf{g}}_t = \mathbf{0}$. Fix $t \in [T]$ such that $\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} \leq 2d$. By Lemma 2.1 and Assumption 2.2, we have

$$\mathbf{c}_{t+1} \in \mathcal{K} \subseteq \mathbb{B}(R).$$

For the rest of this proof, we fix $\mathbf{u} \in \mathcal{K}$. By definition of \mathbf{z}_{t+1} in Algorithm 2, we have

$$\mathbf{z}_{t+1} - \mathbf{u} = \mathbf{u}_t - \mathbf{u} - \Sigma_{t+1}^{-1} \tilde{\mathbf{g}}_t.$$

Multiplying both sides by $\Sigma_{t+1}^{1/2}$, we get

$$\Sigma_{t+1}^{1/2}(\mathbf{z}_{t+1} - \mathbf{u}) = \Sigma_{t+1}^{1/2}(\mathbf{u}_t - \mathbf{u}) - \Sigma_{t+1}^{-1/2} \tilde{\mathbf{g}}_t.$$

Taking the norm of both sides and squaring leads to

$$\|\mathbf{z}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}}^2 = \|\mathbf{u}_t - \mathbf{u}\|_{\Sigma_{t+1}}^2 + \|\tilde{\mathbf{g}}_t\|_{\Sigma_{t+1}^{-1}}^2 - 2\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle. \quad (42)$$

Using that $\|\mathbf{u}_t - \mathbf{u}\|_{\Sigma_{t+1}}^2 = \|\mathbf{u}_t - \mathbf{u}\|_{\Sigma_t}^2 + \eta \langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle^2$ (since $\Sigma_{t+1} = \Sigma_t + \eta \tilde{\mathbf{g}}_t \tilde{\mathbf{g}}_t^\top$) and rearranging (42) gives

$$\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle - \frac{\eta}{2} \langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle^2 \leq \frac{1}{2} \|\mathbf{u}_t - \mathbf{u}\|_{\Sigma_t}^2 - \frac{1}{2} \|\mathbf{z}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}}^2 + \frac{1}{2} \|\tilde{\mathbf{g}}_t\|_{\Sigma_{t+1}^{-1}}^2. \quad (43)$$

We will apply Lemma C.1 (guarantee of PoE) to bound the term $\frac{1}{2} \|\mathbf{z}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}}^2$ on left-hand side of (43). Then, summing the resulting inequality over $t = 1, \dots, T$ will give us the desired result.

Invoking the guarantee of PoE. Note that at iteration t , Algorithm 2 calls PoE with input $(\mathbf{z}, \mathbf{c}, Q, H, \lambda_-, \lambda_+, \delta) = (\mathbf{z}_{t+1}, \mathbf{c}_{t+1}, \beta R^2 \Sigma_{t+1}^{-1}, H_{t+1}, \lambda_{\min}, \lambda_{\max}, \varepsilon)$, where $\varepsilon = \frac{1}{\kappa^{18} T^2}$, $\lambda_{\min} = \frac{\varepsilon}{24\kappa^2}$, $\lambda_{\max} = \frac{40d^2\kappa}{\varepsilon}$, and $\beta = \eta G^2$ (G is as in Assumption 2.1). To invoke Lemma C.1, we need to check that the conditions on λ_- and λ_+ are satisfied. By Lemma 4.1, we have that

$$\sigma_{\min}(H_{t+1}) \geq r^2 \quad \text{and} \quad \sigma_{\max}(H_{t+1}) \leq \kappa^{16} R^2. \quad (44)$$

On the other hand, since $\Sigma_{t+1} = \beta I + \eta \sum_{\tau=1}^t \tilde{\mathbf{g}}_\tau \tilde{\mathbf{g}}_\tau^\top$ and $\|\tilde{\mathbf{g}}_t\| \leq G(1 + 4\kappa d)$ (by Lemma 4.2), we have

$$\sigma_{\min}(\Sigma_{t+1}) \geq \beta \quad \text{and} \quad \sigma_{\max}(\Sigma_{t+1}) \leq \beta + \eta T \max_{\tau \in [t+1]} \|\tilde{\mathbf{g}}_\tau\|^2 \leq \beta + \eta T G^2 (1 + 4\kappa d)^2. \quad (45)$$

Therefore, since $(Q, H, \delta) = (\beta R^2 \Sigma_{t+1}^{-1}, H_{t+1}, \varepsilon)$, we have

$$\begin{aligned} \delta \cdot \frac{\sigma_{\min}(H)^2}{24R^2 \sigma_{\max}(Q)} &= \varepsilon \cdot \frac{\sigma_{\min}(H_{t+1})^2}{24R^2 \sigma_{\max}(\beta R^2 \Sigma_{t+1}^{-1})}, \\ &\geq \varepsilon \cdot \frac{r^4 \sigma_{\min}(\Sigma_{t+1})}{24\beta R^4}, \quad (\text{by (44)}) \\ &\geq \varepsilon \cdot \frac{r^4 \beta}{24\beta R^4}, \quad (\text{by (45)}) \\ &\geq \frac{\varepsilon}{24\kappa^2} = \lambda_{\min} = \lambda_-. \end{aligned}$$

Thus, the condition on λ_- in Lemma C.1 is satisfied. Now, we show that the condition on λ_+ is satisfied. We have

$$\begin{aligned} \frac{4\sigma_{\max}(H) \cdot R}{\sigma_{\min}(H)^{1/2} \cdot \sigma_{\min}(Q)} &= \frac{4\sigma_{\max}(H_{t+1}) \cdot R}{\sigma_{\min}(H_{t+1})^{1/2} \cdot \sigma_{\min}(\beta R^2 \Sigma_{t+1}^{-1})}, \\ &\leq \frac{4\kappa^{16} R^3 \sigma_{\max}(\Sigma_{t+1})}{\beta R^2}, \quad (\text{by (44)}) \\ &\leq 4\kappa^{17} (1 + \eta \beta^{-1} T G^2 (1 + 4\kappa d)^2), \quad (\text{by (45)}) \\ &= 4\kappa^{17} (1 + T (1 + 4\kappa d)^2), \quad (\text{since } \beta = \eta G^2) \\ &\leq 40T d^2 \kappa^{19} \leq \lambda_{\max} = \lambda_+. \end{aligned}$$

Thus, the condition on λ_+ is also satisfied. Therefore, we can apply Lemma C.1 in this proof. In particular, Lemma C.1 implies that

$$\mathbf{u}_{t+1} \in \mathcal{E}_{t+1}, \quad (46)$$

which we will use in the sequel.

Computational cost. The cost of computing the new iterate \mathbf{u}_{t+1} given \mathbf{g}_t is dominated by the cost of the PoE call on Line 15 of Algorithm 2. Thus, by Lemma C.1, this cost is at most $O(d^\omega \cdot \log(\frac{\lambda_{\max} L}{\lambda_{\min} \varepsilon}))$, where

$$L := 4(B + R) \cdot \frac{\sigma_{\max}(H) \cdot \sigma_{\max}(Q)^2}{\sigma_{\min}(H)^3 \cdot \sigma_{\min}(Q)} \cdot R; \quad B := R \cdot \frac{\frac{\sigma_{\max}(H)}{\sigma_{\min}(Q)} + \lambda_{\max}}{\frac{\sigma_{\min}(H)}{\sigma_{\max}(Q)}},$$

and $(Q, H, \delta) = (\beta R^2 \Sigma_{t+1}^{-1}, H_{t+1}, \varepsilon)$. Using (44) and (45), we get that $\frac{\lambda_{\max} L}{\lambda_{\min} \varepsilon} \leq \text{poly}(T, R/r)$, and so the cost of computing \mathbf{u}_{t+1} given $\tilde{\mathbf{g}}_t$ is at most $O(d^\omega \cdot \log(TR/r))$.

Bounding the instantaneous ONS regret. We now bound the instantaneous regret $\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle$ of the inner ONS algorithm within Algorithm 2. We consider cases based on the value of

$$\|\mathbf{z}_{t+1} - \mathbf{c}_{t+1}\|_{H_{t+1}^{-1}}^2.$$

Case where $\|\mathbf{z}_{t+1} - \mathbf{c}_{t+1}\|_{H_{t+1}^{-1}}^2 \leq 1 + \varepsilon$. If $(\mathbf{z}_{t+1} - \mathbf{c}_{t+1})^\top H_{t+1}^{-1} (\mathbf{z}_{t+1} - \mathbf{c}_{t+1}) \leq 1 + \varepsilon$, then PoE returns $\mathbf{u}_{t+1} = (\mathbf{z}_{t+1} - \mathbf{c}_{t+1}) / (1 + \varepsilon) + \mathbf{c}_{t+1} \in \mathcal{E}_{t+1}$. In this case, by the triangle inequality, we have

$$\begin{aligned} \|\mathbf{z}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} &= \|(1 + \varepsilon)(\mathbf{u}_{t+1} - \mathbf{c}_{t+1}) + \mathbf{c}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}}, \\ &\geq \|\mathbf{u}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} - \varepsilon \cdot \|\mathbf{u}_{t+1} - \mathbf{c}_{t+1}\|_{\Sigma_{t+1}}, \\ &\geq \|\mathbf{u}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} - \varepsilon \cdot \sigma_{\max}(\Sigma_{t+1})^{1/2} \cdot \|\mathbf{u}_{t+1} - \mathbf{c}_{t+1}\|, \\ &\geq \|\mathbf{u}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} - \varepsilon \cdot \sigma_{\max}(\Sigma_{t+1})^{1/2} \cdot \sigma_{\max}(H_{t+1})^{1/2} \cdot \|\mathbf{u}_{t+1} - \mathbf{c}_{t+1}\|_{H_{t+1}^{-1}}, \\ &\geq \|\mathbf{u}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} - \varepsilon \cdot \sigma_{\max}(\Sigma_{t+1})^{1/2} \cdot \sigma_{\max}(H_{t+1})^{1/2}, \quad (\text{by (46)}), \\ &\geq \|\mathbf{u}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} - \varepsilon \kappa^8 R \cdot \left(\sqrt{\beta} + \sqrt{\eta T G} (1 + 4\kappa d) \right), \end{aligned} \quad (47)$$

where the last inequality follows by (44) and (45). Now, by (46), we have $\mathbf{u}_{t+1} \in \mathcal{E}_{t+1}$. On the other hand, by Lemma 4.1, $\mathcal{K} \subseteq \mathcal{E}_{t+1}$ and so $\mathbf{u} \in \mathcal{E}_{t+1}$. Therefore, $\|\mathbf{u}_{t+1} - \mathbf{u}\|_{H_{t+1}^{-1}} \leq 2$, and so

$$\begin{aligned} \|\mathbf{u}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} &\leq \sigma_{\max}(\Sigma_{t+1})^{1/2} \cdot \sigma_{\max}(H_{t+1}^{1/2}) \cdot \|\mathbf{u}_{t+1} - \mathbf{u}\|_{H_{t+1}^{-1}}, \\ &\leq 2\kappa^8 R \cdot (\sqrt{\beta} + \sqrt{\eta T G} (1 + 4\kappa d)), \end{aligned} \quad (48)$$

where the last inequality follows by (45) and (44). Squaring (47) and using (48), we get

$$\begin{aligned} &[\text{case: } (\mathbf{z}_{t+1} - \mathbf{c}_{t+1})^\top H_{t+1}^{-1} (\mathbf{z}_{t+1} - \mathbf{c}_{t+1}) \leq 1 + \varepsilon] \\ \|\mathbf{z}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}}^2 &\geq \|\mathbf{u}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}}^2 - 8\varepsilon \kappa^{16} R^2 \cdot (\beta + \eta T G^2 (1 + 4\kappa d)^2). \end{aligned} \quad (49)$$

Case where $\|\mathbf{z}_{t+1} - \mathbf{c}_{t+1}\|_{H_{t+1}^{-1}}^2 > 1 + \varepsilon$. Now, suppose that $(\mathbf{z}_{t+1} - \mathbf{c}_{t+1})^\top H_{t+1}^{-1} (\mathbf{z}_{t+1} - \mathbf{c}_{t+1}) > 1 + \varepsilon$. In this case, by Lemma C.1, there exists $\nu \in [0, \varepsilon]$ such that

$$\mathbf{u}_{t+1} \in \arg \min_{\mathbf{x} \in \mathcal{E}_{t+1}(\nu)} (\mathbf{x} - \mathbf{z}_{t+1})^\top \Sigma_{t+1} (\mathbf{x} - \mathbf{z}_{t+1}), \quad (50)$$

where $\mathcal{E}_{t+1}(\nu) := \{\mathbf{x} \in \mathbb{R}^d \mid (\mathbf{x} - \mathbf{c}_{t+1})^\top H_{t+1}^{-1} (\mathbf{x} - \mathbf{c}_{t+1}) \leq 1 - \nu\} \subseteq \mathcal{E}_{t+1}$. Fix such a ν . By Lemma 2.1, we have that $\mathcal{K} \subseteq \mathcal{E}_{t+1}$, and so $\mathbf{u} \in \mathcal{E}_{t+1}$. This in turn implies that

$$\mathbf{u}_\nu := \frac{1}{1 + 2\nu} (\mathbf{u} - \mathbf{c}_{t+1}) + \mathbf{c}_{t+1} \in \mathcal{E}_{t+1}(\nu),$$

where we used that $\frac{1}{1+x} \leq 1 - \frac{x}{2}$, for $x \in [0, 1/2]$. Thus, by (50) and Lemma G.2, we have

$$\|\mathbf{u}_{t+1} - \mathbf{u}_\nu\|_{\Sigma_{t+1}} \leq \|\mathbf{z}_{t+1} - \mathbf{u}_\nu\|_{\Sigma_{t+1}}.$$

On the other hand, by the expression of \mathbf{u}_ν and the triangle inequality, we have that

$$\begin{aligned}
\|\mathbf{z}_{t+1} - \mathbf{u}_\nu\|_{\Sigma_{t+1}} &= \left\| \mathbf{z}_{t+1} - \frac{1}{1+2\nu}(\mathbf{u} - \mathbf{c}_{t+1}) + \mathbf{c}_{t+1} \right\|_{\Sigma_{t+1}}, \\
&\leq \|\mathbf{z}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} + \frac{2\nu}{1+2\nu} \|\mathbf{u} - \mathbf{c}_{t+1}\|_{\Sigma_{t+1}}, \\
&\leq \|\mathbf{z}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} + 2\varepsilon \cdot \sigma_{\max}(\Sigma_{t+1})^{1/2} \cdot \|\mathbf{u} - \mathbf{c}_{t+1}\|, \quad (\nu \in [0, \varepsilon]) \\
&\leq \|\mathbf{z}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} + 2\varepsilon \cdot \sigma_{\max}(\Sigma_{t+1})^{1/2} \cdot \sigma_{\max}(H_{t+1})^{1/2} \cdot \|\mathbf{u} - \mathbf{c}_{t+1}\|_{H_{t+1}^{-1}}, \\
&\leq \|\mathbf{z}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} + 2\varepsilon \cdot \sigma_{\max}(\Sigma_{t+1})^{1/2} \cdot \sigma_{\max}(H_{t+1})^{1/2}, \quad (\text{since } \mathbf{u} \in \mathcal{E}_{t+1}) \\
&\leq \|\mathbf{z}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} + 2\varepsilon \kappa^8 R \cdot \left(\sqrt{2\beta} + \sqrt{\eta T G(1+4\kappa d)} \right), \tag{51}
\end{aligned}$$

where the last inequality follows by (45) and (44). Similarly, we have that

$$\begin{aligned}
\|\mathbf{u}_{t+1} - \mathbf{u}_\nu\|_{\Sigma_{t+1}} &= \left\| \mathbf{u}_{t+1} - \frac{1}{1+2\nu}(\mathbf{u} - \mathbf{c}_{t+1}) + \mathbf{c}_{t+1} \right\|_{\Sigma_{t+1}}, \\
&\geq \|\mathbf{u}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} - \frac{2\nu}{1+2\nu} \|\mathbf{u} - \mathbf{c}_{t+1}\|_{\Sigma_{t+1}}, \\
&\geq \|\mathbf{u}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} - 2\varepsilon \kappa^8 R \cdot \left(\sqrt{2\beta} + \sqrt{\eta T G(1+4\kappa d)} \right). \tag{52}
\end{aligned}$$

Thus, combining (51) and (52), we get

$$\|\mathbf{u}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}} - 2\varepsilon \kappa^8 R \cdot \left(\sqrt{2\beta} + \sqrt{\eta T G(1+4\kappa d)} \right) \leq \|\mathbf{z}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}}. \tag{53}$$

Taking the square in (53) and using (48), we get

$$\begin{aligned}
&[\text{case: } (\mathbf{z}_{t+1} - \mathbf{c}_{t+1})^\top H_{t+1}^{-1} (\mathbf{z}_{t+1} - \mathbf{c}_{t+1}) > 1 + \varepsilon] \\
&\|\mathbf{u}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}}^2 - 24\varepsilon \kappa^{16} R^2 (2\beta + \eta T G^2(1+4\kappa d)^2) \leq \|\mathbf{z}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}}^2. \tag{54}
\end{aligned}$$

Plugging (49) and (54) into (43) yields

$$\begin{aligned}
&\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle - \frac{\eta}{2} \langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle^2 \\
&\leq \frac{1}{2} \|\mathbf{u}_t - \mathbf{u}\|_{\Sigma_t}^2 - \frac{1}{2} \|\mathbf{u}_{t+1} - \mathbf{u}\|_{\Sigma_{t+1}}^2 + \frac{1}{2} \|\tilde{\mathbf{g}}_t\|_{\Sigma_{t+1}^{-1}}^2 + 24\varepsilon \kappa^{16} R^2 \cdot (2\beta + \eta T G^2(1+4\kappa d)^2). \tag{55}
\end{aligned}$$

So far, we have considered rounds $t \in [T]$ satisfying $\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} \leq 2d$. As remarked at the beginning of this proof, when $\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} > 2d$, we have $\tilde{\mathbf{g}}_t = \mathbf{0}$, and so (55) remains true since $\Sigma_{t+1} = \Sigma_t$ and $\mathbf{u}_{t+1} = \mathbf{u}_t$.

Bounding the ONS regret. We now bound the regret of ONS (not just the instantaneous regret).

Summing (55) over $t \in [T]$ and telescoping the terms $(\|\mathbf{u}_t - \mathbf{u}\|_{\Sigma_t}^2)$, we get

$$\begin{aligned}
&\sum_{t=1}^T \left(\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle - \frac{\eta}{2} \langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle^2 \right) \\
&\leq \frac{1}{2} \|\mathbf{u}_1 - \mathbf{u}\|_{\Sigma_1}^2 + \frac{1}{2} \sum_{t=1}^T \|\tilde{\mathbf{g}}_t\|_{\Sigma_{t+1}^{-1}}^2 + 24T\varepsilon \kappa^{16} R^2 \cdot (2\beta + T\eta G^2(1+4\kappa d)^2), \\
&\leq 2\beta R^2 + \frac{1}{2} \sum_{t=1}^T \|\tilde{\mathbf{g}}_t\|_{\Sigma_{t+1}^{-1}}^2 + 24T\varepsilon \kappa^{16} R^2 \cdot (2\beta + \eta T G^2(1+4\kappa d)^2). \tag{56}
\end{aligned}$$

On the other hand, by [11, Lemma 11] and Lemma 4.2, we have that

$$\sum_{t=1}^T \|\tilde{\mathbf{g}}_t\|_{\Sigma_{t+1}^{-1}}^2 \leq \frac{d \log(\eta G^2(1+4\kappa d)^2 T / \beta + 1)}{\eta}.$$

Plugging this into (56) and using that $\beta = \eta G^2$ and $\varepsilon = \frac{1}{T^2 \kappa^{18}}$, we get

$$\begin{aligned} & \sum_{t=1}^T \left(\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle - \frac{\eta}{2} \langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle^2 \right) \\ & \leq 2\beta R^2 + \frac{d \log(\eta G^2 (1 + 4\kappa d)^2 T / \beta + 1)}{2\eta} + 24T \varepsilon \kappa^{16} R^2 \cdot (2\beta + \eta T G^2 (1 + 4\kappa d)^2), \\ & \leq 2\eta G^2 R^2 + \frac{d \log(1 + 9\kappa^2 d^2 T^2)}{2\eta} + 240\eta R^2 G^2 d^2. \end{aligned}$$

This completes the proof. \square

E OCO Analysis

E.1 Algorithm Invariants (Proofs of Lemma 4.1 and Lemma 4.2)

Proof of Lemma 4.1. We will show the claim via induction over $t = 1, \dots, T$. The base case holds trivially because $H_1 = R^2 I$ and $\mathcal{K} \subseteq \mathbb{B}(R) = \mathcal{E}_1$, where the set inclusion follows by Assumption 2.2. Now, suppose that the claim holds for $t \in [T-1]$ and we show that it holds for $t+1$. Note that $(\mathbf{c}_{t+1}, H_{t+1}) \neq (\mathbf{c}_t, H_t)$ only if $\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} > 2d$. Suppose that $\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} > 2d$. By Lemma A.1, we have that

$$\forall \mathbf{u} \in \mathcal{K}, \quad \langle \mathbf{u} - \mathbf{c}_t, \mathbf{s}_t \rangle \leq 1.$$

Therefore, we have

$$\forall \mathbf{u} \in \mathcal{K}, \quad \left\langle H_t^{-1/2}(\mathbf{u} - \mathbf{c}_t), \frac{H_t^{1/2} \mathbf{s}_t}{\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t}} \right\rangle \leq \frac{1}{\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t}} \leq \frac{1}{2d}.$$

This implies that for the unit-norm vector $\mathbf{v}_t = \frac{H_t^{1/2} \mathbf{s}_t}{\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t}}$, we have that

$$H_t^{-1/2}(\mathcal{K} - \mathbf{c}_t) \subseteq \{\mathbf{u} \in \mathbb{R}^d \mid \mathbf{v}_t^\top \mathbf{u} \leq \frac{1}{2d}\}. \quad (57)$$

And, by the induction hypothesis, we also have that $H_t^{-1/2}(\mathcal{K} - \mathbf{c}_t) \subseteq \mathbb{B}(1)$. Combining this with (57) implies that

$$H_t^{-1/2}(\mathcal{K} - \mathbf{c}_t) \subseteq \{\mathbf{u} \in \mathbb{B}(1) \mid \mathbf{v}_t^\top \mathbf{u} \leq \frac{1}{2d}\}. \quad (58)$$

Now, by Lemma B.1, we have that

$$\{\mathbf{u} \in \mathbb{B}(1) \mid \mathbf{v}_t^\top \mathbf{u} \leq \frac{1}{2d}\} \subseteq \left\{ \mathbf{u} \in \mathbb{R}^d \mid \left(\mathbf{u} + \frac{1}{2(d+1)} \mathbf{v}_t \right)^\top \tilde{H}_{t+1}^{-1} \left(\mathbf{u} + \frac{1}{2(d+1)} \mathbf{v}_t \right) \leq 1 \right\}, \quad (59)$$

where $\tilde{H}_{t+1} := \frac{4d^2-1}{4d^2-4} \cdot \left(I_d - \frac{2d}{2d^2+d-1} \mathbf{v}_t \mathbf{v}_t^\top \right)$. Combining (59) with (58) implies that

$$H^{-1/2}(\mathcal{K} - \mathbf{c}_t) \subseteq \left\{ \mathbf{u} \in \mathbb{R}^d \mid \left(\mathbf{u} + \frac{1}{2(d+1)} \mathbf{v}_t \right)^\top \tilde{H}_{t+1}^{-1} \left(\mathbf{u} + \frac{1}{2(d+1)} \mathbf{v}_t \right) \leq 1 \right\}.$$

This means that

$$\forall \mathbf{u} \in \mathcal{K}, \quad 1 \geq \left(H^{-1/2} \mathbf{u} - H^{-1/2} \mathbf{c}_t + \frac{1}{2(d+1)} \mathbf{v}_t \right)^\top \tilde{H}_{t+1}^{-1} \left(H^{-1/2} \mathbf{u} - H^{-1/2} \mathbf{c}_t + \frac{1}{2(d+1)} \mathbf{v}_t \right),$$

and by using the definitions of \tilde{H}_{t+1} , H_{t+1} , \mathbf{v}_t , and \mathbf{c}_{t+1} , we have

$$\begin{aligned} & = \left(\mathbf{u} - \mathbf{c}_t + \frac{1}{2(d+1)} H^{1/2} \mathbf{v}_t \right)^\top H_{t+1}^{-1} \left(\mathbf{u} - \mathbf{c}_t + \frac{1}{2(d+1)} H^{1/2} \mathbf{v}_t \right), \\ & = (\mathbf{u} - \mathbf{c}_{t+1})^\top H_{t+1}^{-1} (\mathbf{u} - \mathbf{c}_{t+1}). \end{aligned} \quad (60)$$

Thus, (60) implies that $\mathcal{K} \subseteq \mathcal{E}_{t+1}$.

Showing Item 2. This item follows from Lemma 2.1.

Showing Item 3. If $\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} \leq 2d$, then $(\mathbf{c}_{t+1}, H_{t+1}) = (\mathbf{c}_t, H_t)$ and Item 3 follows immediately by the induction hypothesis. Now, assume that $\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} > 2d$. In this case, we have $H_{t+1} = \frac{4d^2-1}{4d^2-4} \cdot H_t^{1/2} \left(I - \frac{2d}{2d^2+d-1} \frac{H_t^{1/2} \mathbf{s}_t \mathbf{s}_t^\top H_t^{1/2}}{\mathbf{s}_t^\top H_t \mathbf{s}_t} \right) H_t^{1/2}$. Thus,

$$H_t^{-1/2} \mathcal{E}_{t+1} = \{ \mathbf{u} \in \mathbb{R}^d \mid (\mathbf{u} - H_t^{-1/2} \mathbf{c}_{t+1})^\top \tilde{H}_{t+1}^{-1} (\mathbf{u} - H_t^{-1/2} \mathbf{c}_{t+1}) \leq 1 \}$$

where

$$\tilde{H}_{t+1} := \frac{4d^2-1}{4d^2-4} \cdot \left(I - \frac{2d}{2d^2+d-1} \frac{H_t^{1/2} \mathbf{s}_t \mathbf{s}_t^\top H_t^{1/2}}{\mathbf{s}_t^\top H_t \mathbf{s}_t} \right),$$

and so by Lemma B.1, we have that

$$\frac{\text{vol}(H_t^{-1/2} \mathcal{E}_{t+1})}{\text{vol}(\mathbb{B}(1))} \leq e^{-\frac{1}{8d}}. \quad (61)$$

On the other hand, we have

$$\text{vol}(H_t^{-1/2} \mathcal{E}_{t+1}) = |\det(H_t^{-1/2})| \cdot \text{vol}(\mathcal{E}_{t+1}) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} \cdot \frac{\text{vol}(\mathcal{E}_{t+1})}{\text{vol}(\mathcal{E}_t)},$$

where the last equality follows by the fact that $\text{vol}(\mathcal{E}_t) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} \frac{1}{\sqrt{\det(H_t^{-1})}}$ and $|\det(H_t^{-1/2})| = \sqrt{\det(H_t^{-1})}$ because H_t is positive definite. Now, using that $\text{vol}(\mathbb{B}(1)) = \pi^{d/2}/\Gamma(d/2+1)$, we get that

$$\frac{\text{vol}(H_t^{-1/2} \mathcal{E}_{t+1})}{\text{vol}(\mathbb{B}(1))} = \frac{\text{vol}(\mathcal{E}_{t+1})}{\text{vol}(\mathcal{E}_t)}.$$

Combining this with (61), we get

$$\frac{\text{vol}(\mathcal{E}_{t+1})}{\text{vol}(\mathcal{E}_t)} \leq e^{-\frac{1}{8d}},$$

as desired. This proves Item 3.

Showing Item 4. By Item 1 and Assumption 2.2, we have that $\mathbb{B}(\mathbf{c}_0, r) \subseteq \mathcal{K} \subseteq \mathcal{E}_{t+1}$. Therefore, $\text{vol}(\mathcal{E}_{t+1}) \geq r^d$. On the other hand, by Item 3, we have $\text{vol}(\mathcal{E}_{t+1}) \leq e^{-\frac{N_t}{8d}} \cdot R^d$, and so $N_t \leq 8d^2 \log(R/r)$; otherwise, we would contradict $\text{vol}(\mathcal{E}_{t+1}) \geq r^d$.

Showing Item 5. We now prove Item 5 in the lemma statement. By Assumption 2.2, there exists $\mathbf{c}_0 \in \mathbb{B}(R)$ such that $\mathbb{B}(\mathbf{c}_0, r) \subseteq \mathcal{K}$, and by Item 1 we have $\mathbb{B}(\mathbf{c}_0, r) \subseteq \mathcal{E}_{t+1}$. Thus, by Lemma G.3, we have

$$\mathbb{B}(\mathbf{c}_{t+1}, r) \subseteq \mathcal{E}_{t+1}.$$

Let \mathbf{z}_{t+1} be the unit-norm eigenvector corresponding to the largest eigenvalue of H_{t+1}^{-1} . Since $r\mathbf{z}_{t+1} + \mathbf{c}_{t+1} \in \mathbb{B}(\mathbf{c}_{t+1}, r) \subseteq \mathcal{E}_{t+1}$, we have that

$$\begin{aligned} 1 &\geq r^2 \mathbf{z}_{t+1}^\top H_{t+1}^{-1} \mathbf{z}_{t+1}, \\ &= r^2 \|H_{t+1}^{-1}\|_{\text{op}}, \\ &= \frac{r^2}{\sigma_{\min}(H_{t+1})}. \end{aligned}$$

Rearranging implies that $\sigma_{\min}(H_{t+1}) \geq r^2$. It remains to bound $\sigma_{\max}(H_{t+1})$. By Lemma G.4, we have that

$$\sigma_{\max}(H_{t+1}) \leq \left(1 + \frac{2}{d^2} \right)^{\mathbb{I}\{\|H_t^{1/2} \mathbf{s}_t\| > 2d\}} \cdot \sigma_{\max}(H_t).$$

Thus, by the induction hypothesis, we have

$$\sigma_{\max}(H_{t+1}) \leq \left(1 + \frac{2}{d^2}\right)^{N_t} R^2,$$

and so by Item 4, we have

$$\begin{aligned} &\leq \left(1 + \frac{2}{d^2}\right)^{8d^2 \log(R/r)} R^2, \\ &\leq e^{16 \log(R/r)} R^2, \quad (\text{see below}) \\ &\leq \kappa^{16} R^2, \end{aligned} \tag{62}$$

where (62) follows from $(1 + x/n)^n \leq e^x$ for all $n \geq 1$ and $|x| \leq n$. \square

Proof of Lemma 4.2. Fix $t \in [T]$, and let S_t , \mathbf{s}_t , \mathbf{u}_t , $\tilde{\mathbf{g}}_t$, and \mathbf{w}_t be as in Algorithm 2. When the condition in Line 4 of Algorithm 2 is satisfied (i.e., when $\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} > 2d$), we simply have $\tilde{\mathbf{g}}_t = \mathbf{0}$ and $\mathbf{w}_t = \mathbf{c} \in \mathcal{K}$, in which case all the items hold. For the rest of the proof, we assume that

$$\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} \leq 2d. \tag{63}$$

By Lemma 2.1, we have that $\mathbf{c}_t \in \mathcal{K}$. Using this, the triangle inequality, Assumption 2.2, and Hölder's inequality, we have

$$\begin{aligned} \|\tilde{\mathbf{g}}_t\| &\leq \|\mathbf{g}_t\| + |\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{c}_t \rangle| \cdot \|\mathbf{s}_t\|, \\ &\leq G + 2GR \|H_t^{-1/2}\|_{\text{op}} \cdot \sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t}, \\ &= G + 2GR \frac{\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t}}{\sigma_{\min}(H_t)^{1/2}}, \\ &\leq G(1 + 4\kappa d), \end{aligned}$$

where the last inequality follows from (63) and the fact that $\sigma_{\min}(H_t) \geq r^2$ by Lemma 4.1. This shows Item 1.

Proving Item 2. By definition of \mathbf{w}_t , we have $\mathbf{w}_t = \frac{\mathbf{u}_t - \mathbf{c}_t}{1 + S_t} + \mathbf{c}_t$ (recall that we are in the case where $\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} \leq 2d$). Therefore, by the homogeneity of the Gauge function (see Lemma G.5), we have

$$\begin{aligned} \gamma_{\mathcal{K}-\mathbf{c}_t}(\mathbf{w}_t - \mathbf{c}_t) &= \frac{\gamma_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t)}{1 + S_t}, \\ &\leq \frac{1 + S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t)}{1 + S_t}, \quad (\text{since } S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t) = \max(0, \gamma_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t) - 1)) \\ &\leq 1, \end{aligned} \tag{64}$$

where the last inequality follows from $S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t) \leq S_t$ by Lemma 2.1. Eq.(64) implies that $\mathbf{w}_t - \mathbf{c}_t \in \mathcal{K} - \mathbf{c}_t$ by definition of the Gauge function (see Definition 2.2), which is equivalent to $\mathbf{w}_t \in \mathcal{K}$.

Proving Item 3. We now show Item 3. Fix $\mathbf{u} \in \mathcal{K}$. Using the expression of $\tilde{\mathbf{g}}_t$ and the triangle inequality, we have that

$$|\langle \mathbf{u} - \mathbf{u}_t, \tilde{\mathbf{g}}_t \rangle| \leq |\langle \mathbf{u} - \mathbf{u}_t, \mathbf{g}_t \rangle| + |\langle \mathbf{u} - \mathbf{u}_t, \mathbf{s}_t \rangle| \cdot |\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{c}_t \rangle|,$$

and since $\mathbf{c}_t \in \mathcal{K}$ (Lemma 4.1) and $\mathbf{w}_t \in \mathcal{K} \subseteq \mathbb{B}(R)$ by Item 2 and Assumption 2.2, we have

$$\begin{aligned} &\leq 2RG + 2RG \cdot |\langle \mathbf{u} - \mathbf{u}_t, \mathbf{s}_t \rangle|, \\ &\leq 2RG + 2RG \cdot \|H_t^{-1/2}(\mathbf{u} - \mathbf{u}_t)\| \cdot \sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t}, \quad (\text{by Hölder's inequality}) \\ &\leq 2RG + 2RG \cdot \left(\|H_t^{-1/2}(\mathbf{u} - \mathbf{c}_t)\| + \|H_t^{-1/2}(\mathbf{c}_t - \mathbf{u}_t)\| \right) \cdot \sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t}, \\ &\leq 2RG + 4RG \cdot \sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t}, \quad (\text{see below}) \\ &\leq 2RG(1 + 4d), \end{aligned} \tag{65}$$

where (65) follows by the fact that $\mathcal{K} \subseteq \mathcal{E}_t$ (Lemma 4.1) and that $\mathbf{u}_t \in \mathcal{E}_t$; this is because \mathbf{u}_t is the output of \mathcal{A} which is constrained to output a vector in \mathcal{E}_t at round t .

Proving Item 4. We now prove that

$$\forall \mathbf{u} \in \mathcal{K}, \quad \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle \leq \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle + \frac{3GR}{T}. \quad (66)$$

For this, define the surrogate loss function ℓ_t :

$$\forall \mathbf{u} \in \mathbb{R}^d, \quad \ell_t(\mathbf{u}) := \langle \mathbf{g}_t, \mathbf{u} \rangle - \mathbb{I}\{\langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle < 0\} \cdot \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{c}_t \rangle \cdot S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u} - \mathbf{c}_t).$$

Since the pair (S_t, \mathbf{s}_t) is the output of $\text{GaugeDist}(\mathbf{u}_t, \mathcal{K}, H_t, \mathbf{c}_t, \varepsilon)$ with $\varepsilon = \frac{1}{\kappa^8 T^2}$, we have by Lemma 2.1:

$$\forall \mathbf{u} \in \mathbb{R}^d, \quad S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u} - \mathbf{c}_t) \geq S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t) + (\mathbf{u} - \mathbf{u}_t)^\top \mathbf{s}_t - \frac{1}{\kappa^8 T}. \quad (67)$$

Now, since $\mathbf{w}_t = (\mathbf{u}_t - \mathbf{c}_t)/(1 + S_t) + \mathbf{c}_t$ (see Algorithm 2) and $S_t \geq S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t) \geq 0$ (by Lemma 2.1), we have that $-\mathbb{I}\{\langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle < 0\} \cdot \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{c}_t \rangle \geq 0$. And so, using (67) and the definition of ℓ_t , we get

$$\begin{aligned} \forall \mathbf{u} \in \mathbb{R}^d, \quad \ell_t(\mathbf{u}_t) - \ell_t(\mathbf{u}) &\leq \langle \mathbf{g}_t - \mathbb{I}\{\langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle < 0\} \cdot \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{c}_t \rangle \cdot \mathbf{s}_t, \mathbf{u}_t - \mathbf{u} \rangle + |\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{c}_t \rangle| \cdot \frac{1}{T}, \\ &= \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle + |\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{c}_t \rangle| \cdot \frac{1}{T}, \quad (\text{by definition of } \tilde{\mathbf{g}}_t \text{ in Algorithm 2}) \\ &\leq \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle + \frac{2GR}{T}, \end{aligned} \quad (68)$$

where the last inequality uses that $\mathbf{w}_t \in \mathcal{K} \subseteq \mathbb{B}(R)$ (by Item 2 and Assumption 2.2), $\mathbf{c}_t \in \mathcal{K}$ (by Lemma 4.1), and $\|\mathbf{g}_t\| \leq G$ (Assumption 2.1).

It remains to prove that $\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle \leq \ell_t(\mathbf{u}_t) - \ell_t(\mathbf{u})$, for all $\mathbf{u} \in \mathcal{K}$. First, note that we have for all $\mathbf{u} \in \mathcal{K}$, $S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u} - \mathbf{c}_t) = \max(0, \gamma_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u} - \mathbf{c}_t) - 1) = 0$ (by definition of the Gauge function), and so

$$\ell_t(\mathbf{u}) = \langle \mathbf{g}_t, \mathbf{u} \rangle, \quad \forall \mathbf{u} \in \mathcal{K}. \quad (69)$$

We will now compare $\langle \mathbf{g}_t, \mathbf{w}_t \rangle$ to $\ell_t(\mathbf{u}_t)$ by considering cases. Suppose that $S_t = 0$. In this case, we have $\mathbf{w}_t = \mathbf{u}_t$ and so $\langle \mathbf{g}_t, \mathbf{w}_t \rangle = \langle \mathbf{g}_t, \mathbf{u}_t \rangle = \ell_t(\mathbf{u}_t)$. Now suppose that $S_t > 0$ and $\langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle \geq 0$. In this case, since $\mathbf{w}_t = \frac{\mathbf{u}_t - \mathbf{c}_t}{1 + S_t} + \mathbf{c}_t$, we have

$$\begin{aligned} \langle \mathbf{g}_t, \mathbf{w}_t \rangle &= \frac{1}{1 + S_t} \cdot \langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle + \langle \mathbf{g}_t, \mathbf{c}_t \rangle, \\ &\leq \langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle + \langle \mathbf{g}_t, \mathbf{c}_t \rangle = \langle \mathbf{g}_t, \mathbf{u}_t \rangle = \ell_t(\mathbf{u}_t). \quad [\text{case where } \langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle \geq 0] \end{aligned} \quad (70)$$

Now suppose that $S_t > 0$ and $\langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle < 0$. Again, using that $\mathbf{w}_t = \frac{\mathbf{u}_t - \mathbf{c}_t}{1 + S_t} + \mathbf{c}_t$, we have

$$\begin{aligned} &\langle \mathbf{g}_t, \mathbf{w}_t \rangle + \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{c}_t \rangle \cdot S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t) \\ &= \langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle \cdot \frac{1 + S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t)}{1 + S_t} + \langle \mathbf{g}_t, \mathbf{c}_t \rangle, \end{aligned}$$

and so since $\langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle < 0$ and $S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t) \geq S_t - \frac{1}{\kappa^{18} T^2}$ (by Lemma 2.1)

$$\begin{aligned} &\leq \langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle \cdot \frac{1 + S_t - \frac{1}{\kappa^{18} T^2}}{1 + S_t} + \langle \mathbf{g}_t, \mathbf{c}_t \rangle, \\ &\leq \langle \mathbf{g}_t, \mathbf{u}_t \rangle + |\langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle| \cdot \frac{1}{\kappa^{18} T^2}, \\ &\leq \langle \mathbf{g}_t, \mathbf{u}_t \rangle + \|H^{1/2} \mathbf{g}_t\| \cdot \|H^{-1/2}(\mathbf{u}_t - \mathbf{c}_t)\| \cdot \frac{1}{\kappa^{18} T^2}, \quad (\text{H\"older's inequality}) \\ &\leq \langle \mathbf{g}_t, \mathbf{u}_t \rangle + \|H^{1/2} \mathbf{g}_t\| \cdot \frac{1}{\kappa^{18} T^2}, \quad (\mathbf{u}_t \in \mathcal{E}_t) \\ &\leq \langle \mathbf{g}_t, \mathbf{u}_t \rangle + \frac{GR}{T}, \end{aligned} \quad (71)$$

where the last inequality follows from the fact that $\sigma_{\max}(H_t) \leq \kappa^{16} R^2$ (by Lemma 4.1) and $\|\mathbf{g}_t\| \leq G$ (Assumption 2.1). Rearranging (71), we get

$$\langle \mathbf{g}_t, \mathbf{w}_t \rangle - \frac{GR}{T} \leq \langle \mathbf{g}_t, \mathbf{u}_t \rangle - \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{c}_t \rangle \cdot S_{\mathcal{K}-\mathbf{c}_t}(\mathbf{u}_t - \mathbf{c}_t) = \ell_t(\mathbf{u}_t). \quad [\text{case where } \langle \mathbf{g}_t, \mathbf{u}_t - \mathbf{c}_t \rangle < 0] \quad (72)$$

By combining (68), (69), (70), and (72), we obtain

$$\forall \mathbf{u} \in \mathcal{K}, \quad \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle - \frac{GR}{T} \leq \ell_t(\mathbf{u}_t) - \ell_t(\mathbf{u}) \leq \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle + \frac{2GR}{T},$$

which shows the inequality in (66). \square

E.2 OCO Regret (Proofs of Theorem 4.1)

Proof. Fix $\mathbf{u} \in \mathcal{K}$. By Lemma 4.2, the sequence of vectors $(\tilde{\mathbf{g}}_t)$ satisfies $(\tilde{\mathbf{g}}_t) \subset \mathbb{B}(\tilde{\mathbf{G}})$ with $\tilde{\mathbf{G}} = G \cdot (1 + 4\kappa d)$. Thus, by invoking Lemma 4.3, we get

$$(\mathbf{u}_t) \subset \mathcal{E}_{t+1} := \{\mathbf{x} \in \mathbb{R}^d \mid (\mathbf{x} - \mathbf{c}_{t+1})^\top H_{t+1}^{-1}(\mathbf{x} - \mathbf{c}_{t+1}) \leq 1\}$$

and

$$\begin{aligned} \sum_{t=1}^T \left(\langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle - \frac{\eta}{2} \langle \mathbf{u}_t - \mathbf{u}, \tilde{\mathbf{g}}_t \rangle^2 \right) &\leq 2\eta G^2 R^2 + 240\eta R^2 G^2 d^2 + \frac{d \log(1 + 9\kappa^2 d^2 T^2)}{2\eta}, \\ &\leq 24RGd + \frac{d \log(1 + 9\kappa^2 d^2 T^2)}{2\eta}, \end{aligned} \quad (73)$$

where in the last step we used that $\eta \leq \frac{1}{10dGR}$. We now prove that

$$\begin{aligned} \sum_{t=1}^T \left(\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle - \frac{\eta}{2} \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle^2 \right) &\leq \sum_{t=1}^T \left(\langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle - \frac{\eta}{2} \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle^2 \right) \\ &\quad + 10GR + 16RGd^2 \log(R/r), \end{aligned}$$

which together with (73) would complete the proof. Using Lemma 4.1, we have

$$\forall t \in [T], \quad |\langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle| \leq 2RG(1 + 4d). \quad (74)$$

Combining this with the facts that:

- $\mathbb{I}\{\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} \leq 2d\} \cdot \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle \leq \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle + \frac{3GR}{T}$, for all $t \in [T]$ (by Lemma 4.2);
- $x \rightarrow x - \frac{\eta}{2} x^2$ in non-decreasing for all $x \leq \frac{1}{\eta}$ (we instantiate this with $x = \mathbb{I}\{\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} \leq 2d\} \cdot \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle$ and $x = \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle + \frac{3GR}{T}$); and
- $\eta \leq \frac{1}{10dGR}$;

we get that for all $t \in [T]$

$$\begin{aligned} &\mathbb{I}\{\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} \leq 2d\} \cdot (\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle - \eta \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle^2) \\ &\leq \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle + \frac{3GR}{T} - \eta \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle^2 - \frac{6\eta GR}{T} \cdot \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle - \frac{9\eta G^2 R^2}{T^2}, \\ &\leq \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle - \eta \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle^2 + \frac{10GR}{T}, \end{aligned}$$

where the last step follows by (74) and $\eta \leq \frac{1}{10dGR}$. Summing this over $t = 1, \dots, T$, we obtain

$$\begin{aligned} &\sum_{t=1}^T \left(\langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle - \frac{\eta}{2} \langle \tilde{\mathbf{g}}_t, \mathbf{u}_t - \mathbf{u} \rangle^2 \right) + 10GR \\ &\geq \sum_{t=1}^T \mathbb{I}\{\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} \leq 2d\} \cdot \left(\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle - \frac{\eta}{2} \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle^2 \right) \\ &\geq \sum_{t=1}^T \left(\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle - \frac{\eta}{2} \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle^2 \right) - 16d^2 GR \log(R/r), \end{aligned} \quad (75)$$

where the last inequality follows by the fact that $\sum_{t=1}^T \mathbb{I}\{\sqrt{\mathbf{s}_t^\top H_t \mathbf{s}_t} \leq 2d\} \leq 8d^2 \log(R/r)$ (by Lemma 4.1); $\mathbf{u}_t, \mathbf{u} \in \mathcal{K} \subseteq \mathbb{B}(R)$; and $\mathbf{g}_t \in \mathbb{B}(G)$, for all $t \in [T]$. Rearranging (75) and combining with (73), we get the desired result.

Computational cost. The per-round computational cost of Algorithm 2 is dominated by the calls of the GaugeDist (Algorithm 1) and PoE (Algorithm 3) subroutines. As established in Lemma 2.1, the cost of a single call to GaugeDist is at most $O(C_{\text{sep}}(\mathcal{K}) \cdot \log(TR/r))$. Similarly, Lemma 4.3 show that the cost of a single call to PoE is bounded by $O(d^\omega \cdot \log(TR/r))$. Consequently, the total per-round cost of Algorithm 2 is $O((d^\omega + C_{\text{sep}}(\mathcal{K})) \cdot \log(TR/r))$. \square

F Rates for Stochastic Convex Optimization

In this section, we use our regret bound from Theorem 4.1 to derive a state-of-the-art convergence rate for projection-free stochastic convex optimization that only depends on the asphercity κ of the set \mathcal{K} logarithmically. We start by stating our assumptions for the stochastic optimization setting.

Assumption F.1. *There is a function $f : \mathcal{K} \rightarrow \mathbb{R}$ and parameters $\sigma \geq 0$ and $G > 0$ such that the loss vector \mathbf{g}_t that the algorithm receives at round $t \geq 1$ is of the form $\mathbf{g}_t = \bar{\mathbf{g}}_t + \boldsymbol{\xi}_t$, where*

- For all $t \geq 1$, $\bar{\mathbf{g}}_t \in \partial f(\mathbf{w}_t)$, where \mathbf{w}_t is the output of the algorithm at round t ;
- $(\boldsymbol{\xi}_t) \subset \mathbb{R}^d$ are i.i.d. noise vectors such that $\mathbb{E}[\boldsymbol{\xi}_t] = \mathbf{0}$ and $\mathbb{E}[\|\boldsymbol{\xi}_t\|^2] \leq \sigma^2$, for all $t \geq 1$; and
- For all $t \geq 1$, $\|\bar{\mathbf{g}}_t\| \leq G$.

The conditions in Assumption F.1 are standard in the stochastic convex optimization literature; see, e.g., [19]. Under Assumption F.1, we now state the guarantee of Algorithm 2 when setting the input parameter η to

$$\eta = \min\left(\frac{1}{10dRG}, \sqrt{\frac{d \log(1 + d^2 T^2 \kappa^2)}{16R^2 \sigma^2 T}}\right), \quad (76)$$

where $\kappa := R/r$ and $r, R > 0$ are as in Assumption 2.2.

Theorem F.1. *Let $T \geq 2$ and $\mathbf{c} \in \mathcal{K}$ be given and suppose that Assumption 2.2 and Assumption 2.1 hold with $0 < r \leq R$ and $G > 0$, respectively. Consider a call to Algorithm 2 with input $(T, \mathbf{c}, r, R, G, \eta)$, for η as in (76). Then, for $\kappa := R/r$, we have*

$$\mathbb{E}[f(\hat{\mathbf{w}}_T)] - \inf_{\mathbf{w} \in \mathcal{K}} f(\mathbf{w}) \leq 4R\sigma \cdot \sqrt{\frac{d \log(1 + 9d^2 T^2 \kappa^2)}{T}} + \frac{56d^2 GR(1 + \log 1 + 9d^2 T^2 \kappa^2)}{T},$$

where $\hat{\mathbf{w}}_T := \frac{1}{T} \sum_{t=1}^T \mathbf{w}_t$ and (\mathbf{w}_t) are the iterates of Algorithm 2. The computational cost is at most

$$\mathcal{O}(T \cdot (C_{\text{sep}}(\mathcal{K}) + d^\omega) \cdot \log(T\kappa)).$$

The proof of Theorem F.1 is very similar to that of [19, Theorem 5.1].

Proof. Let $\mathbf{u}^* \in \arg \min_{\mathbf{u} \in \mathcal{K}} f(\mathbf{u})$. Using Jensen's inequality, we get

$$\begin{aligned} \mathbb{E}[f(\hat{\mathbf{w}}_T)] - f(\mathbf{u}^*) &\leq \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^T (f(\mathbf{w}_t) - f(\mathbf{u}^*))\right], \\ &\leq \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^T \langle \bar{\mathbf{g}}_t, \mathbf{w}_t - \mathbf{u}^* \rangle\right], \quad (\text{by convexity and } \bar{\mathbf{g}}_t \in \partial f(\mathbf{w}_t)) \\ &= \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}^* \rangle\right]. \quad (\mathbb{E}[\boldsymbol{\xi}_t] = \mathbf{0} \text{ by Assumption F.1}) \end{aligned} \quad (77)$$

Now, by instantiating the bound in Theorem 4.1 with comparator $\mathbf{u}^* \in \mathcal{K}$, we get:

$$\begin{aligned} \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}^* \rangle &\leq \frac{\eta}{2} \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}^* \rangle^2 \\ &\quad + \frac{d \log(1 + 9d^2 T^2 R^2 / r^2)}{2\eta} + 50d^2 GR(1 + \log(R/r)). \end{aligned} \quad (78)$$

Now, by Assumption F.1 (in particular, the fact that $\mathbf{g}_t = \bar{\mathbf{g}}_t + \boldsymbol{\xi}_t$) together with the fact that $(a+b)^2 \leq 2a^2 + 2b^2$ and $\mathbf{w}_t, \mathbf{u}^* \in \mathbb{B}(R)$, we have

$$\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}^* \rangle^2 \leq 2\langle \bar{\mathbf{g}}_t, \mathbf{w}_t - \mathbf{u}^* \rangle^2 + 8R^2 \|\boldsymbol{\xi}_t\|^2. \quad (79)$$

On the other hand, by definition of \mathbf{u}^* and the facts that $\mathbf{w}, \mathbf{w}_1, \mathbf{w}_2, \dots \in \mathbb{B}(R)$, we have for all $t \in [T]$:

$$\begin{aligned} 2GR &\geq \langle \bar{\mathbf{g}}_t, \mathbf{w}_t - \mathbf{u}^* \rangle \\ &\geq f(\mathbf{w}_t) - f(\mathbf{u}^*), \quad (\text{by convexity of } f \text{ and } \bar{\mathbf{g}}_t \in \partial f(\mathbf{w}_t)) \\ &\geq 0, \end{aligned} \quad (80)$$

where the last inequality follows by the fact that $\mathbf{w}_t \in \mathcal{K}$ (Lemma 4.2) and that \mathbf{u}^* is the minimizer of f within \mathcal{K} . Note that (80) implies that for all $t \in [T]$,

$$|\langle \bar{\mathbf{g}}_t, \mathbf{w}_t - \mathbf{u}^* \rangle| \leq \langle \bar{\mathbf{g}}_t, \mathbf{w}_t - \mathbf{u}^* \rangle + \frac{2GR}{T}. \quad (81)$$

Picking up from (79), we get

$$\begin{aligned} &\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}^* \rangle^2 \\ &\leq 2 \sum_{t=1}^T \langle \bar{\mathbf{g}}_t, \mathbf{w}_t - \mathbf{u}^* \rangle^2 + 8R^2 \sum_{t=1}^T \|\boldsymbol{\xi}_t\|^2, \\ &\leq 4GR \sum_{t=1}^T |\langle \bar{\mathbf{g}}_t, \mathbf{w}_t - \mathbf{u}^* \rangle| + 8R^2 \sum_{t=1}^T \|\boldsymbol{\xi}_t\|^2, \quad (\text{by the left-hand side inequality in (80)}) \\ &\leq 4GR \sum_{t=1}^T \langle \bar{\mathbf{g}}_t, \mathbf{w}_t - \mathbf{u}^* \rangle + 8G^2 R^2 + 8R^2 \sum_{t=1}^T \|\boldsymbol{\xi}_t\|^2, \quad (\text{by (81)}) \end{aligned}$$

Plugging this into (78) and rearranging, we get

$$\begin{aligned} &\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}^* \rangle - 2GR\eta \sum_{t=1}^T \langle \bar{\mathbf{g}}_t, \mathbf{w}_t - \mathbf{u}^* \rangle \\ &\leq 4\eta G^2 R^2 + 4\eta R^2 \sum_{t=1}^T \|\boldsymbol{\xi}_t\|^2 + \frac{d \log(1 + 9d^2 T^2 R^2 / r^2)}{2\eta} + 50d^2 GR(1 + \log(R/r)). \end{aligned}$$

Taking the expectation on both sides and using that $\mathbb{E}[\mathbf{g}_t] = \bar{\mathbf{g}}_t$ and $\mathbb{E}[\|\boldsymbol{\xi}_t\|^2] \leq \sigma^2$, we get

$$\begin{aligned} &4\eta R^2 T \sigma^2 + 4\eta G^2 R^2 + \frac{d \log(1 + 9d^2 T^2 R^2 / r^2)}{2\eta} + 50d^2 GR(1 + \log(R/r)) \\ &\geq \mathbb{E} \left[\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}^* \rangle \right] - 2GR\eta \cdot \mathbb{E} \left[\sum_{t=1}^T \langle \bar{\mathbf{g}}_t, \mathbf{w}_t - \mathbf{u}^* \rangle \right], \\ &= (1 - 2GR\eta) \cdot \mathbb{E} \left[\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}^* \rangle \right], \\ &\geq \frac{T}{2} \cdot (\mathbb{E}[f(\bar{\mathbf{w}}_T)] - f(\mathbf{u}^*)), \end{aligned}$$

where the last inequality follows by the fact that $\eta \leq \frac{1}{4GR}$ and (77). Now, dividing by $\frac{T}{2}$ on both sides and rearranging, we get

$$\begin{aligned} \mathbb{E}[f(\bar{\mathbf{w}}_T)] - f(\mathbf{u}^*) &\leq 8\eta R^2 \sigma^2 + \frac{8\eta G^2 R^2}{T} + \frac{d \log(1 + 9d^2 T^2 R^2 / r^2)}{2\eta T} + \frac{50d^2 GR(1 + \log(R/r))}{T}, \\ &\leq 8\eta R^2 \sigma^2 + \frac{GR}{T} + \frac{d \log(1 + 9d^2 T^2 \kappa^2)}{2\eta T} + \frac{50d^2 GR(1 + \log \kappa)}{T}, \end{aligned} \quad (82)$$

where the last inequality follows by $\eta \leq \frac{1}{10dGR}$. Note that the optimal tuning of η in (82) is given by

$$\eta^* = \sqrt{\frac{d \log(1 + d^2 T^2 \kappa^2)}{16R^2 \sigma^2 T}}.$$

We now consider cases.

Case where $\eta^* \leq \frac{1}{10dGR}$. First, note that this implies that $\eta = \eta^*$. And so, plugging this into (82), we get

$$\mathbb{E}[f(\widehat{\mathbf{w}}_T)] - f(\mathbf{u}^*) \leq 4R\sigma \sqrt{\frac{d \log(1 + 9d^2 T^2 \kappa^2)}{T}} + \frac{GR}{T} + \frac{50d^2 GR(1 + \log \kappa)}{T}. \quad (83)$$

Case where $\eta^* \geq \frac{1}{10dGR}$. In this case, we have $\eta = \frac{1}{10dGR}$. Now, using that $\eta^* \geq \frac{1}{10dGR}$ and the expression of η^* , we have

$$\sigma \leq \frac{5dG}{2} \cdot \sqrt{\frac{d \log(1 + 9d^2 T^2 \kappa^2)}{T}}. \quad (84)$$

Plugging $\eta = \frac{1}{10dGR}$ into (82) and using (84), we get

$$\begin{aligned} (\text{case } \eta^* \geq \frac{1}{10dGR}) \quad & \mathbb{E}[f(\widehat{\mathbf{w}}_T)] - f(\mathbf{u}^*) \\ & \leq 8\eta R^2 \sigma^2 + \frac{GR}{T} + \frac{d \log(1 + 9d^2 T^2 \kappa^2)}{2\eta T} + \frac{50d^2 GR(1 + \log \kappa)}{T}, \\ & = 2R\sigma \cdot \sqrt{\frac{d \log(1 + 9d^2 T^2 \kappa^2)}{T}} + \frac{GR}{T} + \frac{5d^2 GR \log(1 + 9d^2 T^2 \kappa^2)}{T} \\ & \quad + \frac{50d^2 GR(1 + \log \kappa)}{T}, \end{aligned} \quad (85)$$

where the last inequality follows by (84). Thus, combining (83) and (85), we get

$$\mathbb{E}[f(\widehat{\mathbf{w}}_T)] - f(\mathbf{u}^*) \leq 4R\sigma \cdot \sqrt{\frac{d \log(1 + 9d^2 T^2 \kappa^2)}{T}} + \frac{56d^2 GR(1 + \log 1 + 9d^2 T^2 \kappa^2)}{T}.$$

This proves the desired convergence rate. \square

G Helper Lemmas

Lemma G.1 ([21]). Let $A, E \in \mathbb{R}^{d \times d}$ be such that A is invertible and $r = \|A^{-1}E\| < 1$. Then,

$$\|(A + E)^{-1} - A^{-1}\| \leq \|E\| \|A^{-1}\|^2 / (1 - r).$$

Lemma G.2. Let \mathcal{C} be a convex set and let $H \in \mathbb{S}_{>0}^{d \times d}$ and $\mathbf{z} \in \mathbb{R}^d$ be given. Further, let $\mathbf{u}^* \in \arg \min_{\mathbf{u} \in \mathcal{C}} \|\mathbf{u} - \mathbf{z}\|_H$. Then,

$$\forall \mathbf{u} \in \mathcal{C}, \quad \|\mathbf{z} - \mathbf{u}\|_H \geq \|\mathbf{u}^* - \mathbf{u}\|_H.$$

Proof. See, e.g., [11]. \square

Lemma G.3. Let $r > 0$, $\mathbf{c}_0, \mathbf{c} \in \mathbb{R}^d$, and $H \in \mathbb{S}_{>0}^{d \times d}$ be given, and define $\mathcal{E} := \{\mathbf{u} \in \mathbb{R}^d \mid (\mathbf{u} - \mathbf{c})^\top H^{-1}(\mathbf{u} - \mathbf{c}) \leq 1\}$. If $\mathbb{B}(\mathbf{c}_0, r) \subseteq \mathcal{E}$, then $\mathbb{B}(\mathbf{c}, r) \subseteq \mathcal{E}$.

Proof. Let $\mathcal{X}^+ := \mathbb{B}(\mathbf{c}_0, r)$ and suppose that $\mathcal{X}^+ \subseteq \mathcal{E}$. Since \mathcal{E} is centrally-symmetric around \mathbf{c} (i.e., $-(\mathbf{u} - \mathbf{c}) + \mathbf{c} \in \mathcal{E}$ for all $\mathbf{u} \in \mathcal{E}$), we have $-(\mathbb{B}(\mathbf{c}_0, r) - \mathbf{c}) + \mathbf{c} \subseteq \mathcal{E}$. Since $\mathbb{B}(r) = -\mathbb{B}(r)$, this implies that

$$\mathcal{X}^- := \mathbb{B}(-\mathbf{c}_0 + 2\mathbf{c}, r) \subseteq \mathcal{E}.$$

Now, since \mathcal{E} is convex, we have that

$$\frac{1}{2}\mathcal{X}^+ + \frac{1}{2}\mathcal{X}^- \subseteq \mathcal{E}. \quad (86)$$

Fix $\mathbf{z} \in \mathbb{B}(r)$. We have that $\mathbf{z} - \mathbf{c}_0 + 2\mathbf{c} \in \mathcal{X}^-$ and $\mathbf{z} + \mathbf{c}_0 \in \mathcal{X}^+$ and so by (86), we have $\mathbf{z} + \mathbf{c} \in \mathcal{E}$, which establishes that $\mathbb{B}(\mathbf{c}, r) \subseteq \mathcal{E}$ and completes the proof. \square

Lemma G.4. *Let $H \in \mathbb{R}^{d \times d}$ be a positive semi-definite matrix, and let $\mathbf{v} \in \mathbb{R}^d \setminus \{0\}$ be given. Define*

$$H_{\mathbf{v}} := \frac{4d^2 - 1}{4d^2 - 4} \cdot \left(H - \frac{2d}{2d^2 + d - 1} \frac{H\mathbf{v}\mathbf{v}^\top H}{\mathbf{v}^\top H\mathbf{v}} \right).$$

Then, we have $\sigma_{\max}(H_{\mathbf{v}}) \leq (1 + 2/d^2) \cdot \sigma_{\max}(H)$.

Proof of Lemma G.4. Since $\left(H - \frac{2d}{2d^2 + d - 1} \frac{H\mathbf{v}\mathbf{v}^\top H}{\mathbf{v}^\top H\mathbf{v}} \right) \preceq H$, we have that

$$H_{\mathbf{v}} \preceq \frac{4d^2 - 1}{4d^2 - 4} \cdot H \preceq \left(1 + \frac{2}{d^2} \right) \cdot H,$$

where the last inequality follows from the fact that $\frac{4d^2 - 1}{4d^2 - 4} \leq 1 + \frac{2}{d^2}$ for $d \geq 2$ and that H is positive semi-definite. This implies that $\sigma_{\max}(H_{\mathbf{v}}) \leq (1 + 2/d^2) \cdot \sigma_{\max}(H)$. \square

We need the following properties of the Gauge function (see e.g. [20] for a proof).

Lemma G.5. *Let $\mathbf{w} \in \mathbb{R}^d \setminus \{0\}$ and $0 < r \leq R$. Further, let \mathcal{C} be a closed convex set such that $\mathcal{B}(r) \subseteq \mathcal{C} \subseteq \mathcal{B}(R)$. Then, the following properties hold:*

- a. $\gamma_{\mathcal{C}}(\mathbf{w}) = \sigma_{\mathcal{C}^\circ}(\mathbf{w}) = \sup_{\mathbf{x} \in \mathcal{C}^\circ} \mathbf{x}^\top \mathbf{w}$ and $(\mathcal{C}^\circ)^\circ = \mathcal{C}$.
- b. $\sigma_{\mathcal{C}}(\alpha \mathbf{w}) = \alpha \sigma_{\mathcal{C}}(\mathbf{w})$ and $\partial \sigma_{\mathcal{C}}(\alpha \mathbf{w}) = \partial \sigma_{\mathcal{C}}(\mathbf{w}) = \arg \max_{\mathbf{u} \in \mathcal{C}} \langle \mathbf{u}, \mathbf{w} \rangle$, for all $\alpha \geq 0$.
- c. $r \|\mathbf{w}\| \leq \sigma_{\mathcal{C}}(\mathbf{w}) \leq R \|\mathbf{w}\|$, $\|\mathbf{w}\|/R \leq \gamma_{\mathcal{C}}(\mathbf{w}) \leq \|\mathbf{w}\|/r$, and $\mathcal{B}(1/R) \subseteq \mathcal{C}^\circ \subseteq \mathcal{B}(1/r)$.