

# Gradient Methods with Online Scaling

Wenzhi Gao<sup>\*1</sup>, Ya-Chi Chu<sup>†2</sup>, Yinyu Ye<sup>‡1,3</sup>, and Madeleine Udell<sup>§1,3</sup>

<sup>1</sup>ICME, Stanford University

<sup>2</sup>Department of Mathematics, Stanford University

<sup>3</sup>Department of Management Science and Engineering, Stanford University

November 5, 2024

## Abstract

We introduce a framework to accelerate the convergence of gradient-based methods with online learning. The framework learns to scale the gradient at each iteration through an online learning algorithm and provably accelerates gradient-based methods asymptotically. In contrast with previous literature, where convergence is established based on worst-case analysis, our framework provides a strong convergence guarantee with respect to the optimal scaling matrix for the *iteration trajectory*. For smooth strongly convex optimization, our results provide an  $\mathcal{O}(\kappa^* \log(1/\varepsilon))$  complexity result, where  $\kappa^*$  is the condition number achievable by the optimal preconditioner, improving on the previous  $\mathcal{O}(\sqrt{n}\kappa^* \log(1/\varepsilon))$  result. In particular, a variant of our method achieves superlinear convergence on convex quadratics. For smooth convex optimization, we show for the first time that the widely-used hypergradient descent heuristic improves on the convergence of gradient descent.

## 1 Introduction

We consider the unconstrained smooth strongly convex optimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x),$$

where  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  is  $L$ -smooth and  $\mu$ -strongly convex with  $f(x^*) := \min_x f(x) > -\infty$ . It is well-known that gradient descent with stepsize  $1/L$  converges with iteration complexity  $\mathcal{O}(\kappa \log(1/\varepsilon))$ , where  $\kappa = L/\mu$  is the condition number of the problem. Two major techniques have been developed in the literature to accelerate gradient descent. One is to improve the dependence on  $\kappa$  through Nesterov's fast gradient method [31, 32]; the other is through preconditioning: a positive definite matrix stepsize  $P$ , known as preconditioner, premultiplies the gradient to improve convergence:

$$x^{k+1} = x^k - P\nabla f(x^k).$$

Preconditioning has been a standard tool in convex optimization and numerical linear algebra to improve convergence of gradient descent [27, 29, 25, 14, 15] or other iterative methods [37], and it is closely related to the well-known adaptive gradient methods [12, 22], either for online learning or for a general optimization problem. Some recent results quantify the effect of adaptive methods on problem conditioning [7]. In the context of machine learning, adaptively choosing a preconditioner is also relevant to hyperparameter tuning

---

\*gwz@stanford.edu

†ycchu97@stanford.edu

‡yye@stanford.edu

§udell@stanford.edu

[20, 31], especially choosing a learning rate schedule [9].

Despite the great empirical success of adaptive methods in practice, they usually cannot improve the theoretical complexity as a function of the condition number. Recently, [23] showed that hypergradient, the gradient of the optimization objective with respect to the preconditioner, can be used to improve the convergence of gradient descent. The idea in [23] is to use a cutting plane subroutine to update the (diagonal) preconditioner, and an  $\mathcal{O}(\sqrt{n\kappa^*} \log(1/\varepsilon))$  complexity result is obtained, where  $n$  is the variable dimension and  $\kappa^*$  is the condition number of the optimally preconditioned problem. Although the result in [23] is dimension-dependent and requires a nontrivial subroutine to update the preconditioner, it provides a valuable direction to improve the performance of first-order adaptive methods theoretically. Whether a simple adaptive first-order method can achieve  $\mathcal{O}(\kappa^* \log(1/\varepsilon))$  complexity or even stronger guarantees remains open.

This paper answers this question affirmatively by proposing the online scaled gradient method, a framework that accelerates gradient-based methods through online convex optimization.

### Contributions.

- We develop a framework that accelerates gradient-based algorithms through online learning. Unlike previous work, our framework guarantees convergence with respect to the scaling matrix optimized for the iteration trajectory, rather than the worst-case analysis.
- We propose a simple adaptive first-order gradient-based method with asymptotic  $\mathcal{O}(\kappa^* \log(1/\varepsilon))$  complexity, where  $\kappa^*$  is the optimal condition number achievable by the optimal preconditioner, improving on the  $\mathcal{O}(\sqrt{n\kappa^*} \log(1/\varepsilon))$  complexity in the previous literature. In particular, one realization of our framework achieves superlinear convergence on strongly convex quadratics using first-order information.
- For the first time, we prove that the hypergradient heuristic improves the convergence of gradient descent.

## 1.1 Related literature

**Preconditioned iterative methods.** Preconditioning is a well-established technique to enhance the convergence of iterative algorithms in both optimization [15, 33, 1, 10] and numerical linear algebra [37, 35, 17, 16, 11]. By applying a linear transformation to the optimization variables, preconditioning aims to reduce the heterogeneity of the optimization landscape. Recent research has focused on understanding the properties of optimal preconditioners [35, 17, 21]. While these methods demonstrate empirical success, identifying a good preconditioner can be computationally intensive and often depends on the specific structure of the problem.

**Hypergradient descent heuristic.** Our method is closely related to the hypergradient descent heuristic [2, 28, 3], which updates the stepsize (hyperparameters) using the gradient of the optimization objective with respect to it. Despite strong empirical results [2, 3], the theoretical understanding of hypergradient descent remains limited. The existing results [36] are unable to fully justify the observed improvements [23]. Recently, [23] introduced a novel multi-dimensional backtracking approach that uses hypergradients to generate separating hyperplanes in the space of candidate preconditioners. This work provides the first theoretical justification for hypergradient descent. However, it solves a small cutting plane subproblem at every step and incurs a  $\sqrt{n}$  dependence on the problem dimension. No theoretical proof exists that the original hypergradient descent heuristics accelerate gradient-based methods. And our paper provides the first proof that quantifies the acceleration effect of the hypergradient descent heuristic.

**Adaptive first-order methods.** Adaptive stepsize is a well-established technique to enhance the convergence of optimization algorithms. The most notable example is **AdaGrad** [12, 30], which provides strong theoretical guarantees in the context of online convex optimization. Other methods, such as **Adam** [22] and **RMSProp** [19], have demonstrated competitive empirical performance, though they generally yield weaker online regret bounds. Our approach also leverages online learning techniques to accelerate gradient-based methods,

with a particular focus on improving the dependence on problem conditioning. We show that online learning applied to the hypergradient gives optimal guarantees on problem conditioning.

**Learning to optimize and meta-learning.** The learning to optimize [26, 5] and meta-learning [20, 6, 13] literatures also use online learning to improve algorithm performance. These approaches are typically designed to solve a sequence of related optimization problems, providing performance guarantees across multiple tasks. In contrast, our work applies online learning to improve first-order methods over the course of solving a single optimization instance.

## 2 Background and preliminaries

**Notations.** Throughout the paper, we use  $\|\cdot\|$  to denote vector Euclidean norm or matrix spectral norm, and  $\langle \cdot, \cdot \rangle$  to denote Euclidean inner product. Letters  $A$  and  $a$  denote matrices and vectors, respectively.  $\|A\|_F = \sqrt{\sum_{ij} a_{ij}^2}$  denotes the matrix Frobenius norm. Given two vectors  $a, b$  of the same dimension,  $a \odot b$  denotes their element-wise Hadamard product. The Clarke subdifferential of a function  $f(x)$  at  $x$  is defined by  $\partial f(x) := \{v \in \mathbb{R}^n : f(y) \geq f(x) + \langle v, y - x \rangle + o(\|x - y\|), y \rightarrow x\}$ . We use  $f'(x) \in \partial f(x)$  to denote a subgradient. If  $f(x)$  is differentiable at  $x$ ,  $\partial f(x) = \{\nabla f(x)\}$ . For symmetric matrices  $A, B$ ,  $A \succeq B$  if  $A - B \in \mathbb{S}_+^n$  is positive semidefinite. The condition number for an  $L$ -smooth and  $\mu$ -strongly convex problem is defined by  $\kappa = L/\mu$ . Given a closed convex set  $C$ ,  $\Pi_C[x]$  denotes the orthogonal projection of  $x$  onto  $C$ . We use  $\mathcal{L}_\alpha := \{x : f(x) \leq \alpha\}$  to denote the  $\alpha$ -sublevel set of  $f$  and  $\mathcal{X}^*$  to denote the optimal set of  $f$ .  $x$  is an  $\varepsilon$ -optimal solution if  $f(x) \leq f(x^*) + \varepsilon$  and  $x$  is an  $\varepsilon$ -critical point if  $\|\nabla f(x)\| \leq \varepsilon$ .

### 2.1 Assumptions

We make the following two assumptions throughout the paper.

**A1:**  $f(x)$  is  $L$ -smooth.  $|f(x) - f(y) - \langle \nabla f(y), x - y \rangle| \leq \frac{L}{2} \|x - y\|^2$

**A2:**  $f(x)$  is  $\mu$ -strongly convex.  $f(x) - f(y) - \langle \nabla f(y), x - y \rangle \geq \frac{\mu}{2} \|x - y\|^2$

We also assume that  $f$  is twice-differentiable for simplicity. However, our algorithm does not necessarily require twice-differentiability to work. In addition,  $\mu$ -strong convexity can be relaxed to weaker conditions such as convexity with quadratic growth [31].

### 2.2 Preconditioned and scaled gradient method

It is well-known that under **A1** and **A2**, the vanilla gradient descent

$$x^+ = x - \frac{1}{L} \nabla f(x)$$

achieves  $\mathcal{O}(\kappa \log(1/\varepsilon))$  complexity [18]. The dependence on the condition number  $\kappa$  is unfortunate since the condition number can be very large and substantially slows down convergence. Two techniques are often used to improve dependence on  $\kappa$ . One is through Nesterov's fast gradient method [31], which achieves  $\mathcal{O}(\sqrt{\kappa} \log(1/\varepsilon))$  complexity; the other is through preconditioning [14, 27, 15], which replaces the scalar stepsize by some positive semidefinite matrix  $P \in \mathbb{S}_+^n$

$$x^+ = x - P \nabla f(x). \tag{1}$$

Typically,  $P$  is chosen to be diagonal and positive definite. This paper also considers updates of the form (1) but allows  $P$  to be an arbitrary matrix of proper dimension from some closed convex set  $\mathcal{P}$ . Moreover,  $P$  is allowed to vary across iterations:

$$x^{k+1} = x^k - P_k \nabla f(x^k) \tag{2}$$

To differentiate our method from standard preconditioning techniques, which generally consider only positive definite preconditioners, we call our method the scaled gradient method:  $P$  serves as a (not necessarily positive

definite or symmetric) scaling matrix. Preconditioned gradient descent can be viewed as a special case of the scaled gradient method. We define  $\mathcal{P}_+ := \mathcal{P} \cap \mathbb{S}_+^n$  when we switch to the context of preconditioned gradient descent, and without loss of generality we assume  $0 \in \mathcal{P}$ ,  $L^{-1}I \in \mathcal{P}$  and that  $\mathcal{P}$  is bounded.

**A3:** Closed convex set  $\mathcal{P}$  satisfies  $0 \in \mathcal{P}$ ,  $L^{-1}I \in \mathcal{P}$  and  $\text{diam}(\mathcal{P}) \leq D$ .

### 2.3 Monotone descent oracle

When  $P$  is not positive definite, a scaled gradient update will not necessarily decrease the function value. To guarantee convergence under weak assumptions, the scaled gradient method optionally uses a *monotone descent oracle*  $\mathcal{M}$ , defined below.

**Definition 2.1.** Given the scaled gradient update  $x^+ = x - P\nabla f(x)$ ,  $\mathcal{M}_{\varphi, P} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is called a *monotone descent oracle* associated with the scaled gradient update and measure  $\varphi$  if its output  $\mathcal{M}_{\varphi, P}(x)$  satisfies

$$\varphi(\mathcal{M}_{\varphi, P}(x)) \leq \min\{\varphi(x), \varphi(x^+)\}.$$

We use  $\mathcal{M}(x)$  to denote the oracle when the context is clear. Three typical realizations of  $\mathcal{M}$  are as follows.

- *Line-search.*

$\mathcal{M}(x) = x + \alpha(x^+ - x)$  such that  $\varphi(\mathcal{M}(x)) \leq \varphi(x)$ . Additional regularity conditions, such as  $\mathcal{P} = \mathcal{P}_+$ , are required to ensure that line-search stops in a finite number of steps.

- *Steepest descent.*

$\mathcal{M}(x) = x + \alpha(x^+ - x)$  and  $\alpha = \arg \min_{\alpha} \varphi(x + \alpha(x^+ - x))$ . It applies to simple functions such as quadratic.

- *Simple comparison.*

$\mathcal{M}(x) = x^+$  if  $\varphi(x^+) \leq \varphi(x)$ . Otherwise  $\mathcal{M}(x) = x$ . It takes one extra measure evaluation.

## 3 Online Scaled Gradient Methods

This section introduces our main methodology, which relates the scaled gradient method to online convex optimization with  $P$  as the decision variable.

### 3.1 Scaled gradient method and online learning

Let  $\varphi(x)$  be a non-negative measure or potential energy that characterizes the optimality of  $x$ . For example, function value gap  $\varphi(x) = f(x) - f(x^*)$  and gradient norm  $\varphi(x) = \|\nabla f(x)\|$  are common measures. The progress of an algorithm at step  $K + 1$  with respect to measure  $\varphi$  can be expressed as the telescoping product

$$\varphi(x^{K+1}) = \varphi(x^1) \prod_{k=1}^K \frac{\varphi(x^{k+1})}{\varphi(x^k)}. \quad (3)$$

Then the arithmetic-geometric mean inequality upper-bounds  $\varphi(x^{K+1})$ :

**Theorem 3.1.** Given a non-negative function  $\varphi(x) : \mathbb{R}^n \rightarrow \mathbb{R}_+$  and a sequence of iterations  $\{x^k\}$ ,

$$\varphi(x^{K+1}) \leq \varphi(x^1) \left( \frac{1}{K} \sum_{k=1}^K \frac{\varphi(x^{k+1})}{\varphi(x^k)} \right)^K.$$

The quantity  $\frac{1}{K} \sum_{k=1}^K \frac{\varphi(x^{k+1})}{\varphi(x^k)}$  on the right-hand side is the averaged contraction factor across all previous iterates: a smaller contraction factor ensures stronger convergence. Suppose the iterates  $\{x^k\}_{k \geq 2}$  are generated

by the scaled gradient method in (2). Then

$$\frac{1}{K} \sum_{k=1}^K \frac{\varphi(x^{k+1})}{\varphi(x^k)} = \frac{1}{K} \sum_{k=1}^K \frac{\varphi(x^k - P_k \nabla f(x^k))}{\varphi(x^k)}. \quad (4)$$

To maximize the progress in the scaled gradient method, we aim to minimize the quantity in (4) over the choice of scaling matrices  $P_k$  with online learning. We will show that online convex optimization can learn a sequence of  $\{P_k\}$  that asymptotically accelerates gradient-based methods. Define the *surrogate loss*

$$\ell_x(P) := \frac{\varphi(x - P \nabla f(x))}{\varphi(x)}$$

with respect to the measure  $\varphi$ . Note that  $\ell_{x^k}$  only depends on  $x^1$  and all previous scaling matrices  $\{P_j\}_{j \leq k-1}$ . Online learning generates a sequence  $\{P_k\}$  such that the cumulative regret is bounded by  $\rho_K$ :

$$\sum_{k=1}^K \ell_{x^k}(P_k) - \min_{P \in \mathcal{P}} \sum_{k=1}^K \ell_{x^k}(P) \leq \rho_K. \quad (5)$$

Existing results in online optimization can guarantee sublinear regret if the losses  $\{\ell_{x^k}\}$  are convex and are either Lipschitz continuous or have Lipschitz continuous gradient [34]. In this case, we say the family of surrogate losses  $\{\ell_{x^k}\}$  is *online-learnable*. The definition of regret  $\rho_K$  and **Theorem 3.1** imply

$$\varphi(x^{K+1}) \leq \varphi(x^1) \left( \frac{1}{K} \sum_{k=1}^K \ell_{x^k}(P) \right)^K \leq \varphi(x^1) \left( \min_{P \in \mathcal{P}} \frac{1}{K} \sum_{k=1}^K \ell_{x^k}(P) + \frac{\rho_K}{K} \right)^K. \quad (6)$$

When the regret  $\rho_K$  grows sublinearly in  $K$ , the bound in (6) suggests that for large enough  $K$ ,

$$\varphi(x^{K+1}) \leq \varphi(x^1) \left( \min_{P \in \mathcal{P}} \frac{1}{K} \sum_{k=1}^K \ell_{x^k}(P) + \frac{\rho_K}{K} \right)^K \approx \varphi(x^1) \left( \min_{P \in \mathcal{P}} \frac{1}{K} \sum_{k=1}^K \ell_{x^k}(P) \right)^K.$$

This result is powerful: it suggests that a scaled gradient method, in the long run, can achieve convergence that is competitive with any fixed scaling matrix optimized for the *iteration trajectory*. To the best of our knowledge, this trajectory-based convergence guarantee is rare in the literature. Moreover, as long as there exists some pre-specified scaling matrix  $P^*$  (or simply stepsize  $P^* = \alpha I$ ) such that  $\ell_x(P^*) \leq \theta^* < 1$  for any  $x$ , we obtain the global convergence guarantee

$$\varphi(x^{K+1}) \leq \varphi(x^1) (\theta^* + \frac{\rho_K}{K})^K \approx \varphi(x^1) (\theta^*)^K \quad (7)$$

The algorithm, which updates the scaling matrix  $P_k$  on the fly, is called a realization of the *online scaled gradient method (OSGM)*.

## 3.2 Framework of online scaled gradient method

The online scaled gradient method is determined by the components below:

- *Optimality measure*. A measure  $\varphi$  to characterize the convergence of **OSGM**.
- *Surrogate loss*. A surrogate loss  $\ell_x(P)$  that relates  $\varphi$  with an online learning problem in  $P$ .
- *Online learning algorithm*. An online learning algorithm  $\mathcal{A}$  that guarantees sublinear regret for  $\sum_k \ell_{x^k}(P_k)$ .
- (Optional) *Monotone oracle*. An oracle  $\mathcal{M}$  (**Definition 2.1**) that guarantees monotonicity.
- (Theoretical purpose) *Hindsight scaling matrix*. A hindsight scaling matrix  $P^*$  to ensure global convergence.

The tuple  $(\varphi, \ell, \mathcal{A}, \mathcal{M})$  determines a realization of the online scaled gradient method (**OSGM, Algorithm 1**). The surrogate loss  $\ell_x$  is the key component of this framework. Finding a good surrogate, for given assumptions on the function to be optimized, is the challenging aspect of the theoretical analysis. In the rest of the paper, we provide several realizations of the framework for different function classes, summarized in **Table 1**.

---

**Algorithm 1:** Online scaled gradient method (OSGM)

---

```

input  $x^1, P_1, \varphi, \ell, \mathcal{A}, \mathcal{M}$ 
for  $k = 1, 2, \dots$  do
  if  $\mathcal{M} = \emptyset$  then
     $x^{k+1} = x^k - P_k \nabla f(x^k)$ 
  else
     $x^{k+1} = \mathcal{M}_{\varphi, P_k}(x^k)$ 
  end
   $P_{k+1} = \mathcal{A}(\ell_{x^k}, P_k)$ 
end
output  $x^{\text{best}}$  with minimum objective value

```

---

$\varphi(x)$	Surrogate $\ell$	Strong convexity	$\mathcal{A}$	$\mathcal{M}$ oracle	Complexity	Reference
$f(x) - f(x^*)$	$r_x(P) = \frac{f(x^+) - f(x^*)}{f(x) - f(x^*)}$	Yes		Optional	$\mathcal{O}(\kappa^* \log(\frac{1}{\varepsilon}))$	<b>Section 4</b>
$\ \nabla f(x)\ $	$g_x(P) = \frac{\ \nabla f(x^+)\ }{\ \nabla f(x)\ }$	Yes	OGD	Required	$\mathcal{O}(\lambda^* \log(\frac{1}{\varepsilon}))$	<b>Section 5</b>
$f(x) - f(x^*)$	$h_x(P) = \frac{f(x^+) - f(x)}{\ \nabla f(x)\ ^2}$	Yes No		Required Required	$\mathcal{O}(\frac{1}{2\mu\gamma^*} \log(\frac{1}{\varepsilon}))$ $\mathcal{O}(\frac{1}{\gamma^*\varepsilon})$	<b>Section 6</b>

---

Table 1: Realizations of OSGM. OGD: online (sub)gradient method.  $\kappa^*, \lambda^*, \gamma^*$  and their optimal scaling matrix  $P^*$  will be defined in the next sections.

## 4 Function value ratio surrogate

The first surrogate, function value *ratio surrogate*, is defined as follows:

$$r_x(P) := \frac{f(x^+) - f(x^*)}{f(x) - f(x^*)} = \frac{f(x - P\nabla f(x)) - f(x^*)}{f(x) - f(x^*)}. \quad (8)$$

The ratio surrogate  $r_x$  measures the contraction factor of the function value gap between two consecutive OSGM steps. We assume strong convexity ( $\mu > 0$ ) throughout this section. Without loss of generality, we assume all the intermediate iterates generated by the algorithm satisfy  $f(x) > f(x^*)$  so that  $r_x$  is well-defined; otherwise,  $x \in \mathcal{X}^*$  and we can immediately stop the algorithm. The ratio surrogate  $r_x$  assumes the optimal value  $f(x^*)$  is known, and this assumption will be relaxed later in this section. The monotone oracle is optional for  $r_x$ . We present the results without a monotone oracle:  $\mathcal{M} = \emptyset$ .

### 4.1 Surrogate loss

The function value ratio  $r_x$  in (8) can be viewed as a surrogate loss since its average along OSGM iterates serves as an upper bound on the function value gap. By substituting  $\varphi(x) = f(x) - f(x^*)$  in **Theorem 3.1** and plugging in the definition of  $r_x$ , OSGM iterates  $\{x^k\}$  satisfy the following bound:

**Lemma 4.1** (Surrogate loss and measure). *For all  $K \geq 1$ , the online scaled gradient method satisfies*

$$f(x^{K+1}) - f(x^*) \leq (f(x^1) - f(x^*)) \left( \frac{1}{K} \sum_{k=1}^K r_{x^k}(P_k) \right)^K. \quad (9)$$

The ratio surrogate  $r_x$  inherits several important properties from  $f$ , which we summarize in **Proposition 4.1**. These properties are crucial for establishing the online learnability of  $\{r_{x^k}\}$  later in **Section 4.2**.

**Proposition 4.1** (Properties of  $r_x$ ). *Under **A1** and **A2**, for any fixed  $x \notin \mathcal{X}^*$ , the surrogate loss  $r_x(P)$  defined in (8) is convex, non-negative, and  $2L^2$ -smooth as a function in  $P$ . In addition, the derivative of  $r_x$  takes the*

form

$$\nabla r_x(P) = -\frac{\nabla f(x - P\nabla f(x))\nabla f(x)^\top}{f(x) - f(x^*)}. \quad (10)$$

## 4.2 Online learning algorithm

Online gradient descent is known to ensure sublinear regret for a family of smooth, convex, and lower-bounded losses [34], which is the case for ratio surrogate loss  $r_x$  by **Proposition 4.1**. We tailor the classical  $L^*$  regret bound from online convex optimization literature [34] to our settings in **Lemma 4.2** below.

**Lemma 4.2** (Learnability). *Given **A1**, **A2**, and the ratio surrogate losses  $\{r_{x^k}\}$ , online gradient descent*

$$P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta\nabla r_{x^k}(P_k)] \quad (11)$$

with stepsize  $\eta \leq 1/(4L^2)$  generates a sequence of scaling matrices  $\{P_k\}_{k \geq 2}$  such that

$$\sum_{k=1}^K r_{x^k}(P_k) - \sum_{k=1}^K r_{x^k}(P) \leq \frac{1}{\eta} \|P - P_1\|_F^2 + 4L^2\eta \sum_{k=1}^K r_{x^k}(P) \quad \text{for any } P \in \mathcal{P}. \quad (12)$$

In particular, if **A3** is further assumed, the choice of stepsize  $\eta = \min\left\{\frac{1}{4L^2}, \frac{D}{2L(1+LD)\sqrt{K}}\right\}$  ensures

$$\sum_{k=1}^K r_{x^k}(P_k) - \min_{P \in \mathcal{P}} \sum_{k=1}^K r_{x^k}(P) \leq \rho_K := \max\{4LD(1+LD)\sqrt{K}, 8L^2D^2\}. \quad (13)$$

*Remark 1.* The relation (11) suggests additional complexity from a rank-one update with an orthogonal projection. But as we will discuss in **Section 7**, we can choose  $\mathcal{P}$  to have arbitrary sparsity (e.g., diagonal), and it is only necessary to update the nonzero elements. Moreover, the orthogonal projection is often easy to compute since we do not require  $P_k$  to be positive semidefinite or symmetric.

## 4.3 Algorithm design and analysis

We now state a realization of OSGM with the ratio surrogate loss  $r_x$ , denoted by OSGM-R. We choose the optimality measure  $\varphi$ , the surrogate loss  $\ell$ , and the online learning algorithm  $\mathcal{A}$  to be

$$\varphi(x) := f(x) - f(x^*), \quad \ell_x(P) := r_x(P), \quad \mathcal{A} := \text{online gradient descent in (11)},$$

and the monotone oracle  $\mathcal{M}$  is optional. **Algorithm 2** presents OSGM-R without the monotone oracle.

---

**Algorithm 2:** Online scaled gradient method with ratio surrogate (OSGM-R)

---

**input**  $x^1, P_1 \in \mathcal{P}$ , online gradient stepsize  $\eta > 0$   
**for**  $k = 1, 2, \dots$  **do**  
     $x^{k+1} = x^k - P_k \nabla f(x^k)$   
     $P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta \nabla r_{x^k}(P_k)]$   
**end**  
**output**  $x^{\text{best}}$  with minimum objective value

---

Combining **Lemma 4.1** and **Lemma 4.2**, **Theorem 4.1** characterizes the trajectory-based convergence behavior of OSGM-R.

**Theorem 4.1** (Trajectory-based convergence). *Under **A1** to **A3**, **Algorithm 2** (OSGM-R) with  $\eta = \min\left\{\frac{1}{4L^2}, \frac{D}{2L(1+LD)\sqrt{K}}\right\}$  satisfies*

$$f(x^{K+1}) - f(x^*) \leq (f(x^1) - f(x^*))(\theta_K^* + \frac{\rho_K}{K})^K, \quad (14)$$

where  $\theta_K^* := \min_{P \in \mathcal{P}} \frac{1}{K} \sum_{k=1}^K r_{x^k}(P)$  and  $\rho_K = \max\{4LD(1+LD)\sqrt{K}, 8L^2D^2\}$  is defined in (13).

From (14), when  $K$  is large enough,  $\frac{\rho_K}{K}$  is negligible, and OSGM-R behaves like an algorithm with linear convergence rate  $\theta_K^*$ . Note that  $\theta_K^*$  is based on the optimization trajectory, and the behavior of OSGM-R is competitive with the scaling matrix that minimizes  $\theta_K^*$ . To our knowledge, this trajectory-based convergence guarantee is new in the literature. To show global convergence, we need to show that  $\theta_K^* < 1$ , and this fact follows from the existence of the *optimal preconditioner*.

### 4.3.1 Hindsight and global convergence

Define  $P_r^*$  to be the ratio scaling matrix that solves the following semidefinite optimization problem

$$\kappa^* := \min_{P \in \mathcal{P}_+} \kappa \quad \text{subject to} \quad \frac{1}{\kappa} I \preceq P^{1/2} \nabla^2 f(x) P^{1/2} \preceq I \quad \text{for all } x. \quad (15)$$

$P_r^*$  is known in the literature [23, 35] as the *universal optimal preconditioner*. The optimal value  $\kappa^*$  is called the *optimal condition number* with respect to subset  $\mathcal{P}_+ = \mathcal{P} \cap \mathbb{S}_+^n$ . Since **A3** assumes  $L^{-1}I \in \mathcal{P}$ , we have  $\kappa^* \leq \kappa$  and preconditioned gradient descent with preconditioner  $P_r^*$  converges as if the condition number of the underlying minimization problem is reduced from  $\kappa = L/\mu$  to  $\kappa^*$ . A standard argument using the descent lemma and strong convexity of  $f(x)$  (**A2**) ensures

$$f(x - P_r^* \nabla f(x)) - f(x^*) \leq (1 - \frac{1}{\kappa^*})(f(x) - f(x^*)) \quad \text{for all } x, \quad (16)$$

which can be equivalently expressed in terms of the ratio surrogate loss  $r_x$  in the lemma below:

**Lemma 4.3** (Hindsight). *Under **A1** to **A3**,  $r_x(P_r^*) \leq 1 - \frac{1}{\kappa^*}$  for all  $x \notin \mathcal{X}^*$ .*

Combining **Theorem 4.1** and  $\theta_K^* \leq 1 - \frac{1}{\kappa^*}$  from **Lemma 4.3**, the asymptotic linear convergence of OSGM-R follows immediately.

**Corollary 4.1** (Global convergence). *Under the same assumptions as **Theorem 4.1**,  $\theta_K^* \leq 1 - \frac{1}{\kappa^*}$  and the asymptotic complexity of OSGM-R to find an  $\varepsilon$ -optimal point is  $\mathcal{O}(\kappa^* \log(1/\varepsilon))$ , where  $\kappa^*$  is the optimal condition number defined in (15).*

In fact, a slightly better convergence result can be obtained by evaluating the regret bound (12) at  $P = P_r^*$ , which we state as the following theorem.

**Theorem 4.2** (Refined global convergence). *Under **A1** to **A2**, **Algorithm 2** (OSGM-R) with  $\eta = \min\{\frac{1}{4L^2}, \frac{\|P_r^* - P_1\|_F}{2L\sqrt{K}}\}$  satisfies*

$$f(x^{K+1}) - f(x^*) \leq (f(x^1) - f(x^*)) \left(1 - \frac{1}{\kappa^*} + \max\left\{\frac{4L\|P_r^* - P_1\|_F}{\sqrt{K}}, \frac{8L^2\|P_r^* - P_1\|_F^2}{K}\right\}\right)^K. \quad (17)$$

*Remark 2.* Note that **Theorem 4.2** has no dependence on  $D$ , the diameter of  $\mathcal{P}$ . Therefore, OSGM-R can be applied even if  $\mathcal{P} = \mathbb{R}^{n \times n}$ , and there is no need to project  $P$  onto  $\mathcal{P}$ .

The asymptotic linear convergence rate of OSGM-R is comparable to that of preconditioned gradient descent using the universal optimal preconditioner  $P_r^*$ . This result removes the dimension dependence from the  $\mathcal{O}(\sqrt{n}\kappa^* \log(1/\varepsilon))$  result in [23]. As previously remarked, the practical convergence behavior of OSGM-R could be even better: the linear convergence rate  $\theta_K^*$  is determined by the best possible choice of  $P \in \mathcal{P}$  optimized for the iteration trajectory  $\{x^k\}$ , while the universal optimal preconditioner  $P_r^*$  is chosen against all possible  $x$  in the domain. For convex quadratics, we have the following equivalent characterization of  $P_r^*$  through  $r_x$ :

**Proposition 4.2** (Relation between  $P_r^*$  and  $r_x$ ). *For  $f(x) = \frac{1}{2}\langle x, Ax \rangle$ ,  $A \in \mathbb{S}_{++}^n$ , the optimal solutions to the following two problems coincide:*

$$\min_{P \in \mathcal{P}_+} \kappa \quad \text{subject to} \quad \frac{1}{\kappa} I \preceq P^{1/2} A P^{1/2} \preceq I; \quad (18)$$

$$\min_{P \in \mathcal{P}_+} \max_{x \notin \mathcal{X}^*} r_x(P). \quad (19)$$



Since no practical algorithm will visit every  $x \in \mathbb{R}^n$ , the trajectory-based convergence guarantee more precisely characterizes the practical performance of OSGM.

**Unknown optimal value.** Our method can be extended to the case where  $f(x^*)$  is unknown, but instead, a lower bound  $z < f(x^*)$  is available. In this case, we can define the auxiliary surrogate loss:

$$r_x^z(P) := \frac{f(x - P\nabla f(x)) - z}{f(x) - z}, \quad (20)$$

obtained by replacing  $f(x^*)$  in the surrogate loss  $r_x$  with lower bound  $z$ . Using an additional outer loop to update the lower bound  $z$ , the resulting algorithm (**Algorithm 6** in appendix) can achieve  $\mathcal{O}(\kappa^* \log^2(1/\varepsilon))$  iteration complexity.

**Theorem 4.3** (Global convergence without knowing  $f(x^*)$ , informal). *Instate the same assumptions as **Theorem 4.1** and suppose  $z < f(x^*)$  is known. There exists a variant of OSGM-R that finds an  $\varepsilon$ -optimal point in  $\mathcal{O}(\kappa^* \log^2(1/\varepsilon))$  asymptotic complexity.*

The analysis for **Algorithm 6** is more involved, and we leave it to the appendix.

**Convex quadratics.** For strongly convex quadratics  $f(x) = \frac{1}{2}\langle x, Ax \rangle - \langle b, x \rangle$ ,  $P_r^* = A^{-1}$  gives  $r_x(A^{-1}) = 0$  for all  $x$ . This implies the following superlinear convergence guarantee.

**Theorem 4.4** (Superlinear convergence on quadratics). *For strongly convex quadratics with  $\nabla^2 f(x) \equiv A \succ 0$ , OSGM-R with  $\mathcal{P} = \mathbb{R}^{n \times n}$  and  $\eta = \frac{1}{4L^2}$  satisfies  $f(x^{K+1}) - f(x^*) \leq (f(x^1) - f(x^*)) \left(\frac{4L^2 \|P_1 - A^{-1}\|_F^2}{K}\right)^K$ .*

## 5 Gradient norm surrogate

The second surrogate, *gradient norm surrogate*, is defined as follows:

$$g_x(P) := \frac{\|\nabla f(x^+)\|}{\|\nabla f(x)\|} = \frac{\|\nabla f(x - P\nabla f(x))\|}{\|\nabla f(x)\|}. \quad (21)$$

Similar to the ratio surrogate, the gradient norm surrogate is defined with respect to the contraction of the gradient norm. We assume strong convexity ( $\mu > 0$ ) throughout this section. The gradient norm surrogate  $g_x$  can be evaluated without knowing  $f(x^*)$ . However, as we will discuss later in the section, it is more challenging to establish the learnability of  $g_x$ , which requires the following extra assumption:

**A4:**  $f(x)$  has  $H$ -Lipschitz Hessian.  $\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq H\|x - y\|$  for all  $x, y$ .

In this section, we assume a nonempty monotone oracle with respect to gradient norm  $\mathcal{M}_{\|\nabla f(x)\|, P} \neq \emptyset$ .

### 5.1 Surrogate loss

Substituting  $\varphi(x)$  in **Theorem 3.1** with  $\varphi(x) = \|\nabla f(x)\|$  and applying the definition of the gradient norm surrogate  $g_x$ , we obtain the following lemma.

**Lemma 5.1** (Surrogate loss and measure). *For all  $K \geq 1$ , the online scaled gradient method with nonempty monotone oracle  $\mathcal{M}_{\|\nabla f(x)\|, P} \neq \emptyset$  satisfies*

$$\|\nabla f(x^{K+1})\| \leq \|\nabla f(x^1)\| \left(\frac{1}{K} \sum_{k=1}^K g_{x^k}(P_k)\right)^K.$$

Although the gradient norm surrogate  $g_x$  can be nonconvex, **Proposition 6.1** shows that  $g_x$  can be approximated by an  $L$ -Lipschitz continuous convex function.

**Proposition 5.1** (Properties of  $g_x$ ). *Under **A1** to **A4**, for any fixed  $x \notin \mathcal{X}^*$ , the surrogate loss  $g_x(P)$  defined in (21) is  $L$ -Lipschitz continuous as a function in  $P$  and*

$$|g_x(P) - \hat{g}_x(P)| \leq H \|\nabla f(x)\| \|P\|^2,$$

where  $\hat{g}_x(P) = \left\| \frac{\nabla f(x)}{\|\nabla f(x)\|} - \nabla^2 f(x) P \frac{\nabla f(x)}{\|\nabla f(x)\|} \right\|$  is convex and  $L$ -Lipschitz continuous. In particular,

$$g_x(P_1) - g_x(P_2) - \langle g'_x(P_2), P_1 - P_2 \rangle \geq -HD^2 \|\nabla f(x)\|.$$

In addition, if  $x - P\nabla f(x) \notin \mathcal{X}^*$ , the loss  $g_x(P)$  is differentiable at  $P$  and its derivative takes the form

$$\nabla g_x(P) = -\frac{\nabla^2 f(x - P\nabla f(x)) \nabla f(x - P\nabla f(x)) \nabla f(x)^\top}{\|\nabla f(x)\| \cdot \|\nabla f(x - P\nabla f(x))\|}. \quad (22)$$

**Proposition 5.1** bounds the nonconvexity of  $g_x$  by  $\|\nabla f(x)\|$ , the non-stationarity at  $x$ . We can still apply online learning algorithms to  $g_x$  using these properties.

## 5.2 Online learning algorithm

Given Lipschitz loss functions whose nonconvexity can be bounded, online subgradient method gives the following regret guarantee.

**Lemma 5.2** (Learnability). *Given **A1** to **A3** and the gradient norm surrogate losses  $\{g_{x^k}\}$ , online subgradient method*

$$P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta g'_{x^k}(P_k)] \quad (23)$$

with stepsize  $\eta = c/\sqrt{K}$  generates a sequence of scaling matrices  $\{P_k\}_{k \geq 2}$  such that

$$\sum_{k=1}^K g_{x^k}(P_k) - \min_{P \in \mathcal{P}} \sum_{k=1}^K g_{x^k}(P) \leq \left(\frac{2D^2}{c} + \frac{cL^2}{2}\right)\sqrt{K} + \frac{HD^2}{2} \|\nabla f(x^1)\| K.$$

In particular, optimizing the constant  $c$  suggests the stepsize  $\eta = \frac{2D}{L\sqrt{K}}$  and the regret bound:

$$\sum_{k=1}^K g_{x^k}(P_k) - \min_{P \in \mathcal{P}} \sum_{k=1}^K g_{x^k}(P) \leq \rho_K := 2LD\sqrt{K} + \frac{HD^2}{2} \|\nabla f(x^1)\| K. \quad (24)$$

## 5.3 Algorithm design and analysis

We now state a realization of OSGM with the gradient norm surrogate loss  $g_x$ , denoted by OSGM-G. We choose the optimality measure  $\varphi$ , the surrogate loss  $\ell$ , and the online learning algorithm  $\mathcal{A}$  to be

$$\varphi(x) := \|\nabla f(x)\|, \quad \ell_x(P) := g_x(P), \quad \mathcal{A} := \text{online subgradient method in (23)},$$

and the monotone oracle  $\mathcal{M}$  is necessary. **Algorithm 3** presents the pseudocode for OSGM-G.

---

**Algorithm 3:** Online scaled gradient method with gradient norm surrogate (OSGM-G)

---

**input**  $x^1, P_1 \in \mathcal{P}$ , online gradient stepsize  $\eta > 0$ , nonempty  $\mathcal{M}_{\|\nabla f(x)\|, P}$   
**for**  $k = 1, 2, \dots$  **do**  
     $x^{k+1} = \mathcal{M}_{\|\nabla f(x^k)\|, P_k}(x^k)$   
     $P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta g'_{x^k}(P_k)]$   
**end**  
**output**  $x^{\text{best}}$  with minimum objective value

---

Combining **Lemma 5.1** and **Lemma 5.2**, **Theorem 5.1** characterizes the trajectory-based convergence behavior of OSGM-G.

**Theorem 5.1** (Trajectory-based convergence). *Under **A1** to **A4**, **Algorithm 3** (OSGM-G) with  $\eta = \frac{2D}{L\sqrt{K}}$  satisfies*

$$\|\nabla f(x^{K+1})\| \leq \|\nabla f(x^1)\|(\theta_K^* + \frac{\rho_K}{K})^K,$$

where  $\theta_K^* := \min_{P \in \mathcal{P}} \frac{1}{K} \sum_{k=1}^K g_{x^k}(P)$  and  $\rho_K = 2LD\sqrt{K} + \frac{HD^2}{2}\|\nabla f(x^1)\|K$  is defined in (24).

**Theorem 5.1** itself does not necessarily yield convergence. The regret  $\rho_K$  contains  $\frac{HD^2}{2}\|\nabla f(x^1)\|K$  and thus is linear in  $K$ . One solution is to start the algorithm at a near-stationary point with sufficiently small  $\|\nabla f(x^1)\|$ . This strategy leads to a two-stage algorithm, and our main result is based on this strategy for brevity of exposition.

*Remark 3.* For convex quadratics, the Lipschitz constant for Hessian is zero (i.e.,  $H = 0$ ) and the convergence of OSGM-G follows immediately.

### 5.3.1 Hindsight and global convergence

Define  $P_g^*$  to be the gradient norm scaling matrix that solves

$$\omega^* := \min_{P \in \mathcal{P}} \max_x \|I - \nabla^2 f(x)P\|.$$

The definition is motivated by

$$\begin{aligned} \|\nabla f(x - P\nabla f(x))\| &= \|\nabla f(x) - \int_0^1 \nabla^2 f(x - tP\nabla f(x))P\nabla f(x)dt\| \\ &= \|\int_0^1 [I - \nabla^2 f(x - tP\nabla f(x))P]\nabla f(x)dt\| \\ &\leq [\int_0^1 \|I - \nabla^2 f(x - tP\nabla f(x))P\|dt] \cdot \|\nabla f(x)\| \end{aligned}$$

and a contraction is established if  $\|I - \nabla^2 f(x)P\| < 1$  for all  $x$ . Define the quantity  $\lambda^* := \frac{1}{1-\omega^*}$ . Then the following facts follow:

**Lemma 5.3** (Hindsight). *Under **A1** to **A3**, the followings hold:*

- *Contraction.*  $\|\nabla f(x - P_g^*\nabla f(x))\| \leq (1 - \frac{1}{\lambda^*})\|\nabla f(x)\|$  for all  $x$ .
- *Conditioning.*  $\lambda^* \leq \frac{L}{\mu} = \kappa$ .
- *Surrogate loss bound.*  $g_x(P_g^*) \leq 1 - \frac{1}{\lambda^*}$  for all  $x \notin \mathcal{X}^*$ .

**Corollary 5.1** (Global convergence). *Under the same assumptions as **Theorem 5.1**,  $\theta_K^* \leq 1 - \frac{1}{\lambda^*}$  and with  $\|\nabla f(x^1)\| \leq \frac{1}{HD^2\lambda^*}$ , the asymptotic complexity of OSGM-G to find an  $\varepsilon$ -critical point is  $\mathcal{O}(2\lambda^* \log(1/\varepsilon))$ .*

*Remark 4.* It is possible to sharpen **Corollary 5.1** using a more fine-grained analysis: the nonconvexity will vanish as the algorithm converges, and  $\mathcal{O}(\lambda^* \log(1/\varepsilon))$  complexity still holds asymptotically.

*Remark 5.* OSGM-G can also output an  $\varepsilon$ -optimal solution due to the relation  $f(x) - f(x^*) \leq \frac{1}{2\mu}\|\nabla f(x)\|^2$  from strong convexity.

## 6 Hypergradient surrogate

The last surrogate loss, *hypergradient surrogate*, is defined as follows:

$$h_x(P) := \frac{f(x^+) - f(x)}{\|\nabla f(x)\|^2} = \frac{f(x - P\nabla f(x)) - f(x)}{\|\nabla f(x)\|^2}. \quad (25)$$

The name hypergradient comes from [2], by which the hypergradient descent heuristic improves the convergence of gradient-based methods. Unlike the ratio surrogate  $r_x$  or the gradient norm surrogate  $g_x$ , the hypergradient

surrogate  $h_x$  itself is not directly derived from  $\varphi(x)$  using a telescopic product, but instead motivated by the descent lemma:

$$f\left(x - \frac{1}{L}\nabla f(x)\right) - f(x) \leq -\frac{1}{2L}\|\nabla f(x)\|^2.$$

Dividing both sides of the inequality by  $\|\nabla f(x)\|^2$  for  $x \notin \mathcal{X}^*$  gives  $h_x$ . The descent lemma does not depend on the strong convexity coefficient  $\mu$ , so the hypergradient surrogate  $h_x$  applies to general convex (non-strongly convex) optimization problems. Throughout this section, we assume a nonempty monotone oracle with respect to function value gap  $\mathcal{M}_{f(x)-f(x^*),P} \neq \emptyset$ .

## 6.1 Surrogate loss

To analyze the hypergradient surrogate  $h_x$ , we must connect it with a measure of convergence. **Lemma 6.1** presents this relation.

**Lemma 6.1** (Surrogate loss and measure). *Under **A1**, **A2**, for all  $K \geq 1$ , the online scaled gradient method with nonempty monotone oracle  $\mathcal{M}_{f(x)-f(x^*),P} \neq \emptyset$  satisfies:*

- If  $\mu > 0$ , then

$$f(x^{K+1}) - f(x^*) \leq (f(x^1) - f(x^*)) \left(1 - 2\mu \max\left\{\frac{1}{K} \sum_{k=1}^K -h_{x^k}(P_k), 0\right\}\right)^K. \quad (26)$$

- If  $\mu \geq 0$ , then

$$\min_{1 \leq k \leq K} \|\nabla f(x^k)\|^2 \leq \frac{f(x^1) - f(x^*)}{K} \cdot \frac{1}{\max\left\{\frac{1}{K} \sum_{k=1}^K -h_{x^k}(P_k), 0\right\}}, \quad (27)$$

$$f(x^{K+1}) - f(x^*) \leq \frac{\Delta^2}{K} \cdot \frac{1}{\max\left\{\frac{1}{K} \sum_{k=1}^K -h_{x^k}(P_k), 0\right\}}, \quad (28)$$

where  $\Delta = \max_{x \in \mathcal{L}_{f(x^1)}} \min_{x^* \in \mathcal{X}^*} \|x - x^*\|$ .

*Remark 6.* The  $\max\{\cdot, 0\}$  terms arise from the monotone oracle. Here, we slightly abuse the notation: if the denominator in (27) or (28) is 0, the bound simplifies to a trivial bound, with the right-hand side being infinity. Note that the surrogate loss appears in the denominator of the sublinear convergence rate, which differs from the previous analyses.

Now we establish the properties of the hypergradient surrogate  $h_x$ .

**Proposition 6.1** (Properties of  $h_x$ ). *Under **A1** to **A3**, for any fixed  $x \notin \mathcal{X}^*$ , the surrogate loss  $h_x(P)$  defined in (25) is convex and  $(LD+1)$ -Lipschitz continuous as a function in  $P$ . In addition, the derivative of  $h_x$  takes the form*

$$\nabla h_x(P) = -\frac{\nabla f(x - P\nabla f(x))\nabla f(x)^\top}{\|\nabla f(x)\|^2}. \quad (29)$$

## 6.2 Online learning algorithm

Given convex and Lipschitz-continuous losses, online gradient descent gives the following regret guarantee.

**Lemma 6.2** (Learnability). *Given **A1** to **A3** and the hypergradient surrogate losses  $\{h_{x^k}\}$ , online gradient descent*

$$P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta\nabla h_{x^k}(P_k)] \quad (30)$$

with stepsize  $\eta = c/\sqrt{K}$  generates a sequence of scaling matrices  $\{P_k\}_{k \geq 2}$  such that

$$\sum_{k=1}^K h_{x^k}(P_k) - \min_{P \in \mathcal{P}} \sum_{k=1}^K h_{x^k}(P) \leq \left(\frac{2D^2}{c} + \frac{c(LD+1)^2}{2}\right)\sqrt{K}.$$

In particular, optimizing the constant  $c$  suggests the stepsize  $\eta = \frac{2D}{(LD+1)\sqrt{K}}$  and the regret bound:

$$\sum_{k=1}^K h_{x^k}(P_k) - \min_{P \in \mathcal{P}} \sum_{k=1}^K h_{x^k}(P) \leq \rho_K := 2D(LD+1)\sqrt{K}. \quad (31)$$

### 6.3 Algorithm design and analysis

We now state a realization of OSGM with the hypergradient surrogate loss  $h_x$ , denoted by OSGM-H. We choose the optimality measure  $\varphi$ , the surrogate loss  $\ell$ , and the online learning algorithm  $\mathcal{A}$  to be

$$\varphi(x) := f(x) - f(x^*) \text{ or } \|\nabla f(x)\|, \quad \ell_x(P) := h_x(P), \quad \mathcal{A} := \text{online gradient descent in (30),}$$

and the monotone oracle  $\mathcal{M}$  is necessary. **Algorithm 4** presents the pseudocode for OSGM-H.

---

**Algorithm 4:** Online scaled gradient method with hypergradient surrogate (OSGM-H)

---

**input**  $x^1, P_1 \in \mathcal{P}$ , online gradient stepsize  $\eta > 0$ , nonempty oracle  $\mathcal{M}_{f(x)-f(x^*),P}$   
**for**  $k = 1, 2, \dots$  **do**  
     $x^{k+1} = \mathcal{M}_{f(x)-f(x^*),P_k}(x^k)$   
     $P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta \nabla h_{x^k}(P_k)]$   
**end**  
**output**  $x^{\text{best}}$  with minimum objective value

---

Combining **Lemma 6.1** and **Lemma 6.2**, **Theorem 6.1** characterizes the trajectory-based convergence behavior of OSGM-H.

**Theorem 6.1** (Trajectory-based convergence). *Under **A1** to **A3**, **Algorithm 4** (OSGM-H) with  $\eta = \frac{2D}{(LD+1)\sqrt{K}}$  satisfies*

• If  $\mu > 0$ , then

$$f(x^{K+1}) - f(x^*) \leq (f(x^1) - f(x^*)) (1 - 2\mu \max\{-\theta_K^* - \frac{\rho_K}{K}, 0\})^K.$$

• If  $\mu \geq 0$ , then

$$\begin{aligned} \min_{1 \leq k \leq K} \|\nabla f(x^k)\|^2 &\leq \frac{f(x^1) - f(x^*)}{K} \cdot \frac{1}{\max\{-\theta_K^* - \frac{\rho_K}{K}, 0\}}, \\ f(x^{K+1}) - f(x^*) &\leq \frac{\Delta^2}{K} \cdot \frac{1}{\max\{-\theta_K^* - \frac{\rho_K}{K}, 0\}}, \end{aligned}$$

where  $\theta_K^* := \min_{P \in \mathcal{P}} \frac{1}{K} \sum_{k=1}^K h_{x^k}(P)$ ,  $\Delta$  is defined in **Lemma 6.1** and  $\rho_K = 2D(LD+1)\sqrt{K}$  is defined in (31).

#### 6.3.1 Hindsight and global convergence

Define  $P_h^*$  to be the hypergradient scaling matrix that solves

$$\gamma^* := \max_{P \in \mathcal{P}} \min_{x \in \mathcal{L}_{f(x^1)} \setminus \mathcal{X}^*} -h_x(P) = \frac{f(x) - f(x - P \nabla f(x))}{\|\nabla f(x)\|^2}.$$

Intuitively,  $\gamma^*$  maximizes the function value progress with respect to the gradient norm and can be interpreted as the inverse of the local Lipschitz smoothness constant. The descent lemma gives a lower bound on  $\gamma^*$ .

**Lemma 6.3** (Hindsight). *Under **A1** to **A3**,  $-h_x(P_h^*) \geq \gamma^* \geq \frac{1}{2L}$  for all  $x \notin \mathcal{X}^*$ .*

**Corollary 6.1** (Global convergence). *Under the same assumptions as **Theorem 6.1**,  $\theta_K^* \leq -\gamma^*$ , and*

- If  $\mu > 0$ , then the asymptotic complexity of OSGM-H to find an  $\varepsilon$ -optimal point is  $\mathcal{O}(\frac{1}{2\mu\gamma^*} \log(1/\varepsilon))$ .
- If  $\mu = 0$ , then the asymptotic complexity of OSGM-H to find an  $\varepsilon$ -optimal point is  $\mathcal{O}(\frac{1}{\gamma^*\varepsilon})$ .

*Remark 7.* Given  $-\theta_K^* \geq \frac{1}{2L}$ , the complexity of OSGM-H is no worse than vanilla gradient descent and can provide acceleration if  $-\theta_K^* > \frac{1}{2L}$ . Our results for the first time show that the hypergradient heuristic, when combined with a monotone oracle  $\mathcal{M}$ , provably accelerates gradient descent.

## 7 Practical considerations

This section considers practical aspects of implementing OSGM.

### 7.1 Gradient evaluations for $r_x$ and $h_x$ using simple comparison oracle

A first look at **Algorithm 2** and **Algorithm 4** suggests an additional gradient evaluation at every iteration. Both  $\nabla r_x$  and  $\nabla h_x$  need to evaluate two gradients in every iteration. However, with  $x^{k+1/2} := x^k - P_k \nabla f(x^k)$  and simple comparison (**Section 2.3**) as the monotone oracle, the gradient  $\nabla r_{x^k}(P)$  in **Algorithm 2** can be expressed as

$$\nabla r_{x^k}(P) = -\frac{\nabla f(x^{k+1/2}) \nabla f(x^k)^\top}{f(x^k) - f(x^*)}. \quad (32)$$

If  $x^{k+1} = x^{k+1/2}$ , then  $\nabla f(x^{k+1/2})$  can be reused in the next iteration; if  $x^{k+1} = x^k$ , then  $\nabla f(x^k)$  can be reused. Therefore, simple comparison oracle ensures that the total number of gradient evaluations in OSGM-R is the same as that of gradient descent but simply requires an additional cache to store  $\nabla f(x^{k+1})$ . Regarding  $\nabla g_x$ , its implementation needs a Hessian-gradient product, which can be efficiently computed in practice.

### 7.2 Efficient scaling matrix updates

The subset  $\mathcal{P}$  can be chosen to have a simple structure, such as diagonal matrices or sparse matrices with some prespecified sparsity pattern. Then the scaling matrix  $P_{k+1}$  can be efficiently updated in the cost of  $\mathcal{O}(\text{supp}(\mathcal{P}))$  since it suffices to compute the non-zero entries of the sparsity pattern in  $\mathcal{P}$ . For example, if  $\mathcal{P}$  is the set of diagonal matrices, then (32) simplifies to  $\nabla r_{x^k}(P) = -\frac{\nabla f(x^k - P_k \nabla f(x^k)) \odot \nabla f(x^k)}{f(x^k) - f(x^*)}$ , where  $\odot$  denotes the element-wise product. A simpler structure in  $\mathcal{P}$  provides more efficient scaling matrix updates. However, more freedom in  $\mathcal{P}$  may provide smaller  $\theta_K^*$ , enhancing the convergence of OSGM.

### 7.3 Choice of candidate set of scaling matrices $\mathcal{P}$

We propose two heuristics for choosing a subset  $\mathcal{P}$  of sparse matrices. Sparsity refers to either entries sparsity or spectrum sparsity. We assume that some Hessian matrix  $\nabla^2 f(x) = A \succ 0$  is known.

- *Nonzero sparsity pattern.*

A preconditioner can be viewed as a cutting plane in the difference of extremal spectrum [17]. Let  $v_{\min}$  and  $v_{\max}$  be two extremal eigenvectors of  $A$ . Then  $|v_{\max} v_{\max}^\top - v_{\min} v_{\min}^\top|$ , an  $n \times n$  grid with nonnegative entries, serves as a score function for the most critical sparsity pattern. The large-magnitude entries in  $|v_{\max} v_{\max}^\top - v_{\min} v_{\min}^\top|$  strongly affect the conditioning of the system.

- *Spectral sparsity (low rank).*

It is common to consider diagonal plus low-rank preconditioners, and randomized preconditioners have proved to be very efficient [16]. Given a low-rank matrix  $M$ , we can parameterize  $\mathcal{P} = \{\text{diag}(d) + \alpha M : (d, \alpha) \in \mathbb{R}^{n+1}\}$  to be the linear combination between diagonal matrices and  $M$ .

### 7.4 Choice of online learning algorithm $\mathcal{A}$

Our convergence analyses show that a good online learning algorithm  $\mathcal{A}$  benefits convergence of OSGM. For simplicity, the simplest possible online learning algorithms are adopted in the theoretical analysis. However, OSGM is compatible with more advanced online learning algorithms such as **AdaGrad**. Using advanced online algorithms often greatly improves the robustness and practical performance of OSGM. In particular, our results on the hypergradient surrogate loss provide new insights into improving the hypergradient descent heuristics.

## 8 Numerical experiments

In this section, we conduct experiments to show the performance of online scaled gradient methods. We test on standard strongly convex optimization problems, including least squares and regularized logistic regression.

### 8.1 Experiment setup

**Synthetic data.** For the least squares problem  $f(x) = \frac{1}{2}\|Ax - b\|^2$ ,  $A \in \mathbb{R}^{n \times n} = CDC^T + \sigma I$  with  $C$  is element-wise generated from  $0.01 \times \mathcal{N}(0, 1)$  and an identity matrix  $I$  is added to it;  $D$  is a diagonal matrix with  $\mathcal{U}[0, 1]^n$  diagonals;  $b \in \mathbb{R}^m$  is generated from  $\mathcal{U}[0, 1]^n$ .

**Real data.** We use datasets from LIBSVM [4] for support vector machine (SVM) problem  $f(x) = \frac{1}{m} \sum_{i=1}^m f_i(x) + \frac{\lambda}{2}\|x\|^2$ , where  $f_i$  is the squared hinge loss [24]. We set  $\lambda = 5/n$ .

**Benchmark algorithms.** Eight algorithms are compared:

- (GD) Gradient descent with  $1/L$  stepsize.
- (OptDiagGD) Gradient descent with the universal optimal diagonal preconditioner [35, 17].
- (OSGM-R) Online scaled gradient method with ratio surrogate.
- (OSGM-G) Online scaled gradient method with gradient norm surrogate.
- (OSGM-H) Online scaled gradient method with hypergradient surrogate.
- (AdaGrad) Adaptive (sub)gradient method [12].
- (AGD) Accelerated gradient descent for general convex problems [8].
- (SAGD) Accelerated gradient descent for strongly convex problems [8].

OptDiagGD and OSGM-G are only tested on problems with fixed Hessian.

**Algorithm configurations.** We configure the algorithms as follows.

- 1) *Dataset generation.* For synthetic data, we pick  $n = 100$  and  $\sigma \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ .
- 2) *Initial point.* Initial points  $x^1$  for all the algorithms are generated from standard normal  $\mathcal{N}(0, I_n)$  and scaled to have unit length. Initial scaling matrix  $P_0 = 0$ .
- 3) *Maximum iteration.* The maximum iteration is set to  $K = 10000$ .
- 4) *Stopping criterion.* Algorithm stops when  $\|\nabla f(x^k)\| \leq 10^{-10}$ .
- 5) *Stepsize configuration.* (AdaGrad) uses the optimal stepsize among  $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ .
- 6) *Monotone oracle.* All OSGM methods use simple comparison (**Section 2.3**) as the monotone oracle.
- 7) *Online learning algorithm.* All OSGM methods use  $\mathcal{A} = (\text{AdaGrad})$  with the optimal stepsize among  $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ .
- 8) *Choice of candidate scaling matrix  $\mathcal{P}$ .* We choose  $\mathcal{P} = \mathbb{R}^{n \times n}$  in **Section 8.3** and  $\mathcal{P}$  as the set of diagonal matrices in the rest of the experiments.
- 9) *Knowledge of optimal value.* When the exact optimal value is unknown, OSGM-R uses the auxiliary surrogate  $r_x^z$  in (20). But we allow  $z$  to be an arbitrary guess of  $f(x^*)$ , and if  $z > f(x^k)$ , we heuristically adjust  $z \leftarrow f(x^k) - \min\{5(z - f(x^k)), 1\}$  to update the lower bound. This strategy is not theoretically justified but performs well in practice.

## 8.2 Toy example: near diagonal convex quadratic

This section verifies the convergence behavior of OSGM on a toy least squares problem with near diagonal Hessian. The problem has  $\kappa \approx 68$  and  $\kappa^* \approx 4.7 < \sqrt{\kappa}$ . Theory predicts that OSGM should outperform SAGD asymptotically. **Figure 1** (left) illustrates the performance of the eight tested algorithms, with OSGM-R and OSGM-H showing the most competitive performance. In particular, the linear convergence rates (slope) of three OSGM algorithms are better than that of SAGD, which aligns with our theory. Moreover, OSGM-R and OSGM-H both converge faster than gradient descent using the universal diagonal preconditioner (OptDiagGD). This is also consistent with our theory and suggests that we can still gain from being adaptive, even on a convex quadratic with fixed curvature. Notably, AdaGrad also achieves competitive performance on this problem.

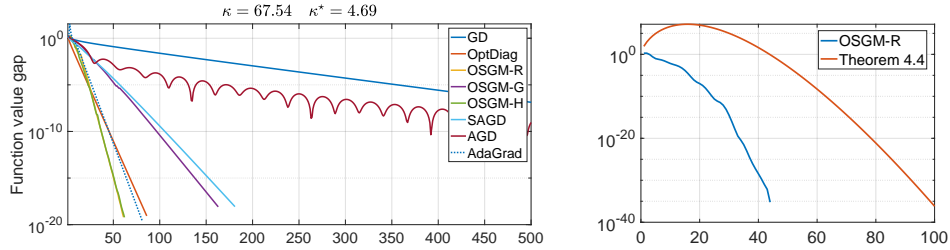


Figure 1: Left: comparison of benchmark algorithms on toy quadratic problem. Right: superlinear convergence of OSGM-R on convex quadratics. x-axis: iteration count.

## 8.3 Superlinear convergence on quadratics

This section verifies the **Theorem 4.4**, the superlinear convergence behavior of OSGM-R. We take  $\mathcal{P} = \mathbb{R}^{n \times n}$  and plot the theoretical bound in **Theorem 4.4** as well as the true performance of OSGM-R. **Figure 1** (right) indicates that OSGM-R exhibits superlinear convergence and validates our theory.

## 8.4 More comparison between the algorithms

This section compares different algorithms on the aforementioned datasets. **Figure 2** shows the results on synthetic least squares problems with different condition numbers.

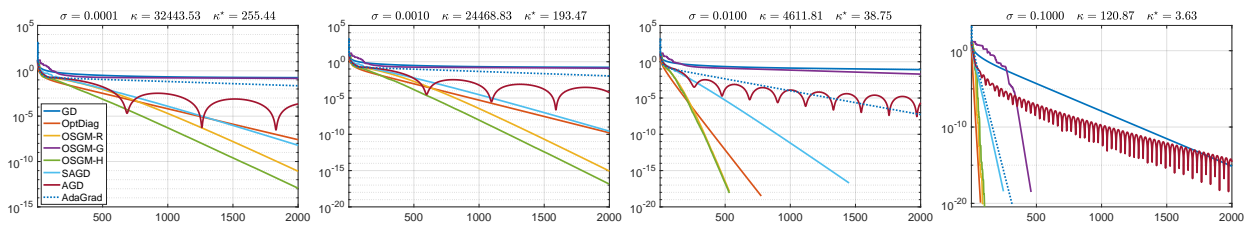


Figure 2: Function value gap on least squares problem with  $\sigma \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$

**Figure 2** suggests that when  $\kappa^* \ll \sqrt{\kappa}$ , OSGM tends to outperform accelerated methods. On the other hand, if  $\kappa^* > \sqrt{\kappa}$ , OSGM is often less competitive compared to SAGD on quadratics.

**Figure 3** shows the results on the SVM problems from LIBSVM, and we observe similar competitive performance of OSGM-R and OSGM-H.



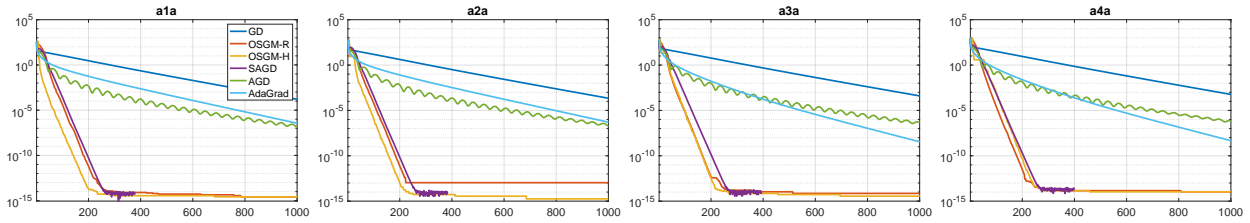


Figure 3: Function value gap on the support vector machine problems

## 9 Conclusions and future directions

In this paper, we discuss OSGM, a general framework that allows online convex optimization algorithms to provably accelerate gradient-based algorithms. Our framework achieves a strong trajectory-based convergence guarantee and explains the success of the popular hypergradient descent heuristic. Future directions include extending the results to accelerated gradient descent, stochastic gradient descent, nonconvex, nonsmooth, and constrained optimization, and to other iterative algorithms where a scaled update affects the algorithm performance.

## Acknowledgement

We appreciate the constructive feedback from Professor Qi Deng from Shanghai Jiaotong University and Zikai Xiong from MIT.

## References

- [1] David Applegate, Mateo Diaz, Oliver Hinder, Haihao Lu, Miles Lubin, Brendan O’Donoghue, and Warren Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. *Advances in Neural Information Processing Systems*, 34:20243–20257, 2021.
- [2] Atilim Gunes Baydin, Robert Cornish, David Martinez Rubio, Mark Schmidt, and Frank Wood. Online learning rate adaptation with hypergradient descent. In *International Conference on Learning Representations*, 2018.
- [3] Kartik Chandra, Audrey Xie, Jonathan Ragan-Kelley, and Erik Meijer. Gradient descent: The ultimate optimizer. *Advances in Neural Information Processing Systems*, 35:8214–8225, 2022.
- [4] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.
- [5] Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research*, 23(189):1–59, 2022.
- [6] Xinyi Chen and Elad Hazan. Online control for meta-optimization. *Advances in Neural Information Processing Systems*, 36, 2024.
- [7] Rudrajit Das, Naman Agarwal, Sujay Sanghavi, and Inderjit S Dhillon. Towards quantifying the preconditioning effect of adam. *arXiv preprint arXiv:2402.07114*, 2024.
- [8] Alexandre d’Aspremont, Damien Scieur, Adrien Taylor, et al. Acceleration methods. *Foundations and Trends® in Optimization*, 5(1-2):1–245, 2021.

- [9] Aaron Defazio, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, Ashok Cutkosky, et al. The road less scheduled. *arXiv preprint arXiv:2405.15682*, 2024.
- [10] Qi Deng, Qing Feng, Wenzhi Gao, Dongdong Ge, Bo Jiang, Yuntian Jiang, Jingsong Liu, Tianhao Liu, Chenyu Xue, Yinyu Ye, et al. An enhanced alternating direction method of multipliers-based interior point method for linear and conic optimization. *INFORMS Journal on Computing*, 2024.
- [11] X Doan and Henry Wolkowicz. Numerical computations and the  $\omega$ -condition number. Technical report, Citeseer, 2011.
- [12] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.
- [13] Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *International conference on machine learning*, pages 1920–1930. PMLR, 2019.
- [14] Zachary Frangella, Pratik Rathore, Shipu Zhao, and Madeleine Udell. Sketchysgd: Reliable stochastic optimization via robust curvature estimates. *arXiv preprint arXiv:2211.08597*, 2022.
- [15] Zachary Frangella, Pratik Rathore, Shipu Zhao, and Madeleine Udell. Promise: Preconditioned stochastic optimization methods by incorporating scalable curvature estimates. *arXiv preprint arXiv:2309.02014*, 2023.
- [16] Zachary Frangella, Joel A Tropp, and Madeleine Udell. Randomized nyström preconditioning. *SIAM Journal on Matrix Analysis and Applications*, 44(2):718–752, 2023.
- [17] Wenzhi Gao, Zhaonan Qu, Madeleine Udell, and Yinyu Ye. Scalable approximate optimal diagonal preconditioning. *arXiv preprint arXiv:2312.15594*, 2023.
- [18] Guillaume Garrigos and Robert M Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.
- [19] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.
- [20] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021.
- [21] Arun Jambulapati, Jerry Li, Christopher Musco, Aaron Sidford, and Kevin Tian. Fast and near-optimal diagonal preconditioning. *arXiv preprint arXiv:2008.01722*, 2020.
- [22] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [23] Frederik Kunstner, Victor Sanches Portella, Mark Schmidt, and Nicholas Harvey. Searching for optimal per-coordinate step-sizes with multidimensional backtracking. *Advances in Neural Information Processing Systems*, 36, 2024.
- [24] Yuh-Jye Lee and Olvi L Mangasarian. Ssvm: A smooth support vector machine for classification. *Computational optimization and Applications*, 20:5–22, 2001.
- [25] Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.
- [26] Ke Li and Jitendra Malik. Learning to optimize. *arXiv preprint arXiv:1606.01885*, 2016.
- [27] Xi-Lin Li. Preconditioned stochastic gradient descent. *IEEE transactions on neural networks and learning systems*, 29(5):1454–1466, 2017.

- [28] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR, 2015.
- [29] Chris J Maddison, Daniel Paulin, Yee Whye Teh, and Arnaud Doucet. Dual space preconditioning for gradient descent. *SIAM Journal on Optimization*, 31(1):991–1016, 2021.
- [30] H Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. *arXiv preprint arXiv:1002.4908*, 2010.
- [31] Ion Necoara, Yu Nesterov, and Francois Glineur. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, 175:69–107, 2019.
- [32] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.
- [33] Brendan O’donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169:1042–1068, 2016.
- [34] Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.
- [35] Zhaonan Qu, Wenzhi Gao, Oliver Hinder, Yinyu Ye, and Zhengyuan Zhou. Optimal diagonal preconditioning. *Operations Research*, 2024.
- [36] David Martinez Rubio. Convergence analysis of an adaptive method of gradient descent. *University of Oxford, Oxford, M. Sc. thesis*, 2017.
- [37] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

# Appendix

## Table of Contents

---

<b>A Proof of results in Section 3</b>	<b>21</b>
A.1 Proof of Theorem 3.1 . . . . .	21
<b>B Proof of results in Section 4</b>	<b>21</b>
B.1 Auxiliary results . . . . .	21
B.2 Proof of Lemma 4.1 . . . . .	22
B.3 Proof of Proposition 4.1 . . . . .	22
B.4 Proof of Lemma 4.2 . . . . .	22
B.5 Proof of Theorem 4.1 . . . . .	23
B.6 Proof of Lemma 4.3 . . . . .	23
B.7 Proof of Corollary 4.1 . . . . .	23
B.8 Proof of Theorem 4.2 . . . . .	23
B.9 Proof of Proposition 4.2 . . . . .	23
B.10 Proof of Theorem 4.4 . . . . .	24
<b>C Function value ratio surrogate with optimal value lower bound</b>	<b>24</b>
C.1 Surrogate loss . . . . .	25
C.2 Online learning algorithm . . . . .	25
C.3 Algorithm design and analysis . . . . .	25
C.4 Proof of Lemma C.1 . . . . .	26
C.5 Proof of Proposition C.1 . . . . .	26
C.6 Proof of Lemma C.2 . . . . .	26
C.7 Proof of Theorem C.1 . . . . .	26
C.8 Proof of Lemma C.3 . . . . .	27
C.9 Proof of Theorem C.2 . . . . .	28
<b>D Proof of results in Section 5</b>	<b>28</b>
D.1 Proof Lemma 5.1 . . . . .	28
D.2 Proof of Proposition 5.1 . . . . .	29
D.3 Proof of Lemma 5.2 . . . . .	29
D.4 Proof of Theorem 5.1 . . . . .	30
D.5 Proof of Lemma 5.3 . . . . .	30
D.6 Proof of Corollary 5.1 . . . . .	31
<b>E Proof of results in Section 6</b>	<b>31</b>
E.1 Proof of Lemma 6.1 . . . . .	31
E.2 Proof of Proposition 6.1 . . . . .	33
E.3 Proof of Lemma 6.2 . . . . .	33
E.4 Proof of Theorem 6.1 . . . . .	33
E.5 Proof of Lemma 6.3 . . . . .	34
E.6 Proof of Corollary 6.1 . . . . .	34

---

## A Proof of results in Section 3

### A.1 Proof of Theorem 3.1

Since the measure  $\varphi$  is non-negative. Applying arithmetic-geometric mean inequality

$$\left(\prod_{k=1}^K a_k\right)^{1/K} \leq \frac{1}{K} \sum_{k=1}^K a_k$$

completes the proof.

## B Proof of results in Section 4

### B.1 Auxiliary results

**Lemma B.1** ([34]). *Let  $r(P)$  be a  $\tau$ -smooth function with  $\min_{P \in \mathcal{P}} r(P) \geq 0$ . Then  $r(P) \geq \frac{1}{2\tau} \|\nabla r(P)\|^2$  for all  $P \in \mathcal{P}$ .*

**Lemma B.2.** *Given a family of non-negative, convex, and  $\tau$ -smooth losses  $\{r_k\}$ , online gradient descent*

$$P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta \nabla r_k(P_k)] \quad (33)$$

with stepsize  $\eta \leq 1/(2\tau)$  generates a sequence of scaling matrices  $\{P_k\}_{k \geq 2}$  such that

$$\sum_{k=1}^K r_k(P_k) - \sum_{k=1}^K r_k(P) \leq \frac{1}{\eta} \|P - P_1\|_F^2 + 2\tau\eta \sum_{k=1}^K r_k(P) \quad \text{for any } P \in \mathcal{P}. \quad (34)$$

*Proof.* The proof follows the standard proof of the  $L^*$  regret bound [34] in online convex optimization but is tailored to our settings. For any  $P \in \mathcal{P}$ , we have

$$\begin{aligned} \|P_{k+1} - P\|_F^2 &= \|\Pi_{\mathcal{P}}[P_k - \eta \nabla r_k(P_k)] - P\|_F^2 \\ &\leq \|P_k - P - \eta \nabla r_k(P_k)\|_F^2 \end{aligned} \quad (35)$$

$$= \|P_k - P\|_F^2 - 2\eta \langle \nabla r_k(P_k), P_k - P \rangle + \eta^2 \|\nabla r_k(P_k)\|_F^2, \quad (36)$$

where (35) uses non-expansiveness of projection. With convexity  $r_k(P) - r_k(P_k) \geq \langle \nabla r_k(P_k), P - P_k \rangle$ ,

$$\|P_{k+1} - P\|_F^2 \leq \|P_k - P\|_F^2 + 2\eta(r_k(P) - r_k(P_k)) + \eta^2 \|\nabla r_k(P_k)\|_F^2.$$

Re-arrangement yields  $r_k(P_k) - r_k(P) \leq \frac{\eta}{2} \|\nabla r_k(P_k)\|_F^2 + \frac{1}{2\eta} [\|P_k - P\|_F^2 - \|P_{k+1} - P\|_F^2]$ . Telescoping over  $k$  and dropping the term  $-\frac{1}{2\eta} \|P_{K+1} - P\|_F^2$  to obtain

$$\sum_{k=1}^K r_k(P_k) - \sum_{k=1}^K r_k(P) \leq \frac{1}{2\eta} \|P_1 - P\|_F^2 + \frac{\eta}{2} \sum_{k=1}^K \|\nabla r_k(P_k)\|_F^2. \quad (37)$$

Using **Lemma B.1**, we have  $\|\nabla r_k(P_k)\|_F^2 \leq 2\tau r_k(P_k)$ . Plugging this bound into (37) gives

$$\sum_{k=1}^K r_k(P_k) - \sum_{k=1}^K r_k(P) \leq \frac{1}{2\eta} \|P_1 - P\|_F^2 + \tau\eta \sum_{k=1}^K r_k(P_k).$$

Re-arrangement gives

$$(1 - \tau\eta) \left[ \sum_{k=1}^K r_k(P_k) - \sum_{k=1}^K r_k(P) \right] \leq \frac{1}{2\eta} \|P_1 - P\|_F^2 + \tau\eta \sum_{k=1}^K r_k(P). \quad (38)$$

For  $\eta \leq \frac{1}{2\tau}$ , we may divide both sides of (38) by  $1 - \tau\eta$  and plug in the bound  $\frac{1}{1 - \tau\eta} \leq 2$  to obtain

$$\sum_{k=1}^K r_k(P_k) - \sum_{k=1}^K r_k(P) \leq \frac{1}{\eta} \|P - P_1\|_F^2 + 2\tau\eta \sum_{k=1}^K r_k(P), \quad (39)$$

and this completes the proof. □

## B.2 Proof of Lemma 4.1

Since  $f(x) - f(x^*) \geq 0$ , applying **Theorem 3.1** with  $\varphi(x) = f(x) - f(x^*)$  completes the proof.

## B.3 Proof of Proposition 4.1

Denote  $u_x(P) := f(x - P\nabla f(x))$ . As a function of  $P$ ,  $u_x(P) = f(x - P\nabla f(x))$  is a composition of convex function  $f(x)$  and affine transformation  $x - P\nabla f(x)$ . Hence  $u_x$  is a convex function. The chain rule gives

$$\nabla u_x(P) = \nabla f(x - P\nabla f(x)) = \nabla f(x - P\nabla f(x))\nabla f(x)^\top.$$

For any  $P_1, P_2 \in \mathcal{P}$ , we can successively deduce that

$$\begin{aligned} \|\nabla u_x(P_1) - \nabla u_x(P_2)\|_F &= \|\nabla f(x - P_1\nabla f(x))\nabla f(x)^\top - \nabla f(x - P_2\nabla f(x))\nabla f(x)^\top\|_F \\ &= \|[\nabla f(x - P_1\nabla f(x)) - \nabla f(x - P_2\nabla f(x))]\nabla f(x)^\top\|_F \\ &\leq \|\nabla f(x - P_1\nabla f(x)) - \nabla f(x - P_2\nabla f(x))\| \cdot \|\nabla f(x)\| \\ &\leq L\|(P_1 - P_2)\nabla f(x)\| \cdot \|\nabla f(x)\| \\ &\leq L\|\nabla f(x)\|^2\|P_1 - P_2\| \\ &\leq L\|\nabla f(x)\|^2\|P_1 - P_2\|_F, \end{aligned} \tag{40}$$

where (40) uses the submultiplicativity of Frobenius norm  $\|AB\|_F \leq \|A\|_F\|B\|_F$ ; and (41) uses  $L$ -smoothness of  $f(x)$ . Hence  $u_x$  is  $L\|\nabla f(x)\|^2$ -smooth. Since the surrogate loss  $r_x(P) = \frac{u_x(P) - f(x^*)}{f(x) - f(x^*)}$  is a positive-scaled convex function  $u_x$  with translation, and hence  $r_x$  is also convex. Next, since  $x \notin \mathcal{X}^*$ , the denominator of  $r_x(P)$  must be positive, and hence  $r_x(P) \geq 0$  for all  $P$ . Lastly, since  $r_x(P) = \frac{u_x(P) - f(x^*)}{f(x) - f(x^*)}$  and  $u_x$  is  $L\|\nabla f(x)\|^2$ -smooth,  $r_x(P)$  is also smooth with smoothness constant no greater than  $2L^2$ :

$$\frac{L\|\nabla f(x)\|^2}{f(x) - f(x^*)} = \frac{L\|\nabla f(x) - \nabla f(x^*)\|^2}{f(x) - f(x^*)} \leq 2L^2,$$

where the last inequality invokes  $L$ -smoothness of  $f(x)$ . This completes the proof.

## B.4 Proof of Lemma 4.2

For simplicity we denote  $r_k(P) := r_{x^k}(P)$ . By **Proposition 4.1**, the surrogate losses  $\{r_k\}$  are  $2L^2$ -smooth and non-negative. Then using **Lemma B.2** with  $\tau = 2L^2$ , we get

$$\sum_{k=1}^K r_k(P_k) - \sum_{k=1}^K r_k(P) \leq \frac{1}{\eta}\|P - P_1\|_F^2 + 4L^2\eta \sum_{k=1}^K r_k(P), \tag{42}$$

which proves (49). Suppose further  $\text{diam}(\mathcal{P}) \leq D$ . Then

$$\begin{aligned} r_x(P) &= \frac{f(x - P\nabla f(x)) - f(x^*)}{f(x) - f(x^*)} \\ &\leq \frac{f(x) - f(x^*) - \langle \nabla f(x), (-P + \frac{1}{2}P^\top P)\nabla f(x) \rangle}{f(x) - f(x^*)} \\ &\leq 1 + \|\frac{1}{2}P^\top P - P\| \frac{\|\nabla f(x)\|^2}{f(x) - f(x^*)} \\ &\leq 1 + 2L(\frac{1}{2}D^2 + D) \\ &= (1 + LD)^2. \end{aligned}$$

Therefore, (42) implies: for  $\eta \leq \frac{1}{4L^2}$ ,

$$\sum_{k=1}^K r_k(P_k) - \sum_{k=1}^K r_k(P) \leq \frac{1}{\eta} D^2 + 4L^2(1+LD)^2 K \eta, \quad (43)$$

in which the right-hand side is minimized (as a function of  $\eta$ ) at  $\eta = \frac{D}{2L(1+LD)\sqrt{K}}$ . By taking the stepsize  $\eta = \min \left\{ \frac{1}{4L^2}, \frac{D}{2L(1+LD)\sqrt{K}} \right\}$  and then minimizing over  $P \in \mathcal{P}$ , we conclude

$$\sum_{k=1}^K r_{x^k}(P_k) \leq \min_{P \in \mathcal{P}} \sum_{k=1}^K r_{x^k}(P) + \max \{4LD(1+LD)\sqrt{K}, 8L^2 D^2\}.$$

and this completes the proof.

## B.5 Proof of Theorem 4.1

By **Lemma 4.2**, we have

$$\frac{1}{K} \sum_{k=1}^K r_{x^k}(P_k) \leq \frac{1}{K} \sum_{k=1}^K r_{x^k}(P) + \frac{\rho K}{K}$$

for all  $P \in \mathcal{P}$ . and plugging the relation into **Lemma 4.1** completes the proof.

## B.6 Proof of Lemma 4.3

For any fixed  $x \notin \mathcal{X}^*$ , the result  $r_x(P_r^*) \leq 1 - \frac{1}{\kappa^*}$  is a direct consequence of (16).

## B.7 Proof of Corollary 4.1

Using **Lemma 4.3** and **Theorem 4.1**,  $\theta_K^* \leq 1 - \frac{1}{\kappa^*}$  and plugging the bound back into **Theorem 4.1** completes the proof.

## B.8 Proof of Theorem 4.2

Combining **Lemma 4.1** and (49) from **Lemma 4.2** and using the relation  $r_x(P_r^*) \leq 1 - \frac{1}{\kappa^*}$  from **Lemma 4.3**, we have, for  $\eta \leq \frac{1}{4L^2}$ , that

$$\begin{aligned} f(x^{K+1}) - f(x^*) &\leq (f(x^1) - f(x^*)) \left( \frac{1}{K} \sum_{k=1}^K r_{x^k}(P_r^*) + \frac{1}{\eta} \|P_r^* - P_1\|_F^2 + 4L^2 \eta \sum_{k=1}^K r_{x^k}(P_r^*) \right)^K \\ &\leq (f(x^1) - f(x^*)) \left( 1 - \frac{1}{\kappa^*} + \frac{1}{\eta} \|P_r^* - P_1\|_F^2 + 4L^2 K \eta \right)^K. \end{aligned} \quad (44)$$

Take the stepsize  $\eta = \min \left\{ \frac{1}{4L^2}, \frac{\|P_r^* - P_1\|_F}{2L\sqrt{K}} \right\}$ . The bound (44) implies the desired result:

$$f(x^{K+1}) - f(x^*) \leq (f(x^1) - f(x^*)) \left( 1 - \frac{1}{\kappa^*} + \max \left\{ \frac{4L\|P_r^* - P_1\|_F}{\sqrt{K}}, \frac{8L^2\|P_r^* - P_1\|_F^2}{K} \right\} \right)^K.$$

## B.9 Proof of Proposition 4.2

Given the optimization problem (18),

$$\min_{P \in \mathcal{P}_+} \quad \kappa \quad \text{subject to} \quad \frac{1}{\kappa} I \preceq P^{1/2} A P^{1/2} \preceq I,$$

we can define  $\tau = 1/\kappa$  and reduce it to a standard semidefinite optimization problem (SDP)

$$\max_{P \in \mathcal{P}_+} \quad \tau = \kappa^{-1} \quad \text{subject to} \quad A^{-1} \tau \preceq P \preceq A^{-1}. \quad (45)$$

On the other hand, using  $f(x) = \frac{1}{2} \langle x, A, x \rangle$ , we can explicitly write

$$r_x(P) = \frac{\frac{1}{2}\langle x, A(PAP - 2P)Ax \rangle}{\frac{1}{2}\langle x, Ax \rangle}$$

and  $r_x$  is degree-zero homogeneous in  $x$ . Therefore, we can consider the following optimization problem

$$\min_{P \in \mathcal{P}_+} \max_{\langle x, Ax \rangle = 1} \langle x, A(PAP - 2P)Ax \rangle,$$

which can be further re-written as

$$\max_{P \in \mathcal{P}_+} \lambda \quad \text{subject to} \quad 2A^{1/2}PA^{1/2} - A^{1/2}PAPA^{1/2} \succeq \lambda I$$

Next we do variable replacement by letting  $M := A^{1/2}PA^{1/2}$  and  $\mathcal{P}'_+ = \{M = A^{1/2}PA^{1/2} : P \in \mathcal{P}_+\}$  and it suffices to show the equivalence between the following two problems.

$$\max_{M \in \mathcal{P}'_+} \tau \quad \text{subject to} \quad \tau I \preceq M \preceq I \quad (\text{SDP})$$

$$\max_{M \in \mathcal{P}'_+} \lambda \quad \text{subject to} \quad 2M - M^2 \succeq \lambda I \quad (\text{Minimax})$$

Given optimal solution  $(M_1^*, \tau^*)$  to (SDP), we have  $\tau^* I \preceq M_1^* \preceq I$  and let  $M_1^* = Q\Lambda_1 Q^\top$ . Plugging  $M_1^*$  into the constraint,

$$2M_1^* - (M_1^*)^2 = Q(2\Lambda_1 - \Lambda_1^2)Q^\top,$$

which corresponds to  $\lambda = 2\tau^* - (\tau^*)^2$ . On the other hand, given optimal solution  $(M_2^*, \lambda^*)$  to (Minimax),  $2M_2^* - M_2^{*2} \succeq \lambda^* I$  and similarly we let  $M_2^* = Q\Lambda_2 Q^\top$ . Then

$$Q(2\Lambda_2 - \Lambda_2^2)Q^\top \succeq \lambda^* I$$

and there exists some  $q_j$  such that  $2\lambda_{2j} - \lambda_{2j}^2 = \lambda^*$ . It corresponds to  $\tau = 2\tau^2 - \tau = \lambda^*$ . This establishes the equivalence between the two problems and completes the proof.

## B.10 Proof of Theorem 4.4

Recall that by (44) we have

$$f(x^{K+1}) - f(x^*) \leq (f(x^1) - f(x^*)) \left( \frac{1}{K} \sum_{k=1}^K r_{x^k}(P_r^*) + \frac{1}{\eta} \|P_r^* - P_1\|_F^2 + 4L^2 \eta \sum_{k=1}^K r_{x^k}(P_r^*) \right).$$

Using  $r_{x^k}(P_r^*) = 0$ ,  $P_r^* = A^{-1}$  and taking  $\eta = 1/(4L^2)$ , we get

$$f(x^{K+1}) - f(x^*) \leq (f(x^1) - f(x^*)) \left( \frac{4L^2 \|P_1 - A^{-1}\|_F^2}{K} \right) K.$$

This completes the proof.

## C Function value ratio surrogate with optimal value lower bound

This section analyzes the sub-optimal ratio surrogate loss  $r_x^z(P)$  defined in (20). Recall that

$$r_x^z(P) := \frac{f(x - P\nabla f(x)) - z}{f(x) - z} = \frac{f(x^+) - z}{f(x) - z}, \quad (46)$$

where  $z < f(x^*)$  is a lower bound for the optimal objective value. The challenging part of the analysis when  $z < f(x^*)$  is that the algorithm is only guaranteed to converge to some suboptimal solution whose



suboptimality is determined by  $f(x^*) - z$ , the accuracy of the lower bound. The analysis in this section is more involved than in **Section 4**, and for clarity, we *only present the global convergence result*.

### C.1 Surrogate loss

**Lemma C.1** (Surrogate loss and measure). *For all  $K \geq 1$ , the online scaled gradient method satisfies*

$$f(x^{K+1}) - z \leq (f(x^1) - z) \left( \frac{1}{K} \sum_{k=1}^K r_{x^k}^z(P_k) \right)^K. \quad (47)$$

**Proposition C.1** (Properties of  $r_{x^k}^z$ ). *Let  $z < f(x^*)$  be a given lower bound. Under **A1** and **A2**, for any fixed  $x$ , the surrogate loss  $r_x^z(P)$  defined in (46) is convex, non-negative, and  $2L^2$ -smooth as a function in  $P$ . In addition, the derivative of  $r_x^z$  takes the form*

$$\nabla r_x^z(P) = -\frac{\nabla f(x - P \nabla f(x)) \nabla f(x)^\top}{f(x) - z}.$$

### C.2 Online learning algorithm

**Lemma C.2** (Learnability). *Given **A1**, **A2**, and the ratio surrogate losses  $\{r_{x^k}^z\}$ , online gradient descent*

$$P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta \nabla r_{x^k}^z(P_k)] \quad (48)$$

*with stepsize  $\eta \leq 1/(4L^2)$  generates a sequence of scaling matrices  $\{P_k\}_{k \geq 2}$  such that*

$$\sum_{k=1}^K r_{x^k}^z(P_k) - \sum_{k=1}^K r_{x^k}^z(P) \leq \frac{1}{\eta} \|P - P_1\|_F^2 + 4L^2 \eta \sum_{k=1}^K r_{x^k}^z(P) \quad \text{for any } P \in \mathcal{P}. \quad (49)$$

### C.3 Algorithm design and analysis

In this section, we show how to obtain an  $\mathcal{O}(\kappa^* \log^2(1/\varepsilon))$  complexity through a double-loop algorithm. Since the double-loop algorithm deviates from our framework, only global convergence is established for brevity. We start by specifying the **OSGM-RZ**, a subroutine that will be invoked in the inner loop.

We choose the optimality measure  $\varphi$ , the surrogate loss  $\ell$ , and the online learning algorithm  $\mathcal{A}$  to be

$$\varphi(x) := f(x) - f(x^*), \quad \ell_x(P) := r_x^z(P), \quad \mathcal{A} := \text{online gradient descent in (48)},$$

and the monotone oracle  $\mathcal{M}$  is optional. **Algorithm 5** presents **OSGM-RZ** without the monotone oracle.

---

**Algorithm 5:** Online scaled gradient method with lower bound ratio surrogate (**OSGM-RZ**)

---

```

input  $x^1, P_1 \in \mathcal{P}, \eta > 0, z < f(x^*)$ 
for  $k = 1, 2, \dots$  do
     $x^{k+1} = x^k - P_k \nabla f(x^k)$ 
     $P_{k+1} = \Pi_{\mathcal{P}}[P_k - \eta \nabla r_{x^k}^z(P_k)]$ 
end
output  $x^{\text{best}}$  with minimum objective value

```

---

**Theorem C.1** characterizes the convergence behavior of **OSGM-RZ**.

**Theorem C.1** (Global convergence with lower bound). *Under **A1** to **A3**, **Algorithm 5** (**OSGM-RZ**) with  $\eta = \min\{\frac{1}{4L^2}, \frac{\|P_r^* - P_1\|_F}{2L\sqrt{K}}\}$  satisfies*

$$\min_{1 \leq k \leq K+1} f(x^k) - f(x^*) \leq \frac{1}{2}(f(x^*) - z) + (f(x^1) - f(x^*)) \left(1 - \frac{1}{2\kappa^*} + \frac{\rho_K}{K}\right)^K,$$

where  $\rho_K := \max\{4L\sqrt{K}\|P_r^* - P_1\|_F, 8L^2\|P_r^* - P_1\|_F^2\}$ .

**Lemma C.3** (Lower bound update). *Under the same assumptions and parameter choice as [Theorem C.1](#) and denote*

$$z^+ = \frac{1}{2} \left[ \min_{1 \leq k \leq K+1} f(x^k) + z \right].$$

*Then exactly one of the cases below happens:*

- $f(x^{K+1}) - f(x^*) \leq (f(x^1) - f(x^*)) \left(1 - \frac{1}{2\kappa^*} + \frac{\rho_K}{K}\right)^K$ , or
- $f(x^*) - z^+ \leq \frac{1}{2}(f(x^*) - z)$  and  $z^+ \leq f(x^*)$ .

**Lemma C.3** suggests that the output of OSGM-RZ either already satisfies the desirable convergence result, or the accuracy of the lower bound can be improved by a factor of 2. This motivates the idea of running OSGM-RZ multiple times and outputting the best solution, as presented in [Algorithm 6](#).

---

**Algorithm 6:** Online scaled gradient method with ratio surrogate and lower bound update

---

```

input  $x^1, P_1 \in \mathcal{P}, \eta > 0, z^1 < f(x^*)$ 
for  $t = 1, 2, \dots$  do
  |  $x^{t+1} = \text{OSGM-RZ}(x^t, P_1, \eta, z^t)$ 
  |  $z^{t+1} = \frac{1}{2}(f(x^{t+1}) + z^t)$ 
end
output  $x^{\text{best}}$  with minimum objective value

```

---

[Theorem C.2](#) provides the final convergence result.

**Theorem C.2.** *Under the same assumptions and parameter choices as [Theorem C.1](#), [Algorithm 6](#) attains  $f(x^{\text{best}}) - f(x^*) \leq \varepsilon$  in at most  $\mathcal{O}(\kappa^* \log^2(1/\varepsilon))$  scaled gradient iterations.*

## C.4 Proof of Lemma C.1

Since  $f(x) - z > 0$ , applying [Theorem 3.1](#) with  $\varphi(x) = f(x) - z$  completes the proof.

## C.5 Proof of Proposition C.1

Since  $z < f(x)$ , both the numerator and the denominator of  $r_x^z$  are positive. Following the proof of [Proposition 4.1](#), we can show that  $r_x^z$  is convex. Since  $u_x$  is  $L\|\nabla f(x)\|^2$ -smooth,  $r_x^z$  is  $\frac{L\|\nabla f(x)\|^2}{f(x)-z}$ -smooth, and

$$\frac{L\|\nabla f(x)\|^2}{f(x)-z} < \frac{L\|\nabla f(x)\|^2}{f(x)-f(x^*)} = \frac{L\|\nabla f(x) - \nabla f(x^*)\|^2}{f(x)-f(x^*)} \leq 2L^2,$$

which completes the proof.

## C.6 Proof of Lemma C.2

By [Proposition C.1](#), the surrogate losses  $\{r_{x^k}^z\}$  are  $2L^2$ -smooth and non-negative. Applying [Lemma B.2](#) with  $\tau = 2L^2$  completes the proof.

## C.7 Proof of Theorem C.1

Using [Lemma C.1](#) and [Lemma C.2](#) with  $P = P_r^*$ ,

$$\begin{aligned}
f(x^{K+1}) - z &\leq (f(x^1) - z) \left( \frac{1}{K} \sum_{k=1}^K r_{x^k}^z(P_k) \right)^K \\
&\leq (f(x^1) - z) \left( \frac{1}{K} \sum_{k=1}^K r_{x^k}^z(P_r^*) + \frac{1}{K} \left[ \frac{1}{\eta} \|P - P_r^*\|_F^2 + 4L^2\eta \sum_{k=1}^K r_{x^k}^z(P_r^*) \right] \right)^K \\
&\leq (f(x^1) - z) \left( \frac{1}{K} \sum_{k=1}^K r_{x^k}^z(P_r^*) + \frac{1}{K} \left[ \frac{1}{\eta} \|P - P_r^*\|_F^2 + 4L^2\eta \right] \right)^K, \tag{50}
\end{aligned}$$

where (50) uses  $f(x - P_r^* \nabla f(x)) \leq f(x)$  and that  $r_x^z(P_r^*) = \frac{f(x - P_r^* \nabla f(x)) - z}{f(x) - z} \leq 1$ .

Taking  $\eta = \min\{\frac{1}{4L^2}, \frac{\|P_r^* - P_1\|_F}{2L\sqrt{K}}\}$  gives

$$\begin{aligned} f(x^{K+1}) - z &\leq (f(x^1) - z) \left( \frac{1}{K} \sum_{k=1}^K r_{x^k}^z(P_r^*) + \max\left\{ \frac{4L\|P_r^* - P_1\|_F}{\sqrt{K}}, \frac{8L^2\|P_r^* - P_1\|_F^2}{K} \right\} \right)^K \\ &= (f(x^1) - z) \left( \frac{1}{K} \sum_{k=1}^K r_{x^k}^z(P_r^*) + \frac{\rho_K}{K} \right)^K. \end{aligned}$$

Next we analyze  $\frac{1}{K} \sum_{k=1}^K r_{x^k}^z(P_r^*)$ , and using

$$f(x - P_r^* \nabla f(x)) - f(x^*) \leq \left(1 - \frac{1}{\kappa^*}\right)(f(x) - f(x^*)),$$

we deduce that

$$\begin{aligned} f(x - P_r^* \nabla f(x)) - z &= f(x - P_r^* \nabla f(x)) - f(x^*) + f(x^*) - z \\ &\leq \left(1 - \frac{1}{\kappa^*}\right)[f(x) - f(x^*)] + f(x^*) - z \\ &= \left(1 - \frac{1}{\kappa^*}\right)[f(x) - z] - \left(1 - \frac{1}{\kappa^*}\right)[f(x^*) - z] + f(x^*) - z \\ &= \left(1 - \frac{1}{\kappa^*}\right)[f(x) - z] + \frac{1}{\kappa^*}[f(x^*) - z] \end{aligned}$$

Dividing both sides by  $f(x) - z$  gives

$$\frac{f(x - P_r^* \nabla f(x)) - z}{f(x) - z} = \left(1 - \frac{1}{\kappa^*}\right) + \frac{1}{\kappa^*} \frac{f(x^*) - z}{f(x) - z} = 1 - \frac{1}{\kappa^*} \frac{f(x) - f(x^*)}{f(x) - z}.$$

Hence  $\frac{1}{K} \sum_{k=1}^K r_{x^k}^z(P_r^*) \leq 1 - \frac{1}{\kappa^*} \left(\frac{1}{K} \sum_{k=1}^K \frac{f(x^k) - f(x^*)}{f(x^k) - z}\right)$ . Now, we do case analysis

**Case 1.** Suppose  $\frac{f(x^k) - f(x^*)}{f(x^k) - z} \geq \frac{1}{2}$  for all  $1 \leq k \leq K$ , then  $\frac{1}{K} \sum_{k=1}^K r_{x^k}^z(P_r^*) \leq 1 - \frac{1}{2\kappa^*}$  and

$$\begin{aligned} \min_{1 \leq k \leq K+1} f(x^k) - f(x^*) &\leq f(x^{K+1}) - f(x^*) \\ &\leq (f(x^1) - f(x^*)) \left(1 - \frac{1}{2\kappa^*} + \frac{\rho_K}{K}\right)^K. \end{aligned}$$

**Case 2.** Otherwise, there is some  $1 \leq j \leq K$  such that  $\frac{f(x^j) - f(x^*)}{f(x^j) - z} \leq \frac{1}{2}$ , a re-arrangement gives  $2f(x^j) - 2f(x^*) \leq f(x^j) - z$  and

$$\min_{1 \leq k \leq K+1} f(x^k) - f(x^*) \leq f(x^j) - f(x^*) \leq \frac{1}{2}(f(x^j) - z).$$

Putting the two cases together, we complete the proof.

### C.8 Proof of Lemma C.3

The argument is the same as in **Theorem C.1**. In **Case 1**, we get the first convergence result. Otherwise, we know that there exists some  $1 \leq j \leq K$  such that  $\frac{f(x^j) - f(x^*)}{f(x^j) - z} \leq \frac{1}{2}$  and since  $\min_{1 \leq k \leq K} f(x^k) \leq f(x^j)$ , we have

$$\frac{\min_{1 \leq k \leq K} f(x^k) - f(x^*)}{\min_{1 \leq k \leq K} f(x^k) - z} \leq \frac{f(x^j) - f(x^*)}{f(x^j) - z} \leq \frac{1}{2}.$$

Rearranging the relation, we have  $z^+ = \frac{1}{2}[\min_{1 \leq k \leq K} f(x^k) + z] \leq f(x^*)$  and

$$\begin{aligned} f(x^*) - \frac{1}{2} \left[ \min_{1 \leq k \leq K} f(x^k) + z \right] &= \frac{1}{2} [f(x^*) - \min_{1 \leq k \leq K} f(x^k)] + \frac{1}{2} [f(x^*) - z] \\ &\leq \frac{1}{2} [f(x^*) - z]. \end{aligned}$$

This completes the proof.

## C.9 Proof of Theorem C.2

Denote  $x^{t+1}$  as the output of OSGM-RZ in iteration  $t$  of **Algorithm 6**. If we fall into **Case 1** in **Lemma C.3** after running OSGM-RZ for  $K$  iterations, then

$$f(x^{t+1}) - f(x^*) \leq (f(x^t) - f(x^*)) \left(1 - \frac{1}{2\kappa^*} + \frac{\rho_K}{K}\right)^K \quad (51)$$

since  $x^t$  is the initial point of OSGM-RZ in iteration  $t$ . Using the fact that  $z^1$  is a lower bound for  $f(x^*)$ , algebraic manipulation shows that the right-hand side of (51) is less than  $\varepsilon$  whenever

$$K \geq \frac{128\|P_r^* - P_1\|_F^2 L^2}{\log^2(1 - \frac{1}{2\kappa^*})} + 2\kappa^* \log\left(\frac{f(x^1) - z^1}{\varepsilon}\right) =: K_0.$$

We claim that if we run OSGM-RZ for  $K_0$  iterations at each iteration  $t$  in **Algorithm 6** and run **Algorithm 6** for  $T := \frac{1}{\log 2} \log\left(\frac{4(f(x^1) - z^1)}{\varepsilon}\right)$  iterations, then we have  $f(x^{\text{best}}) - f(x^*) \leq \varepsilon$  where  $x^{\text{best}}$  is the point in  $\{x^t : t = 1, \dots, T+1\}$  with the smallest function value. Hence, **Algorithm 6** takes at most  $K_0 T = \mathcal{O}(\kappa^* \log^2(1/\varepsilon))$  scaled gradient iterations.

We will show that at least one of the iterates in  $\{x^t : t = 1, \dots, T+1\}$  from our algorithm satisfies  $f(x^t) - f(x^*) < \varepsilon$ . If we fall into **Case 1** in **Lemma C.3** for some iteration  $t$ , then we have  $f(x^{t+1}) - f(x^*) \leq \varepsilon$  by (51). Otherwise, we fall into **Case 2** in **Lemma C.3** for all  $t \leq T$ . In this case, we halve the distance between  $z^t$  and  $f(x^*)$  after every outer iteration, so that after  $T := \lceil \frac{1}{\log 2} \log\left(\frac{4(f(x^1) - z^1)}{\varepsilon}\right) \rceil$  iterations, we have

$$|z^T - f(x^*)| \leq \left(\frac{1}{2}\right)^{T-1} (f(x^*) - z^1) \leq \left(\frac{1}{2}\right)^{T-1} (f(x^1) - z^1) \leq \frac{\varepsilon}{2}.$$

Since  $z^{T+1} = \frac{1}{2}(f(x^{T+1}) + z^T)$  and we fall into **Case 2** at iteration  $T$ , we have

$$|z^{T+1} - f(x^*)| = \left| \frac{f(x^{T+1}) + z^T}{2} - f(x^*) \right| \leq \frac{1}{2} |z^T - f(x^*)| \leq \frac{\varepsilon}{4}.$$

Rearranging the relation, we have  $f(x^{T+1}) \leq f(x^*) + \frac{1}{2}\varepsilon + (f(x^*) - z^T) \leq f(x^*) + \varepsilon$ . This completes the proof.

## D Proof of results in Section 5

### D.1 Proof Lemma 5.1

Given monotone oracle  $\mathcal{M}$  with respect to gradient norm and by definition of  $g_x$ ,

$$\|\nabla f(x^{k+1})\| = \|\nabla f(\mathcal{M}(x^k))\| \leq \|\nabla f(x^k - P_k \nabla f(x^k))\| = g_{x^k}(P_k) \|\nabla f(x^k)\|.$$

Hence, through the same argument as in **Theorem 3.1**, we deduce that

$$\frac{\|\nabla f(x^{K+1})\|}{\|\nabla f(x^1)\|} = \prod_{k=1}^K \frac{\|\nabla f(x^{k+1})\|}{\|\nabla f(x^k)\|} \leq \left(\frac{1}{K} \sum_{k=1}^K \frac{\|\nabla f(x^{k+1})\|}{\|\nabla f(x^k)\|}\right)^K \leq \left(\frac{1}{K} \sum_{k=1}^K g_{x^k}(P_k)\right)^K$$

and this completes the proof.

## D.2 Proof of Proposition 5.1

Lipschitzness of  $g_x$  is straight-forward:

$$\begin{aligned} |g_x(P_1) - g_x(P_2)| &= \left| \frac{\|\nabla f(x - P_1 \nabla f(x))\|}{\|\nabla f(x)\|} - \frac{\|\nabla f(x - P_2 \nabla f(x))\|}{\|\nabla f(x)\|} \right| \\ &\leq \frac{\|\nabla f(x - P_1 \nabla f(x)) - \nabla f(x - P_2 \nabla f(x))\|}{\|\nabla f(x)\|} \end{aligned} \quad (52)$$

$$\begin{aligned} &\leq \frac{L\|P_1 - P_2\| \cdot \|\nabla f(x)\|}{\|\nabla f(x)\|} \\ &= L\|P_1 - P_2\| \leq L\|P_1 - P_2\|_F \end{aligned} \quad (53)$$

where (52) uses the triangle inequality  $\| \|a\| - \|b\| \| \leq \|a - b\|$  and (53) uses  $L$ -smoothness of  $f$ . Next consider  $|g_x(P) - \hat{g}_x(P)|$  and we deduce that

$$\begin{aligned} |g_x(P) - \hat{g}_x(P)| &= \left\| \frac{\nabla f(x)}{\|\nabla f(x)\|} - \int_0^1 \nabla^2 f(x - tP\nabla f(x)) P \frac{\nabla f(x)}{\|\nabla f(x)\|} dt \right\| - \left\| \frac{\nabla f(x)}{\|\nabla f(x)\|} - \nabla^2 f(x) P \frac{\nabla f(x)}{\|\nabla f(x)\|} \right\| \\ &\leq \left\| \int_0^1 \nabla^2 f(x - tP\nabla f(x)) P \frac{\nabla f(x)}{\|\nabla f(x)\|} - \nabla^2 f(x) P \frac{\nabla f(x)}{\|\nabla f(x)\|} dt \right\| \end{aligned} \quad (54)$$

$$\leq \int_0^1 \|\nabla^2 f(x - tP\nabla f(x)) - \nabla^2 f(x)\| dt \cdot \left( \frac{\|P\| \|\nabla f(x)\|}{\|\nabla f(x)\|} \right) \quad (55)$$

$$\leq H \int_0^1 \|P\nabla f(x)\| t dt \cdot \|P\| \quad (56)$$

$$\leq \frac{1}{2} H \|P\|^2 \|\nabla f(x)\| \quad (57)$$

where (54) again uses  $\| \|a\| - \|b\| \| \leq \|a - b\|$  and (55) uses the Lipschitzness of the Hessian; (56) uses  $\|AB\| \leq \|A\| \|B\|$ . Convexity of  $\hat{g}_x$  is straight-forward since  $\hat{g}_x$  is a composition of linear function (in  $P$ ) with norm  $\|\cdot\|$ . To show  $L$ -Lipschitzness of  $\hat{g}_x$ , we have

$$\begin{aligned} |\hat{g}_x(P_1) - \hat{g}_x(P_2)| &= \left\| \frac{\nabla f(x)}{\|\nabla f(x)\|} - \nabla^2 f(x) P_1 \frac{\nabla f(x)}{\|\nabla f(x)\|} \right\| - \left\| \frac{\nabla f(x)}{\|\nabla f(x)\|} - \nabla^2 f(x) P_2 \frac{\nabla f(x)}{\|\nabla f(x)\|} \right\| \\ &\leq \frac{1}{\|\nabla f(x)\|} \|\nabla^2 f(x)(P_1 - P_2)\nabla f(x)\| \end{aligned} \quad (58)$$

$$\leq L\|P_1 - P_2\|, \quad (59)$$

where (58) again uses  $\| \|a\| - \|b\| \| \leq \|a - b\|$  and (59) uses  $\|\nabla^2 f(x)\| \leq L$ . Last, we combine the convex subgradient lower bound of  $\hat{g}_x$  with the approximation

$$g_x(P_1) \geq \hat{g}_x(P_1) - \frac{1}{2} H \|P_1\|^2 \|\nabla f(x)\| \quad (60)$$

$$\geq \hat{g}_x(P_2) + \langle \hat{g}'_x(P_2), P_1 - P_2 \rangle - \frac{1}{2} H \|P_1\|^2 \|\nabla f(x)\| \quad (61)$$

$$\geq g_x(P_2) + \langle \hat{g}'_x(P_2), P_1 - P_2 \rangle - \frac{1}{2} H [\|P_1\|^2 + \|P_2\|^2] \|\nabla f(x)\| \quad (62)$$

$$\geq g_x(P_2) + \langle \hat{g}'_x(P_2), P_1 - P_2 \rangle - HD^2 \|\nabla f(x)\|,$$

where (60) uses (57), (61) uses convexity of  $\hat{g}_x$  and (62) again applies (57). This completes the proof.

## D.3 Proof of Lemma 5.2

Denote  $g_k(P_k) := g_{x^k}(P_k)$ . For any  $P \in \mathcal{P}$ , we have

$$\begin{aligned} \|P_{k+1} - P\|_F^2 &= \|\Pi_{\mathcal{P}}[P_k - \eta g'_k(P_k)]\|_F^2 \\ &\leq \|P_k - P - \eta g'_k(P_k)\|_F^2 \\ &= \|P_k - P\|_F^2 - 2\eta \langle g'_k(P_k), P_k - P \rangle + \eta^2 \|g'_k(P_k)\|_F^2 \\ &\leq \|P_k - P\|_F^2 - 2\eta [g_k(P_k) - g_k(P)] + \eta^2 L^2 + \eta HD^2 \|\nabla f(x^k)\|, \end{aligned} \quad (63)$$

where (63) invokes **Proposition 6.1**. Dividing both sides by  $\eta$  and re-arranging the terms,

$$\begin{aligned} g_k(P_k) - g_k(P) &\leq \frac{\|P_k - P\|_F^2}{2\eta} - \frac{\|P_{k+1} - P\|_F^2}{2\eta} + \frac{\eta}{2}L^2 + \frac{HD^2}{2}\|\nabla f(x^k)\| \\ &\leq \frac{\|P_k - P\|_F^2}{2\eta} - \frac{\|P_{k+1} - P\|_F^2}{2\eta} + \frac{\eta}{2}L^2 + \frac{HD^2}{2}\|\nabla f(x^1)\|, \end{aligned} \quad (64)$$

where (64) uses we use the fact that  $\|\nabla f(x^k)\| \leq \|\nabla f(x^1)\|$ . Summing both sides from  $k = 1, \dots, K$ , we get the desired result:

$$\sum_{k=1}^K g_k(P_k) - \sum_{k=1}^K g_k(P) \leq \frac{\|P_1 - P\|_F^2}{2\eta} + \frac{\eta}{2}L^2K + \frac{HD^2}{2}\|\nabla f(x^1)\|K. \quad (65)$$

Finally, using the bound  $\|P_1 - P\|_F^2 \leq 4D^2$  and plugging in the stepsize  $\eta = \frac{2D}{L\sqrt{K}}$  yield (24).

#### D.4 Proof of Theorem 5.1

By **Lemma 5.2**, we have

$$\frac{1}{K} \sum_{k=1}^K g_{x^k}(P_k) \leq \min_{P \in \mathcal{P}} \frac{1}{K} \sum_{k=1}^K g_{x^k}(P) + \frac{\rho_K}{K}$$

Plugging the relation into **Lemma 5.1** completes the proof.

#### D.5 Proof of Lemma 5.3

The first relation follows from

$$\|\nabla f(x - P_g^* \nabla f(x))\| \leq [\int_0^1 \|I - \nabla^2 f(x - tP_g^* \nabla f(x))P_g^*\| dt] \cdot \|\nabla f(x)\| \leq \omega^* \|\nabla f(x)\| = (1 - \frac{1}{\lambda^*}) \|\nabla f(x)\|. \quad (66)$$

The fact that  $\frac{\mu}{L}I \preceq L^{-1}\nabla^2 f(x) \preceq I$  for all  $x$  implies  $\|I - L^{-1}\nabla^2 f(x)\| \leq 1 - \frac{\mu}{L}$  for all  $x$ . Hence, by taking  $P = \frac{1}{L}I \in \mathcal{P}$ , we conclude

$$\omega^* := \min_{P \in \mathcal{P}} \max_x \|I - \nabla^2 f(x)P\| \leq \max_x \|I - L^{-1}\nabla^2 f(x)\| \leq 1 - \frac{\mu}{L}.$$

The desired inequality  $\lambda^* \leq \frac{L}{\mu}$  immediately follows from the definition  $\lambda^* = \frac{1}{1 - \omega^*}$ . Finally, rearranging (66) gives the desired bound on gradient norm surrogate loss:

$$g_x(P_g^*) = \frac{\|\nabla f(x - P_g^* \nabla f(x))\|}{\|\nabla f(x)\|} \leq 1 - \frac{1}{\lambda^*}.$$

*Remark 8.* We can link  $\lambda^*$  and  $\kappa^*$  through two relations below:

$$\begin{aligned} \|\nabla f(x - P_g^* \nabla f(x))\| &\leq (1 - \frac{1}{\lambda^*}) \|\nabla f(x)\|, \\ \|\nabla f(x - P_r^* \nabla f(x))\|_{P_r^*} &\leq (1 - \frac{1}{\kappa^*}) \|\nabla f(x)\|_{P_r^*}. \end{aligned} \quad (67)$$

The second relation (67) holds by simple algebraic derivation: since  $\frac{1}{\kappa^*}I \preceq (P_r^*)^{1/2}\nabla^2 f(x)(P_r^*)^{1/2} \preceq I$  for all  $x$ , we deduce that

$$\begin{aligned} &\|\nabla f(x - P_r^* \nabla f(x))\|_{P_r^*}^2 \\ &= \|\nabla f(x) - \int_0^1 \nabla^2 f(x - tP_r^* \nabla f(x))P_r^* \nabla f(x) dt\|_{P_r^*}^2 \\ &= \|\int_0^1 (I - \nabla^2 f(x - tP_r^* \nabla f(x))P_r^*) \nabla f(x) dt\|_{P_r^*}^2 \\ &= \langle \int_0^1 (I - \nabla^2 f(x - tP_r^* \nabla f(x))P_r^*) \nabla f(x) dt, P_r^* \int_0^1 (I - \nabla^2 f(x - tP_r^* \nabla f(x))P_r^*) \nabla f(x) dt \rangle \\ &= \langle (P_r^*)^{1/2} \nabla f(x), (\int_0^1 (I - M_t) dt)^2 (P_r^*)^{1/2} \nabla f(x) \rangle, \end{aligned}$$

where  $M_t := (P_r^*)^{1/2} \nabla^2 f(x - tP^* \nabla f(x)) (P_r^*)^{1/2}$ . Using the fact that  $\frac{1}{\kappa^*} \preceq M_t \preceq I$ , we have  $\int_0^1 (I - M)^2 dt \preceq (1 - \frac{1}{\kappa^*})^2 I$  and hence

$$\|\nabla f(x - P_r^* \nabla f(x))\|_{P_r^*}^2 \leq (1 - \frac{1}{\kappa^*})^2 \|\nabla f(x)\|_{P_r^*}^2.$$

Taking square root on both sides gives the desired relation. However, since evaluating  $\|\cdot\|_{P_r^*}$  requires knowledge of  $P_r^*$ , we have to define auxiliary quantity  $\lambda^*$  and  $P_g^*$ .

## D.6 Proof of Corollary 5.1

With convergence results from standard gradient descent, it takes  $\mathcal{O}(\kappa \log(HD^2 \lambda^*))$  iterations to output  $\hat{x}$  such that  $\|\nabla f(\hat{x})\| \leq \frac{1}{HD^2 \lambda^*}$ . Next let  $x^1 = \hat{x}$ . Using **Lemma 5.3** and **Theorem 5.1**,  $\theta_K^* \leq 1 - \frac{1}{\lambda^*}$  and

$$\begin{aligned} \|\nabla f(x^{K+1})\| &\leq \|\nabla f(x^1)\| \left(1 - \frac{1}{\lambda^*} + \frac{2DL}{\sqrt{K}} + \frac{HD^2}{2} \|\nabla f(x^1)\|\right)^K \\ &\leq \|\nabla f(x^1)\| \left(1 - \frac{1}{\lambda^*} + \frac{2DL}{\sqrt{K}} + \frac{1}{2\lambda^*}\right)^K \\ &\leq \|\nabla f(x^1)\| \left(1 - \frac{1}{2\lambda^*} + \frac{2DL}{\sqrt{K}}\right)^K, \end{aligned} \tag{68}$$

where (68) uses the assumption that  $\|\nabla f(x^1)\| \leq \frac{1}{HD^2 \lambda^*}$ . This completes the proof.

## E Proof of results in Section 6

### E.1 Proof of Lemma 6.1

For convenience we denote  $x^{k+1/2} := x^k - P_k \nabla f(x^k)$ . By definition of the monotone oracle, we always have

$$f(x^{k+1}) = f(\mathcal{M}(x^k)) \leq \min\{f(x^k), f(x^{k+1/2})\}.$$

**Proof of relation (26).** Suppose  $\mu \neq 0$ . By definition of  $h_{x^k}(P)$ , we can write

$$\begin{aligned} f(x^{k+1/2}) - f(x^*) &= f(x^k) - f(x^*) + h_{x^k}(P_k) \|\nabla f(x^k)\|^2 \\ &= (f(x^k) - f(x^*)) \left[1 + \frac{h_{x^k}(P_k) \|\nabla f(x^k)\|^2}{f(x^k) - f(x^*)}\right]. \end{aligned} \tag{69}$$

Since  $f(x^{k+1}) = f(\mathcal{M}(x^k)) \leq \min\{f(x^k), f(x^{k+1/2})\}$ , we have

$$\begin{aligned} f(x^{k+1}) - f(x^*) &\leq \min\{f(x^k) - f(x^*), f(x^{k+1/2}) - f(x^*)\} \\ &= (f(x^k) - f(x^*)) \left[1 + \min\left\{\frac{h_{x^k}(P_k) \|\nabla f(x^k)\|^2}{f(x^k) - f(x^*)}, 0\right\}\right], \end{aligned} \tag{70}$$

where (70) uses (69). We successively deduce that

$$\begin{aligned} \frac{f(x^{K+1}) - f(x^*)}{f(x^1) - f(x^*)} &= \prod_{k=1}^K \frac{f(x^{k+1}) - f(x^*)}{f(x^k) - f(x^*)} \\ &\leq \left(\frac{1}{K} \sum_{k=1}^K \frac{f(x^{k+1}) - f(x^*)}{f(x^k) - f(x^*)}\right)^K \\ &\leq \left(1 + \frac{1}{K} \sum_{k=1}^K \min\left\{\frac{h_{x^k}(P_k) \|\nabla f(x^k)\|^2}{f(x^k) - f(x^*)}, 0\right\}\right)^K \\ &\leq \left(1 + \frac{2\mu}{K} \sum_{k=1}^K \min\{h_{x^k}(P_k), 0\}\right)^K, \end{aligned} \tag{71}$$

where (71) is by  $\frac{1}{2\mu} \|\nabla f(x^k)\|^2 \geq f(x^k) - f(x^*)$  and  $\min\{h_{x^k}(P_k), 0\} \leq 0$ :

$$\min\left\{\frac{h_{x^k}(P_k) \|\nabla f(x^k)\|^2}{f(x^k) - f(x^*)}, 0\right\} = \frac{\|\nabla f(x^k)\|^2}{f(x^k) - f(x^*)} \cdot \min\{h_{x^k}(P_k), 0\} \leq 2\mu \min\{h_{x^k}(P_k), 0\}.$$

By concavity of  $\min\{\cdot, 0\}$ , we have

$$1 + \frac{2\mu}{K} \sum_{k=1}^K \min\{h_{x^k}(P_k), 0\} \leq 1 + 2\mu \min\{\frac{1}{K} \sum_{k=1}^K h_{x^k}(P_k), 0\}$$

and using the identity  $\max\{\cdot, 0\} = -\min\{-\cdot, 0\}$  completes the proof.

**Proof of relation (27).** Again by definition of  $h_x(P)$ ,  $f(x^{k+1/2}) - f(x^k) = h_{x^k}(P_k) \|\nabla f(x^k)\|^2$  and

$$f(x^{k+1}) - f(x^k) \leq \min\{f(x^{k+1/2}) - f(x^k), f(x^k) - f(x^k)\} = \min\{h_{x^k}(P_k), 0\} \|\nabla f(x^k)\|^2.$$

Summing the inequality from  $k = 1$  to  $K$ , we have

$$f(x^{K+1}) - f(x^1) \leq \sum_{k=1}^K \min\{h_{x^k}(P_k), 0\} \|\nabla f(x^k)\|^2.$$

Re-arrangement gives

$$\begin{aligned} & (\sum_{k=1}^K \max\{-h_{x^k}(P_k), 0\}) \cdot \min_{1 \leq k \leq K} \|\nabla f(x^k)\|^2 \\ & \leq \sum_{k=1}^K \max\{-h_{x^k}(P_k), 0\} \|\nabla f(x^k)\|^2 \\ & \leq f(x^1) - f(x^{K+1}) \\ & \leq f(x^1) - f(x^*) \end{aligned}$$

Last, using convexity of  $\max\{\cdot, 0\}$ ,

$$\begin{aligned} \min_{1 \leq k \leq K} \|\nabla f(x^k)\|^2 & \leq \frac{f(x^1) - f(x^*)}{K} \frac{1}{\frac{1}{K} \sum_{k=1}^K \max\{-h_{x^k}(P_k), 0\}} \\ & \leq \frac{f(x^1) - f(x^*)}{K} \frac{1}{\max\{\frac{1}{K} \sum_{k=1}^K -h_{x^k}(P_k), 0\}} \end{aligned}$$

and this completes the proof.

**Proof of relation (28).** Take  $x^*$  to be the equilibrium of the inner problem  $\max_{x \in \mathcal{L}_{f(x^1)}} \min_{x^* \in \mathcal{X}^*} \|x - x^*\|$ , we deduce that

$$\begin{aligned} f(x^{k+1}) - f(x^*) & \leq f(x^k) - f(x^*) + \min\{h_{x^k}(P_k), 0\} \|\nabla f(x^k)\|^2 \\ & = f(x^k) - f(x^*) + \min\{h_{x^k}(P_k), 0\} \frac{\|\nabla f(x^k)\|^2 \|x^k - x^*\|^2}{(f(x^k) - f(x^*))^2} \frac{[f(x^k) - f(x^*)]^2}{\|x^k - x^*\|^2} \\ & \leq f(x^k) - f(x^*) + \min\{h_{x^k}(P_k), 0\} \frac{[f(x^k) - f(x^*)]^2}{\|x^k - x^*\|^2}, \end{aligned} \tag{72}$$

where the last inequality uses  $f(x^k) - f(x^*) \leq \|\nabla f(x^k)\| \|x^k - x^*\|$  and that

$$\min\{h_{x^k}(P_k), 0\} \cdot \frac{\|\nabla f(x^k)\|^2 \|x^k - x^*\|^2}{[f(x^k) - f(x^*)]^2} \frac{[f(x^k) - f(x^*)]^2}{\|x^k - x^*\|^2} \leq \min\{h_{x^k}(P_k), 0\} \cdot \frac{[f(x^k) - f(x^*)]^2}{\|x^k - x^*\|^2}$$

since  $\min\{h_{x^k}(P_k), 0\} \leq 0$ . Re-arranging the terms, we get

$$\begin{aligned} \frac{1}{f(x^{k+1}) - f(x^*)} - \frac{1}{f(x^k) - f(x^*)} & = \frac{f(x^k) - f(x^*) - [f(x^{k+1}) - f(x^*)]}{[f(x^{k+1}) - f(x^*)][f(x^k) - f(x^*)]} \\ & \geq \frac{-\min\{h_{x^k}(P_k), 0\} \frac{[f(x^k) - f(x^*)]^2}{\|x^k - x^*\|^2}}{[f(x^{k+1}) - f(x^*)][f(x^k) - f(x^*)]} \end{aligned} \tag{73}$$

$$\begin{aligned} & = \frac{-\min\{h_{x^k}(P_k), 0\} [f(x^k) - f(x^*)]}{[f(x^{k+1}) - f(x^*)] \|x^k - x^*\|^2} \\ & \geq \frac{-\min\{h_{x^k}(P_k), 0\}}{\|x^k - x^*\|^2} \geq -\frac{1}{\Delta^2} \min\{h_{x^k}(P_k), 0\}, \end{aligned} \tag{74}$$



where (73) plugs in (72); (74) uses the fact that  $f(x^k) \leq f(x^1)$  and that

$$\|x^k - x^*\| \leq \max_{x \in \mathcal{L}_{f(x^1)}} \|x - x^*\| = \Delta.$$

Finally, we telescope the relation

$$\frac{1}{f(x^{k+1}) - f(x^*)} - \frac{1}{f(x^k) - f(x^*)} \geq -\frac{1}{\Delta^2} \min\{h_{x^k}(P_k), 0\}$$

from  $k = 1$  to  $K$  and get

$$\begin{aligned} \frac{1}{f(x^{K+1}) - f(x^*)} - \frac{1}{f(x^1) - f(x^*)} &= \sum_{k=1}^K \frac{1}{f(x^{k+1}) - f(x^*)} - \frac{1}{f(x^k) - f(x^*)} \\ &\geq -\frac{1}{\Delta^2} \sum_{k=1}^K \min\{h_{x^k}(P_k), 0\} \\ &= \frac{1}{\Delta^2} \sum_{k=1}^K \max\{-h_{x^k}(P_k), 0\}. \end{aligned}$$

Re-arranging the terms and using convexity of  $\max\{\cdot, 0\}$ ,

$$f(x^{K+1}) - f(x^*) \leq \frac{\Delta^2}{\sum_{k=1}^K \max\{-h_{x^k}(P_k), 0\}} \leq \frac{\Delta^2}{K} \frac{1}{\max\{-\frac{1}{K} \sum_{k=1}^K h_{x^k}(P_k), 0\}}$$

and this completes the proof.

## E.2 Proof of Proposition 6.1

To show the Lipschitzness of  $h_x$ , it suffices to show the gradient is bounded. Given  $\nabla h_x(P) = \frac{\nabla f(x - P\nabla f(x))\nabla f(x)^\top}{\|\nabla f(x)\|^2}$ , we deduce that

$$\begin{aligned} \|\nabla h_x(P)\|_F &= \frac{\|\nabla f(x - P\nabla f(x))\nabla f(x)^\top\|_F}{\|\nabla f(x)\|^2} \\ &= \frac{\|\nabla f(x - P\nabla f(x))\|}{\|\nabla f(x)\|} \end{aligned} \tag{75}$$

$$\leq \frac{\|\nabla f(x - P\nabla f(x)) - \nabla f(x)\| + \|\nabla f(x)\|}{\|\nabla f(x)\|} \tag{76}$$

$$\leq \frac{L\|P\nabla f(x)\|}{\|\nabla f(x)\|} + 1 \tag{77}$$

$$\leq L\|P\| + 1 \leq LD + 1,$$

where (75) uses  $\|ab^\top\|_F = \|a\| \cdot \|b\|$  and (77) applies  $L$ -Lipschitzness of  $\nabla f(x)$ .

## E.3 Proof of Lemma 6.2

The proof is again a direct application of the results in online convex optimization. For any  $P \in \mathcal{P}$ , (37) gives

$$\sum_{k=1}^K h_{x^k}(P_k) - \sum_{k=1}^K h_{x^k}(P) \leq \frac{1}{2\eta} \|P_1 - P\|_F^2 + \frac{\eta}{2} \sum_{k=1}^K \|\nabla h_{x^k}(P_k)\|_F^2 \leq \frac{2D^2}{\eta} + \frac{\eta(LD+1)^2}{2} K,$$

where the last inequality  $\|P_1 - P\|_F \leq \|P_1\|_F + \|P\|_F \leq 2D$  and the bounded gradient  $\|\nabla h_{x^k}(P)\|_F \leq L(D+1)$ . Taking  $\eta$  to minimize the right-hand side completes the proof.

## E.4 Proof of Theorem 6.1

By **Lemma 6.2**, we have

$$\frac{1}{K} \sum_{k=1}^K h_{x^k}(P_k) \leq \frac{1}{K} \sum_{k=1}^K h_{x^k}(P) + \frac{\rho K}{K}$$

for all  $P \in \mathcal{P}$  and plugging  $-\frac{1}{K} \sum_{k=1}^K h_{x^k}(P_k) \geq -\theta_P^* - \frac{\rho K}{K}$  into **Lemma 6.1** completes the proof.

### E.5 Proof of Lemma 6.3

According to **A3**,  $L^{-1}I \in \mathcal{P}$  and descent lemma gives, for all  $x \notin \mathcal{X}^*$ , that

$$h_x(L^{-1}I) = \frac{f(x - \frac{1}{L}\nabla f(x)) - f(x)}{\|\nabla f(x)\|^2} \leq -\frac{1}{2L}$$

and this completes the proof.

### E.6 Proof of Corollary 6.1

Using **Lemma 6.3** and **Theorem 6.1**,  $\theta_K^* \leq -\gamma^*$  and plugging the bound back into **Theorem 6.1** completes the proof.