arXiv:submit/8327146 [math.OC] 7 Nov 2026

A Low-Rank Augmented Lagrangian Frank-Wolfe-Based Method for SDPs with Unbounded Feasible Regions *

Jacob M. Aguirre† Renato D.C. Monteiro‡ Arnesh Sujanani§

November 7, 2026

Abstract

In contrast to HALLaR (Monteiro et al. 2026), which is designed for solving problems with a trace-bounding constraint, this work develops a new augmented Lagrangian (AL) method, HALLaR+, for solving large-scale semidefinite programming problems without such constraint. Even though a trace constraint which preserves an optimal solution can always be added, choosing its bound is challenging: a value that is too small may exclude all optimal solutions, whereas one that is too large may substantially increase the computational effort. HALLaR+ is also a low-rank inexact AL method but, in contrast to HALLaR, it doubles the penalty parameter at every AL iteration and its AL subproblems have unbounded feasible regions that are approximately solved by a relaxed unbounded Frank–Wolfe (RUFW) variant. Its analysis relies on a uniform bound on the diameters of the sublevel sets of all generated AL subproblems instead of a bounded feasible set. Assuming that a primal–dual KKT pair exists and the primal optimal set is nonempty and bounded, it is shown that HALLaR+ returns a ε-primal-dual feasible solution in $\tilde{\mathcal{O}}(\varepsilon^{-2})$ total RUFW iterations. Numerical results suggest that HALLaR+ has very promising competitive performance against current state-of-the-art solvers.

Keywords: semidefinite programming, SDP, augmented Lagrangian, low-rank, proximal point, iteration complexity, parameter-free method, Frank-Wolfe, conditional gradient, GPU acceleration

1 Introduction

This paper develops a hybrid convex–nonconvex augmented Lagrangian (AL) variant of HALLaR of [44], namely HALLaR+, which can solve SDPs, in general standard form, i.e.,

$$\min_{X\in\mathbb{S}^n} \{\langle C, X\rangle : \mathcal{A}X = b,\ X \succeq 0\} \tag{1}$$

where $\mathbb{R}^m$ is the m-dimensional Euclidean space equipped with the standard inner product, $\mathbb{S}^n$ is the space of $n\times n$ real symmetric matrices equipped with the Frobenius inner product, $\succeq 0$ denotes the Löwner partial order on $\mathbb{S}^n$, so that $X \succeq 0$ means X is positive semidefinite (PSD), $C \in \mathbb{S}^n$, $b \in \mathbb{R}^m$, and $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ is a linear operator with adjoint $\mathcal{A}^* : \mathbb{R}^m \to \mathbb{S}^n$. In contrast, HALLaR can only solve SDPs where one of the constraints in (1) is a bounded trace constraint, i.e., $\mathrm{tr}(X) \le \tau$ for some $\tau > 0$. Even though a trace constraint can always be added to (1) with a constant τ that preserves at least one optimal solution, choosing such a τ in practice is challenging: a value that is too small may exclude all optimal solutions, whereas one that is too large may substantially increase the computational effort.

Approach and relation to HALLaR. HALLaR+ is an inexact AL method that generates sequences $\{Z_t\}$ and $\{p_t\}$ according to the recursion

$$Z_t \approx \operatorname*{arg\,min}_{X\in\mathbb{S}^n} \left\{\mathcal{L}_{\beta_{t-1}}(X; p_{t-1}) := \langle C, X\rangle + \langle p_{t-1}, b - \mathcal{A}X\rangle + \frac{\beta_{t-1}}{2}\|b - \mathcal{A}X\|^2 \ :\ X \succeq 0\right\}, \tag{2a}$$

$$p_t = p_{t-1} + \beta_{t-1}(b - \mathcal{A}Z_t), \qquad \beta_t = \gamma\beta_{t-1}, \tag{2b}$$

*Funding: Jacob M. Aguirre is supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2039655. Renato D.C. Monteiro is supported by AFOSR Grant FA9550-25-1-0131.

†H. M. Stewart School of Industrial and Systems Engineering, Georgia Tech, Atlanta, GA, 30332-0205. aguirre@gatech.edu

‡School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205. monteiro@isye.gatech.edu

§Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong. arneshmurlisujanani@cuhk.edu.hk

with $\gamma = 2$. HALLaR differs from the above updates (2a) and (2b) in two aspects. First, as HALLAR solves SDPs over the set $\{X \in \mathbb{S}^n : A(X) = b,\ \mathrm{tr}(X) \le \tau,\ X \succeq 0\}$, its AL subproblems are obtained by dualizing $A(X) = b$, while keeping $\mathrm{tr}(X) \le \tau$ and $X \succeq 0$ as constraints. Hence, in contrast to (2a), their feasible regions are bounded. Second, the theoretical analysis of HALLaR assumes $\gamma = 1$, and hence that the penalty parameters are kept constant.

Both HALLaR and HALLaR+ use Frank–Wolfe (FW), or conditional-gradient, variants to approximately solve their AL subproblem (2a), initialized at Z_{t-1}. However, because the AL subproblems arising in HALLaR+ have an unbounded feasible region, the FW variant used to solve them relies on substantially different techniques from the one used by HALLaR. Both FW variants generate matrices in factored form, i.e., $X = UU^\top$, where the number of columns r of U varies during the algorithm but is usually much smaller than n. This low-rank representation reduces memory footprint from $O(n^2)$ to $O(nr)$ and exploits sparsity in the SDP data [8, 9, 44, 2]. Both X and U are viewed as iterates in our discussion, with U referred to as a factor of X. For a matrix V, $\mathrm{col}(V)$ denotes its number of columns. Starting from a low-rank factor $\tilde{U} \in \mathbb{R}^{n\times r}$, an iteration of both FW variants consists of the following two basic steps:

i) find a factor $U \in \mathbb{R}^{n\times r}$ such that $\mathrm{col}(U) \le \mathrm{col}(\tilde{U})$ and $\mathcal{L}_{\beta_{t-1}}(UU^\top; p_{t-1}) \le \mathcal{L}_{\beta_{t-1}}(\tilde{U}\tilde{U}^\top; p_{t-1})$;

ii) without forming the associated full matrix, check whether $UU^\top$ satisfies the prescribed approximate optimality conditions for (2a). If so, terminate and set $Z_t = UU^\top$; otherwise, perform an FW-type step based at $UU^\top$ to escape from this spurious solution U to obtain a subsequent low-rank factor $\tilde{U}^+$ satisfying $\mathrm{col}(\tilde{U}^+) \le \mathrm{col}(U)+1$, and hence $\mathrm{col}(\tilde{U}^+) \le \mathrm{col}(\tilde{U}) + 1$.

We now discuss (i) in greater detail. The choice $U = \tilde{U}$ trivially satisfies step i) and has been used by methods such as CGAL and T-CGAL [66, 67] (see also the methods in [68, 16, 52] for similar limited memory or reduced storage variants). Such methods can be viewed as essentially pure Frank-Wolfe methods, but have the disadvantage of producing iterates with high number of columns, since steps (i)–(ii) must generally be repeated many times. On the other hand, to implement the second part of (i), HALLaR and HALLaR+ both use the nonlinear solver ADAP-AIPP, initialized at $\tilde{U}$, to find a near-stationary point U of $\min_{V\in\mathbb{R}^{n\times r}} \mathcal{L}_{\beta_{t-1}}(VV^\top; p_{t-1})$ satisfying the descent condition in step i). Since the objective of this unconstrained subproblem is nonconvex and has only a locally Lipschitz continuous gradient, establishing the required complexity guarantees for ADAP-AIPP in this setting requires a more delicate analysis than the one used in HALLaR. The additional difficulty arises from the fact that, whereas the complexity of HALLaR depends on the diameter of the bounded feasible set, the complexity of HALLaR+ depends on a uniform bound on the diameters of all generated sublevel sets, some of which depend on both β_{t-1} and p_{t-1}.

We next discuss (ii). Unlike HALLaR, which uses a computable FW gap to estimate whether the primal gap of $\min_{X\in\mathbb{S}^n} \left\{\mathcal{L}_{\beta_{t-1}}(X; p_{t-1}) : X \succeq 0,\ \mathrm{tr}(X) \le 1\right\}$ at $X = UU^T$ is small, HALLaR+ uses approximate optimality conditions tailored to the unbounded subproblem (2). Specifically, with $S = \nabla_X \mathcal{L}_{\beta_{t-1}}(UU^\top; p_{t-1})$, it checks whether $\lambda_{\min}(S)$ is nearly nonnegative and $|\langle S, UU^\top\rangle|$ is small. If either condition fails, HALLaR+ finds a rank-one matrix $y^F(y^F)^\top$ by solving $\min_{X\succeq 0}\{\langle S, X\rangle : \mathrm{tr}(X) \le 1\}$ and forms the next iterate as a conic combination of this matrix and the current iterate, with the two coefficients of the conic combination chosen by a two- dimensional line search.

This approach for solving each AL subproblem is closely related to those of Laue [35] and Duchi et al. [17, 36], which develop FW variants for the generic static problem of minimizing a convex function $\phi(\cdot)$ over $\mathbb{S}^n_+$. The two-dimensional line search for the conic coefficients used in our FW variant is motivated by the scheme of Laue. However, whereas Laue's convergence analysis is based on a primal-gap criterion that is not directly checkable in practice, our method uses the aforementioned tractable termination criteria. Duchi et al., on the other hand, propose a different unbounded FW variant based on two successive one-dimensional line-searches rather than a single two-dimensional line search. Like our FW variant, their method terminates based on checkable approximate optimality conditions. While closely related to these methods, our FW variant requires a more involved analysis because it must account for the changing AL subproblems in (2) and the dynamics of their associated sublevel sets, a difficulty absent from the static settings considered by Laue and Duchi et al.

Our results and computational impact. Assuming that a primal–dual KKT pair exists and that the set of optimal solutions of (1) is nonempty and bounded, it is shown that HALLaR+ obtains an approximate primal–dual solution of (1) and its dual with provable computational complexity bounds expressed in terms of parameters associated with the SDP instance and user-specified tolerances. In particular, if the tolerances on primal feasibility, dual feasibility, and complementarity are all equal to $\varepsilon \in (0, 1]$, then, HALLaR+ performs $\mathcal{O}(\log(1/\varepsilon))$ AL iterations, $\mathcal{O}(\varepsilon^{-2})$ FW iterations/minimum eigenvector computations, and $\mathcal{O}(\varepsilon^{-3})$) ADAP-AIPP iterations.

Our GPU implementation cuHALLaR+ is faster than cuHALLaR [2] on 12 of 14 Hamming and 24 of 28 GSET maximum stable set instances, as well as all 26 matrix completion instances. On a single NVIDIA H200, it reaches 10^{-5} relative precision in about 44 seconds on a Hamming SDP ($n \approx 8.4 \times 10^6$, $m \approx 9.7 \times 10^7$) and 31 seconds on

a matrix completion SDP ($n = 2 \times 10^6$, $m \approx 2.6 \times 10^8$). On the other hand, cuLoRADS [23] fails on the Hamming instance and takes about $4,100$ seconds on the matrix completion instance.

At 10^{-4} relative precision, cuHALLaR+ solves 80 of 81 QAPLIB relaxations, compared with 56 for CSDP [5], 17 for HALLaR, and one for cuLoRADS. On the Mittelmann benchmarks [43], cuHALLaR+ solves 27 instances within one hour, cuLoRADS solves 15, and CARDAL [37] solves 14. The final cuHALLaR+ iterate has rank two on every Hamming instance, at most $r + 1$ on every matrix completion instance with ground-truth rank r, and 465 on the largest random quadratic assignment instance.

Related works. Interior-point methods are widely used to solve small and medium-scale SDPs to high accuracy. Their applicability to large-scale instances is limited, however, because each Newton-type iteration typically requires solving a large linear system, often through the factorization of a large and potentially dense matrix [5, 3, 45, 59]. We therefore focus below on methods specifically designed for large-scale SDPs.

SDPNAL+ [69, 65] is an augmented Lagrangian (AL) method that approximately solves each AL subproblem using a semismooth Newton–CG method. Spectral bundle methods, and more generally bundle methods, have also been developed for large-scale SDPs (see, for example, [25, 10, 24, 14, 46, 4]). Another substantial line of work applies the alternating direction method of multipliers (ADMM) and related operator-splitting methods to large-scale SDPs (e.g., [47, 20, 13, 40, 70, 33, 12, 63, 48]). Since methods based on nonconvex low-rank (LR) factorizations are most closely related to the approach developed in this paper, we discuss them in greater detail below. We refer the reader to the survey [41] for a broader overview of methods for solving large-scale SDPs.

The nonconvex LR approach introduced in [8, 9] underlies many successful algorithms for solving important classes of SDPs. In particular, the SDPLR method developed in these works applies an AL method to the smooth nonconvex reformulation

$$\min_{U} \quad \left\{ C \bullet UU^T \quad : \quad \mathcal{A}(UU^T) = b, \quad \|U\|_F \le 1, \quad U \in \mathbb{R}^{n \times r} \right\} \tag{P_r}$$

and approximately solves its AL subproblems using a limited-memory BFGS method. Although SDPLR is designed to handle equality constraints, it can be modified to accommodate inequality constraints (see, for example, [34]).

Several methods solve (P_r) without employing an AL framework. For MaxCut SDPs, several specialized algorithms solve (P_r) using optimization techniques that preserve feasibility throughout their execution (e.g., [7, 27, 18, 42, 32]). Riemannian optimization methods have also been employed for structured classes of SDPs for which the feasible set of (P_r) is a smooth manifold (e.g., [51, 32, 30, 42]). Related feasibility-preserving methods based on manifold-optimization techniques have been developed for more challenging classes of SDPs in which the feasible sets of (P_r) are algebraic varieties [55, 56, 57]. Alternative feasibility-preserving strategies have also been explored in [50].

More recently, several low-rank SDP solvers have combined low-rank factorization with augmented-Lagrangian-type schemes [31, 6]. Like SDPLR, LoRADS and its GPU counterpart cuLoRADS [23, 22] use the factorization $X = UU^T$ in their first phase and apply an inexact AL method to (P_r). In their second phase, these methods adopt the bilinear factorization $X = UV^T$, penalize the additional constraint $U = V$, and apply an alternating-minimization scheme to the resulting augmented Lagrangian function. The ALORA method of [15] combines gradient descent with a spectral block Frank–Wolfe method to approximately minimize its nonconvex AL subproblems. CARDAL [37] combines rank-adaptive AL iterations with matrix-free L-BFGS and negative-curvature corrections, provides slack-based approximate optimality conditions, and supports distributed GPU execution. SDPF+ [58] combines a feasible low-rank method with an AL phase using a dynamically weighted quadratic penalty. Other low-rank methods focus on specially structured SDPs whose factorized nonconvex AL subproblems take the form $\arg\min_{U \in \mathcal{M}} \mathcal{L}_{\beta_{t-1}}(UU^\top; p_{t-1})$, where $\mathcal{M}$ is a Riemannian manifold. Such subproblems are solved using either a Riemannian trust-region method [60, 61] or a Riemannian semismooth Newton method [62]. For SDP-RLT and polyhedral-SDP relaxations, respectively, RiNNAL+ and RiNNAL-POP [29, 28] combine low-rank AL iterations with projected-gradient steps in the original matrix variable to adjust the rank and escape spurious factor-space local minima.

Structure of the paper. This paper is organized into five sections and two appendices. Section 2 introduces the pair of primal-dual SDP problems and develops a general AL framework for solving them. Section 3 presents X-AL, an instance of this framework whose iterates are generated in the $\mathbb{S}^n_+$ space, and establishes its main complexity result for solving the SDP pair. Section 4 introduces HALLaR+, a computationally and memory-efficient variant of X-AL implemented in the U-space. Theorem 4.1, the main theorem of the paper, establishes the complexity bounds for HALLaR+. Section 5 reports computational experiments comparing HALLaR+ with several existing solvers on a large collection of SDPs. Appendix A provides technical convex-analysis results used throughout the paper and develops the relaxed unbounded Frank-Wolfe (RUFW) framework used by X-AL. Finally, Appendix B presents the

convergence analyses of an adaptive version of FISTA and an inexact proximal point method, both implemented in the U-space and used as subroutines within HALLaR+.

1.1 Basic definitions and notation

Let $\mathbb{E}$ be a finite-dimensional inner-product space. We denote its inner product by $\langle X, Y\rangle$ and the associated norm by $\|X\| := \sqrt{\langle X, X\rangle}$. When $\mathbb{E} = \mathbb{S}^n$, we use the trace inner product $\langle X, Y\rangle := \mathrm{tr}(XY)$, whose associated norm is the Frobenius norm. We use the notation

$$\log_2^+(t) := \max\{0, \log_2 t\}, \qquad \log_1^+(t) := \max\{1, \log t\},$$

and write $\mathcal{O}_1(\Theta)$ as shorthand for $\mathcal{O}(1+\Theta)$. The operator norm of $\mathcal{A}$ is $\|\mathcal{A}\| := \max\{\|\mathcal{A}X\| : \|X\| \le 1\}$.

We first recall some standard definitions from convex analysis. For a set $\Omega \subseteq \mathbb{E}$, its indicator function is denoted by δ_Ω, where $\delta_\Omega(X) = 0$ if $X \in \Omega$ and $\delta_\Omega(X) = \infty$ otherwise. If Ω is closed and convex, its normal cone at $X \in \Omega$ is

$$\mathcal{N}_\Omega(X) := \{V : \langle V, U - X\rangle \le 0,\ \forall U \in \Omega\}. \tag{3}$$

For a proper closed convex function $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$, a point $X \in \mathrm{dom}\, f$, and a scalar $\varepsilon \ge 0$, the ε-subdifferential of f at X is

$$\partial_\varepsilon f(X) := \{V : f(U) \ge f(X) + \langle V, U - X\rangle - \varepsilon,\ \forall U \in \mathbb{E}\}.$$

For a nonempty closed convex set $\Omega \subseteq \mathbb{E}$, the Euclidean projection of $X \in \mathbb{E}$ onto Ω is

$$\mathrm{proj}_\Omega(X) := \operatorname*{arg\,min}_{Y \in \Omega} \|X - Y\| \tag{4}$$

and the distance from a point $q \in \mathbb{E}$ to Ω is $\mathrm{dist}(q, \Omega) := \inf\{\|q - s\| : s \in \Omega\}$.

Given a set $\Omega \subseteq \mathbb{E}$, a point $X_0 \in \Omega$, and a function $f : \Omega \to \mathbb{R}$, we define the sublevel set of f at X_0 and its radius by

$$\mathcal{S}(X_0; f) := \{X' \in \Omega : f(X') \le f(X_0)\}, \tag{5}$$
$$D(X_0; f) := \sup\{\|X'\| : X' \in \mathcal{S}(X_0; f)\}. \tag{6}$$

We next introduce notation related to a differentiable function $g : \mathbb{E} \to \mathbb{R}$. Its affine approximation at $\bar{z} \in \mathbb{E}$ is

$$\ell_g(z; \bar{z}) := g(\bar{z}) + \langle \nabla g(\bar{z}), z - \bar{z}\rangle \qquad \forall z \in \mathbb{E}. \tag{7}$$

Function g is said to have a locally Lipschitz continuous gradient if, for every $U \in \mathbb{E}$, there exist constants $R = R_U > 0$ and $\bar{L}_U > 0$ such that

$$\|\nabla g(U') - \nabla g(U'')\| \le \bar{L}_U \|U' - U''\|, \quad \forall U', U'' \in U + B(R) \tag{8}$$

where $B(R) := \{X \in \mathbb{E} : \|X\| \le R\}$. The Lipschitz constant of $\nabla g(\cdot)$ and the maximum of $\|\nabla g(\cdot)\|$ both relative to the ball $B(R)$ are denoted, respectively, by

$$[\mathrm{LIP}](R; g) := \inf\left\{L > 0 : \|\nabla g(X) - \nabla g(X')\| \le L\|X - X'\| \quad \forall X, X' \in B(R)\right\}, \tag{9}$$
$$\nabla(R; g) := \sup\{\|\nabla g(U)\| : U \in B(R)\}. \tag{10}$$

2 The SDP problem and an inexact AL framework

This section develops an inexact augmented Lagrangian (AL) framework for computing approximate primal-dual solutions of an SDP. The framework is independent of the particular method used to solve the AL subproblems. At each outer iteration, the inner solver need only return a PSD matrix satisfying prescribed approximate first-order optimality conditions for the current AL subproblem. The multiplier is then updated, and the penalty parameter is increased whenever the primal residual remains above the target tolerance. This separation allows the outer-iteration complexity of the AL framework to be analyzed independently of the specific inner solver used to satisfy the required inexactness conditions.

The section is organized as follows. Subsection 2.1 introduces the primal-dual SDP pair, discusses its assumptions, and specifies the approximate primal-dual solution concept targeted by the framework. Subsection 2.2 formally presents the inexact AL framework. Finally, Subsection 2.3 proves Theorem 2.4, which gives the main outer-iteration complexity bound for the framework.

2.1 The SDP problem and assumptions

We consider the primal-dual pair of SDPs

$$\min_{X \in \mathbb{S}^n_+} \{\langle C, X\rangle \ : \ \mathcal{A}X = b\} \tag{11}$$

and

$$\max \{\langle b, p\rangle : \ C - \mathcal{A}^* p \succeq 0, \quad p \in \mathbb{R}^m\}. \tag{12}$$

In contrast to the formulation in [44], which assumes a bounded feasible region, problem 11 is posed over the full PSD cone $\mathbb{S}^n_+$, without a constraint of the form $\mathrm{tr}(X) \leq \tau$. In the absence of such a bound, the analysis instead relies on properties of the SDP solution set.

We now state the assumptions that are made on the SDP data.

Assumption 2.1. *The set of optimal solutions of* (11) *is nonempty and bounded.*

Assumption 2.2. *There exists an optimal primal dual pair* $(X_*, p_*) \in \mathbb{S}^n_+ \times \mathbb{R}^m$ *satisfying*

$$0 \ \in \ C - \mathcal{A}^* p_* + \mathcal{N}_{\mathbb{S}^n_+}(X_*), \qquad \mathcal{A}X_* = b. \tag{13}$$

The definition of the normal cone in (3), applied with $\Omega = \mathbb{S}^n_+$ and $X = X_*$ implies that (13) is equivalent to

$$\mathcal{A}X_* = b, \qquad X_* \succeq 0, \qquad C - \mathcal{A}^* p_* \succeq 0, \qquad \langle C - \mathcal{A}^* p_*, X_*\rangle = 0. \tag{14}$$

Relation (14) expresses primal feasibility, dual feasibility, and complementarity. In particular, (X_*, p_*) is a primal-dual KKT pair (11)–(12), and the primal and dual problems have zero duality gap at this pair.

The AL framework will terminate when approximate counterparts of the three nontrivial conditions in (14) have been obtained. The next definition preciesely characterizes our notion of inexactness.

Definition 2.3. *For tolerances* $(\varepsilon_p, \varepsilon_c, \varepsilon_d) > 0$*, a pair* $(\bar{Z}, \bar{p}) \in \mathbb{S}^n_+ \times \mathbb{R}^m$ *is called an* $(\varepsilon_p, \varepsilon_c, \varepsilon_d)$*-solution of* (11)–(12) *if*

$$\|\mathcal{A}(\bar{Z}) - b\| \leq \varepsilon_p, \qquad \lambda_{\min}(\bar{S}) \geq -\varepsilon_d, \qquad \left|\langle \bar{S}, \bar{Z}\rangle\right| \leq \varepsilon_c, \tag{15}$$

where $\bar{S} := S(\bar{p})$ *and*

$$S(p) := C - \mathcal{A}^* p. \tag{16}$$

Definition 2.3 is a direct relaxation of the exact KKT conditions. Indeed, setting $\varepsilon_p = \varepsilon_c = \varepsilon_d = 0$ in (15) recovers (14).

2.2 AL Framework

This subsection presents the AL framework for approximately solving the primal-dual SDP pair (11)–(12). For ease of analysis and presentation, the AL framework is presented in the X-space, meaning that all iterates are full $n \times n$ positive semidefinite matrices. Our ultimate goal is to later present an algorithm that never works with such dense matrices X and instead stores only low-rank factors U, which typically have only a few columns. This memory-efficient low-rank algorithm, HALLaR+, is presented in Section 4 and can be viewed as a variant of the AL framework.

The AL framework is based on approximately solving a sequence of augmented Lagrangian subproblems (2). For convenience, we define the shorthand

$$r(X) := b - \mathcal{A}X \qquad \forall X \in \mathbb{S}^n. \tag{17}$$

Hence, for a multiplier $p \in \mathbb{R}^m$ and a penalty parameter $\beta \in \mathbb{R}_+$, the augmented Lagrangian function is defined as

$$\mathcal{L}_\beta(X; p) \ := \ \langle C, X\rangle + \langle p, r(X)\rangle + \frac{\beta}{2}\|r(X)\|^2. \tag{18}$$

Moreover, during the t-th iteration of the AL framework, the augmented Lagrangian subproblem (2a) can be written as

$$\min_{X \in \mathbb{S}^n_+} \mathcal{L}_{\beta_{t-1}}(X; p_{t-1}) \ = \ \min_{X \in \mathbb{S}^n_+} \left\{\langle C, X\rangle + \langle p_{t-1}, r(X)\rangle + \frac{\beta_{t-1}}{2}\|r(X)\|^2\right\}.$$

Since the AL objective in (2a) is convex and differentiable and the feasible region is the closed convex cone $\mathbb{S}^n_+$, its first-order optimality condition is

$$0 \in \nabla_X \mathcal{L}_{\beta_{t-1}}(X; p_{t-1}) + \mathcal{N}_{\mathbb{S}^n_+}(X). \tag{19}$$

By the normal-cone characterization in (3), condition (19) is equivalent to

$$\nabla_X \mathcal{L}_{\beta_{t-1}}(X; p_{t-1}) \succeq 0, \qquad \langle \nabla_X \mathcal{L}_{\beta_{t-1}}(X; p_{t-1}), X \rangle = 0. \tag{20}$$

The AL framework in this subsection approximately solves each AL subproblem based on stopping criteria that are direct relaxations of the optimality conditions in (20) (see (21) below). This differs from the HALLaR and CGAL methods in [44, 66], which assume a compact feasible region and approximately solve their AL subproblems by performing a prescribed number of Frank–Wolfe iterations or by terminating when the Frank–Wolfe gap is sufficiently small. These approaches are not directly applicable to (2a), whose feasible region is unbounded.

The AL framework is now presented.

AL Framework

Input: Initial pair $(Z_0, p_0) \in \mathbb{S}^n_+ \times \mathbb{R}^m$, tolerance triple $(\varepsilon_p, \varepsilon_c, \varepsilon_d) \in \mathbb{R}^3_{++}$, and initial penalty parameter $\beta_0 \in \mathbb{R}_{++}$.
Output: Pair $(\bar{Z}, \bar{p})$ satisfying (15).

0. Set $t \leftarrow 1$.

1. Find $Z_t \in \mathbb{S}^n_+$, which together with $S_t = C - \mathcal{A}^*[p_{t-1} + \beta_{t-1} r(Z_t)]$, satisfies

$$\lambda_{\min}(S_t) \geq -\varepsilon_d, \qquad |\langle S_t, Z_t \rangle| \leq \varepsilon_c. \tag{21}$$

2. Set $p_t = p_{t-1} + \beta_{t-1} r(Z_t)$.

3. **If** $\|r(Z_t)\| \leq \varepsilon_p$, **then stop** and output $(\bar{Z}, \bar{p}) \leftarrow (Z_t, p_t)$.

4. set $\beta_t \leftarrow 2\beta_{t-1}$, $t \leftarrow t + 1$, and go to **Step** 1.

First, Step **1** computes $Z_t \in \mathbb{S}^n_+$ satisfying an inexact optimality condition for the current AL subproblem. Indeed, the identity

$$\nabla_X \mathcal{L}_\beta(X; p) = C - \mathcal{A}^*[p + \beta r(X)] \tag{22}$$

implies that $\nabla_X \mathcal{L}_{\beta_{t-1}}(Z_t; p_{t-1}) = S_t$. Thus, the conditions in (20) are a special case of (21) with $\varepsilon_c = \varepsilon_d = 0$ The AL framework requires only that (21) hold and does not prescribe how such a point is computed. A Frank–Wolfe-type method for computing such a point is developed in later sections.

Second, Step **2** performs the standard AL multiplier update, which, together with the definition of $S(\cdot)$ in (16), implies that $S_t = S(p_t)$. The two conditions in (21) then ensure that (Z_t, p_t) satisfies the approximate dual-feasibility and complementarity requirements in (15). Third, because $Z_t \in \mathbb{S}^n_+$ by construction, Step **3** only needs to check primal feasibility. If $\|r(Z_t)\| \leq \varepsilon_p$, then (Z_t, p_t) satisfies all the conditions in (15) in view of (21). In this case (Z_t, p_t) is returned, otherwise, if $\|r(Z_t)\| > \varepsilon_p$, Step **4** doubles the penalty parameter and proceeds to the next iteration. Consequently, for every $t \geq 1$, the penalty parameters satisfy

$$\beta_{t-1} = 2^{t-1} \beta_0. \tag{23}$$

For the analysis, the same formula defines the penalty parameter at the terminal index.

Theorem 2.4 gives the main outer-iteration complexity bound for the AL framework. Its proof is presented in Subsection 2.3.

Theorem 2.4. *Let $\varepsilon := (\varepsilon_p, \varepsilon_c, \varepsilon_d) > 0$ be given and define*

$$C_1 = C_1(\varepsilon) := \frac{\varepsilon_c + \varepsilon_d}{\varepsilon_p}, \qquad \tau_* := \max\{1, 2\,\mathrm{tr}(X_*)\}, \qquad \bar{\beta}(\varepsilon) := \beta_0 + \frac{1}{\varepsilon_p}\left[4\left(\|p_0\| + \|p_*\|\right) + 16 C_1 \tau_*\right]. \tag{24}$$

Then the following statements about the AL framework hold:

i) it terminates in at most T iterations, where

$$T = T_\varepsilon := 1 + \left\lceil \log_2^+ \left(\frac{2\left(\|p_0\| + \|p_*\|\right) + 8 C_1 \tau_*}{\beta_0 \varepsilon_p} \right) \right\rceil; \tag{25}$$

ii) its last generated penalty parameter $\bar{\beta}$ is bounded by

$$\bar{\beta} \leq \bar{\beta}(\varepsilon) \tag{26}$$

where the bound $\bar{\beta}(\varepsilon)$ is defined in (24);

iii) its output pair $(\bar{Z}, \bar{p})$ is an $(\varepsilon_p, \varepsilon_c, \varepsilon_d)$-solution of (11)–(12), and every multiplier p_t generated by the AL framework satisfies

$$\|p_t\| \leq \|p_*\| + \sqrt{\|p_0 - p_*\|^2 + 4\varepsilon_p C_1 \bar{\beta}(\varepsilon)\tau_*}. \tag{27}$$

2.3 Proof of Theorem 2.4

Besides Assumption 2.2, the proof below uses only relation (21) and Steps **2**–**4** of the AL framework, and hence it does not depend on how Step **1** is implemented. For every $t \geq 1$, define the gap quantity

$$\vartheta_t^* \;:=\; \max_{U \in \Delta_{\tau_*}} \langle S_t, Z_t - U \rangle, \tag{28}$$

where Δ_τ is as in (100) and τ_* is as in (24). We first show that the inner product between the dual error $p_t - p_*$ and the primal residual $r(Z_t)$ is controlled by ϑ_t^*. This relation will be instrumental in bounding the sequence of primal residuals $\|r(Z_t)\|$.

Lemma 2.5. *For every $t \geq 1$, the iterate pair (Z_t, p_t) satisfies*

$$\langle p_t - p_*, r(Z_t) \rangle \;\leq\; \vartheta_t^*, \tag{29}$$

where ϑ_t^* is as in (28).

Proof. Using (28), the inclusion $X_* \in \Delta_{\tau_*}$, and the fact that the identity $p_t = p_{t-1} + \beta_{t-1} r(Z_t)$ and (22) imply that $S_t = C - \mathcal{A}^* p_t$, we have

$$\begin{aligned} \vartheta_t^* &\overset{(28)}{=} \max_{U \in \Delta_{\tau_*}} \langle S_t, Z_t - U \rangle \geq \; \langle S_t, Z_t - X_* \rangle \overset{(22)}{=} \langle C - \mathcal{A}^* p_t, Z_t - X_* \rangle \\ &= \langle \mathcal{A}^*(p_* - p_t), Z_t - X_* \rangle + \langle C - \mathcal{A}^* p_*, Z_t - X_* \rangle. \end{aligned} \tag{30}$$

Using (13) and the definition in (17), the first term in (30) satisfies

$$\begin{aligned} \langle \mathcal{A}^*(p_* - p_t), Z_t - X_* \rangle &= \; \langle p_* - p_t, \mathcal{A}(Z_t - X_*) \rangle \;\overset{(13)}{=}\; \langle p_* - p_t, \mathcal{A} Z_t - b \rangle \\ &= \; \langle p_t - p_*, b - \mathcal{A} Z_t \rangle \;\overset{(17)}{=}\; \langle p_t - p_*, r(Z_t) \rangle. \end{aligned} \tag{31}$$

It is then easy to see from (13) that $C - \mathcal{A}^* p_* \in -\mathcal{N}_{\mathbb{S}^n_+}(X_*)$. This inclusion, the fact that $Z_t \in \mathbb{S}^n_+$, and the definition of normal cone in (3) with $u = Z_t$, $v = -(C - \mathcal{A}^* p_*)$, $\Omega = \mathbb{S}^n_+$, and $x = X_*$ imply that

$$\langle C - \mathcal{A}^* p_*, Z_t - X_* \rangle \geq \; 0. \tag{32}$$

Relation (29) then follows immediately from relations (30), (31), and (32). □

The following lemma bounds the quantity ϑ_t^* by a fixed quantity that does not vary with the iteration index t.

Lemma 2.6. *For every $t \geq 1$, we have*

$$\vartheta_t^* \;\leq\; \varepsilon_c + \varepsilon_d \tau_*, \tag{33}$$

where ϑ_t^* is as in (28).

Proof. Definition (28) and Proposition A.2(a), applied with $(X, G, \tau) = (Z_t, S_t, \tau_*)$, imply that

$$\begin{aligned} \vartheta_t^* &\leq |\langle S_t, Z_t \rangle| + \max\{\lambda_{\max}(-S_t), 0\}\tau_* \\ &\leq \varepsilon_c + \varepsilon_d \tau_*, \end{aligned}$$

where the last inequality follows from (21) and the identity $\lambda_{\max}(-S_t) = -\lambda_{\min}(S_t)$. □

The next lemma establishes key bounds on the sequence of Lagrange multipliers and primal feasibility residuals.

Lemma 2.7. *For every generated iteration index $t \geq 1$, we have*

$$\|p_t - p_*\|^2 + \beta_{t-1}^2 \|r(Z_t)\|^2 \leq \|p_0 - p_*\|^2 + 4\varepsilon_p C_1 \beta_{t-1} \tau_*. \tag{34}$$

Proof. The multiplier update in Step **2** implies

$$\begin{aligned} \|p_t - p_*\|^2 + \beta_{t-1}^2 \|r(Z_t)\|^2 &= \|p_{t-1} - p_*\|^2 + 2\beta_{t-1} \langle p_t - p_*, r(Z_t) \rangle \\ &\leq \|p_{t-1} - p_*\|^2 + 2\varepsilon_p C_1 \beta_{t-1} \tau_*, \end{aligned} \tag{35}$$

where the inequality follows from (29), (33), and (24). For $t = 1$, relation (34) follows directly from (35). For $t \geq 2$, applying (34) at index $t-1$ to the right-hand side of (35) and using $\beta_{t-2} = \beta_{t-1}/2$ proves the induction step. □

Proof of Theorem 2.4. For convenience, we define

$$\Psi_* = \Psi_*(\varepsilon) := 2\left(\|p_0\| + \|p_*\|\right) + 8C_1\tau_*. \tag{36}$$

Using the fact that $y \leq 2^{\lceil \log_2^+ y \rceil} \leq 1 + 2y$ for every $y > 0$ with $y = \Psi_*/(\beta_0 \varepsilon_p)$, the definition of T in (25), relation (23), and the definition of $\bar{\beta}(\varepsilon)$ in (24), we obtain

$$\bar{\beta}(\varepsilon) \;=\; \beta_0 + \frac{2\Psi_*}{\varepsilon_p} \;\geq\; \beta_0 2^{T-1} \;=\; \beta_{T-1} \;\geq\; \frac{\Psi_*}{\varepsilon_p} \overset{(36)}{\geq} \frac{1}{\varepsilon_p} \max\left\{2\left(\|p_0\| + \|p_*\|\right),\; 8C_1\tau_*\right\}. \tag{37}$$

Suppose now, for the sake of contradiction, that the AL framework has not terminated at the end of iteration T, i.e., that $\|r(Z_T)\| > \varepsilon_p$ in Step **3**. Then, dropping the nonnegative term $\|p_T - p_*\|^2$ from the left-hand side of (34) with $t = T$ and dividing the resulting inequality by $\beta_{T-1}^2 \varepsilon_p^2$, we obtain

$$1 \;<\; \frac{\|r(Z_T)\|^2}{\varepsilon_p^2} \overset{(34)}{\leq} \frac{\|p_0 - p_*\|^2}{\varepsilon_p^2 \beta_{T-1}^2} + \frac{4C_1\tau_*}{\varepsilon_p \beta_{T-1}} \;\leq\; \frac{\left(\|p_0\| + \|p_*\|\right)^2}{\varepsilon_p^2 \beta_{T-1}^2} + \frac{4C_1\tau_*}{\varepsilon_p \beta_{T-1}} \overset{(37)}{\leq} \frac{1}{4} + \frac{1}{2} \;=\; \frac{3}{4}, \tag{38}$$

where the second inequality uses the triangle inequality and the last one bounds the two fractions by means of the first and second terms in the maximum in (37), respectively. This contradiction shows that the AL framework performs at most T outer iterations.

Let $\bar{t} \leq T$ be the terminal outer index, so that $(\bar{Z}, \bar{p}) = (Z_{\bar{t}}, p_{\bar{t}})$ and $\bar{\beta} = \beta_{\bar{t}-1}$. Since $\{\beta_t\}$ is nondecreasing by relation (23), we have $\bar{\beta} = \beta_{\bar{t}-1} \leq \beta_{T-1}$, and hence the upper bound in (37) yields (26). Moreover, it is easy to see that (22) implies that $\bar{S} = C - \mathcal{A}^*\bar{p} = S_{\bar{t}}$, and thus the two inequalities in (21) with $t = \bar{t}$ together with the stopping test in Step **3** give the three relations in (15). Hence, $(\bar{Z}, \bar{p})$ is an $(\varepsilon_p, \varepsilon_c, \varepsilon_d)$-solution of (11)–(12).

Finally, let $t \leq \bar{t}$ be any generated iteration index. Dropping the nonnegative term $\beta_{t-1}^2 \|r(Z_t)\|^2$ from the left-hand side of (34) and using the facts that $\beta_{t-1} \leq \beta_{\bar{t}-1} = \bar{\beta}$ and $\bar{\beta} \leq \bar{\beta}(\varepsilon)$, we conclude that

$$\|p_t - p_*\|^2 \;\overset{(34)}{\leq}\; \|p_0 - p_*\|^2 + 4\varepsilon_p C_1 \beta_{t-1} \tau_* \;\overset{(26)}{\leq}\; \|p_0 - p_*\|^2 + 4\varepsilon_p C_1 \bar{\beta}(\varepsilon) \tau_*.$$

Relation (27) now follows by taking square roots in the above inequality and using the triangle inequality $\|p_t\| \leq \|p_*\| + \|p_t - p_*\|$. □

3 X-AL Method

This section presents a concrete method for approximately solving the linearly constrained SDP (11) under Assumption 2.1, and establishes its total inner-iteration complexity. The method is a special instance of the AL framework of Section 2 in which the Relaxed Unbounded Frank–Wolfe routine of Appendix A.2 implements Step **1**. A feature of the resulting bound is that the number of inner iterations performed during an outer iteration does not depend on how suboptimal the warm start Z_{t-1} is for the current AL subproblem.

3.1 X-AL Method and Main Complexity Result

We now make Step **1** of the AL framework clear while keeping the description entirely in the X-space. At outer iteration $t \geq 1$, the X-space implementation of Step **1** applies the relaxed unbounded Frank–Wolfe framework (RUFW) of Appendix A.2 directly to the AL subproblem

$$\min\left\{\mathcal{L}_{\beta_{t-1}}(X; p_{t-1}) : X \in \mathbb{S}_+^n\right\}, \tag{39}$$

with initial point Z_{t-1} and tolerance pair $(\varepsilon_d, \varepsilon_c)$. The output returned by this call is denoted by the matrix Z_t.

We are now ready to present the X-Space AL Method.

X-AL Method

Input: Initial pair $(Z_0, p_0) \in \mathbb{S}^n_+ \times \mathbb{R}^m$, tolerances $(\varepsilon_p, \varepsilon_c, \varepsilon_d) \in \mathbb{R}^3_{++}$, and initial penalty parameter $\beta_0 \in \mathbb{R}_{++}$.
Output: Pair $(\bar{Z}, \bar{p})$ satisfying (15).

0. Set $t \leftarrow 1$.

1. Perform the following X-space relaxed unbounded Frank–Wolfe (RUFW) call:

1.a) Set $\tilde{X} \leftarrow Z_{t-1}$.

1.b) Compute $X \in \mathbb{S}^n_+$ such that

$$\mathcal{L}_{\beta_{t-1}}(X; p_{t-1}) \leq \mathcal{L}_{\beta_{t-1}}(\tilde{X}; p_{t-1}) \tag{40}$$

and set

$$S \leftarrow \nabla_X \mathcal{L}_{\beta_{t-1}}(X; p_{t-1}).$$

1.c) Compute a minimum eigenpair $(\lambda_{\min}(S), v_{\min}(S))$ of S with $\|v_{\min}(S)\| = 1$.

1.d) **If**

$$\lambda_{\min}(S) \geq -\varepsilon_d, \qquad |\langle S, X \rangle| \leq \varepsilon_c, \tag{41}$$

then set $Z_t \leftarrow X$ and go to Step **2**.

1.e) Set

$$y^F := \begin{cases} v_{\min}(S), & \lambda_{\min}(S) < 0, \\ 0, & \lambda_{\min}(S) \geq 0, \end{cases} \qquad X^F := y^F (y^F)^\top \tag{42}$$

and compute

$$(\alpha, \omega) \in \arg\min \left\{ \mathcal{L}_{\beta_{t-1}}(\alpha' X + \omega' X^F; p_{t-1}) : \alpha' \geq 0,\ \omega' \geq 0 \right\}. \tag{43}$$

1.f) Set

$$\tilde{X} \leftarrow \alpha X + \omega X^F, \tag{44}$$

and return to Step **1.b)**.

2. Set

$$p_t \leftarrow p_{t-1} + \beta_{t-1} r(Z_t), \qquad S_t \leftarrow C - \mathcal{A}^* p_t. \tag{45}$$

3. **If** $\|r(Z_t)\| \leq \varepsilon_p$, **then** set $(\bar{Z}, \bar{p}) \leftarrow (Z_t, p_t)$, output $(\bar{Z}, \bar{p})$, and **stop**.

4. Set

$$\beta_t \leftarrow 2\beta_{t-1}, \qquad t \leftarrow t + 1, \tag{46}$$

and return to Step **1**.

Step **1** approximately solves the current AL subproblem by means of an X-space relaxed unbounded Frank–Wolfe (RUFW) procedure described in Appendix A.2. At the beginning of the t-th outer iteration, Step **1.a)** initializes the RUFW iterate at Z_{t-1}. Given the current point $\widetilde{X}$, Step **1.b)** computes a point $X \in \mathbb{S}^n_+$ whose AL value is no larger than that of $\widetilde{X}$, and evaluates the corresponding gradient

$$S = \nabla_X \mathcal{L}_{\beta_{t-1}}(X; p_{t-1}).$$

Step **1.c)** then computes a minimum eigenpair of S, and Step **1.e)** forms the rank-one matrix $X^F = y^F (y^F)^\top$ associated with a direction of negative curvature of the linearized objective. Step **1.d)** tests whether X satisfies the given approximate optimality conditions (41). If so, the inner loop terminates and Z_t is set equal to X. Otherwise, Step **1.e)** minimizes the AL objective over nonnegative combinations of X and X^F, and Step **1.f)** uses the resulting point as the next RUFW iterate. The inner loop in Step **1** exits at Step **1.d)** precisely when the termination

criterion (41) holds, at which point the method sets $Z_t = X$ and proceeds to Step **2**. Hence, upon exiting Step **1**, $S = \nabla_X \mathcal{L}_{\beta_{t-1}}(Z_t; p_{t-1})$.

Step **2** then performs the standard AL multiplier update. Combining the fact that relations (22) and (45) imply that

$$\nabla_X \mathcal{L}_{\beta_{t-1}}(Z_t; p_{t-1}) = C - \mathcal{A}^*[p_{t-1} + \beta_{t-1} r(Z_t)] = C - \mathcal{A}^* p_t = S_t. \tag{47}$$

with the stopping conditions in (41) shows that the pair (Z_t, p_t) produced after Step **2** always satisfies (21). Hence, Step **1** implements the approximate AL-subproblem solve required in Step **1** of the AL framework presented in Subsection 2.2. The remaining steps carry out the corresponding operations of that framework. Step **2** updates the multiplier and defines S_t, Step **3** applies the primal-feasibility stopping test, and Step **4** increases the penalty parameter before the next outer iteration.

The following result provides an iteration-complexity bound for the X-AL method and will be proved in the next subsection.

Theorem 3.1. *Set $\varepsilon := (\varepsilon_p, \varepsilon_c, \varepsilon_d)$. Let $C_1 = C_1(\varepsilon)$ and τ_* be as in* (24)*, and let $T = T_\varepsilon$ and $\bar{\beta}(\varepsilon)$ be as in* (25) *and* (24)*, respectively. Define $\bar{\kappa}$ and $\bar{\tau}$ by*

$$\bar{\kappa} := \mathcal{L}_{\beta_0}(Z_0; p_0) + \frac{9\left(\|p_0\|^2 + \|p_*\|^2\right)}{\beta_0} + (6T+4)\varepsilon_p C_1 \tau_*. \tag{48}$$

$$\bar{\tau} := \max\left\{1, 2\sup\left\{\operatorname{tr}(Z) : Z \in \mathbb{S}^n_+, \langle C, Z\rangle + \frac{\beta_0}{4}\|r(Z)\|^2 \le \bar{\kappa}\right\}\right\}. \tag{49}$$

Then, the X-AL method returns a pair $(\bar{Z}, \bar{p})$ satisfying (15) *in at most T outer iterations and at most*

$$2T + 24\bar{\tau}^2\|\mathcal{A}\|^2\bar{\beta}(\varepsilon)\left(\frac{1}{\varepsilon_c} + \frac{2}{\varepsilon_d}\right) \tag{50}$$

inner iterations.

We now make two remarks about Theorem 3.1. First, the bound in (50) does not involve the objective gaps $\mathcal{L}_{\beta_{t-1}}(Z_{t-1}; p_{t-1}) - \min_{X \in \mathbb{S}^n_+} \mathcal{L}_{\beta_{t-1}}(X; p_{t-1})$ of the warm starts Z_{t-1} for the AL subproblems. This is a consequence of Proposition A.3(b), whose bound is independent of the initial objective gap because the first conic search of every RUFW call contains the full Frank–Wolfe step to the linear-oracle point over the trace ball Δ_{τ_t} (see Lemma A.7(a)). The initial pair (Z_0, p_0) enters (50) only through T, $\bar{\beta}(\varepsilon)$, and the radius $\bar{\tau}$. Second, suppose that $\varepsilon_p = \varepsilon_c = \varepsilon_d = \varepsilon \in (0, 1]$. Then $C_1 = 2$, $\bar{\beta}(\varepsilon) = \mathcal{O}(1/\varepsilon)$, and $T = \mathcal{O}(\log(1/\varepsilon))$. Moreover, since εT is bounded for $\varepsilon \in (0, 1]$, the scalar $\bar{\kappa}$ in (48) is bounded above by a constant that does not depend on ε, and hence so is $\bar{\tau}$ in view of (49) and Proposition A.1. Thus, for fixed problem data, initial pair, and β_0, the bound (50) is $\mathcal{O}(\varepsilon^{-2})$.

3.2 Proof of Theorem 3.1

The proof of Theorem 3.1 relies on the following lemma.

Lemma 3.2. *For every iteration index t generated by the X-space AL method, define*

$$\tau_t \;:=\; \max\left\{1,\; 2\sup\left\{\operatorname{tr}(X) : X \in \mathcal{S}\big(Z_{t-1}; \mathcal{L}_{\beta_{t-1}}(\cdot; p_{t-1})\big)\right\}\right\}. \tag{51}$$

Then the following statements about the t-th outer iteration hold:

(a) the quantity τ_t is finite;

(b) the RUFW call in Step **1** *terminates in at most*

$$\left\lfloor 2 + 12\tau_t^2\beta_{t-1}\|\mathcal{A}\|^2\left(\frac{1}{\varepsilon_c} + \frac{2}{\varepsilon_d}\right)\right\rfloor \tag{52}$$

inner iterations with a pair (Z_t, p_t) satisfying (21) *and the relation*

$$\mathcal{L}_{\beta_{t-1}}(Z_t; p_{t-1}) \;\le\; \mathcal{L}_{\beta_{t-1}}(Z_{t-1}; p_{t-1}). \tag{53}$$

Proof. (a) Fix a generated iteration index t. Definition (5) implies

$$\mathcal{S}\big(Z_{t-1}; \mathcal{L}_{\beta_{t-1}}(\cdot; p_{t-1})\big) = \big\{X \in \mathbb{S}^n_+ : \mathcal{L}_{\beta_{t-1}}(X; p_{t-1}) \le \mathcal{L}_{\beta_{t-1}}(Z_{t-1}; p_{t-1})\big\}.$$

Proposition A.1, applied with $(q, \beta, \kappa) = (p_{t-1}, \beta_{t-1}, \mathcal{L}_{\beta_{t-1}}(Z_{t-1}; p_{t-1}))$, shows that this set is bounded. It is nonempty because it contains Z_{t-1}. Hence, $\tau_t < \infty$ by (51).

(b) Relations (18) and (22) verify RUFW assumption **(A1)** with $L = \beta_{t-1}\|\mathcal{A}\|^2$, while part *(a)* verifies assumption **(A2)**. Apply Proposition A.3 to the objective $\mathcal{L}_{\beta_{t-1}}(\cdot; p_{t-1})$ with initial point Z_{t-1}, Lipschitz constant $\beta_{t-1}\|\mathcal{A}\|^2$, and tolerance pair $(\epsilon, \delta) = (\varepsilon_d, \varepsilon_c)$. Indeed, it follows from (116), (5), and (51) that $\bar{\tau} = \tau_t$, and hence the bound in (117) becomes the bound in (52). It then follows from these conclusions, Proposition A.3(a)-(b), the stopping criteria in (41), and relation (47) that RUFW terminates with a pair (Z_t, p_t) satisfying (21) and (53) in at most (52) inner iterations. □

The remainder of this subsection is dedicated to bounding the local radii τ_t uniformly. For every $\kappa \in \mathbb{R}$, define

$$\mathcal{P}_{\beta_0}(\kappa) \;:=\; \left\{U \in \mathbb{S}^n_+ : \langle C, U\rangle + \frac{\beta_0}{4}\|\mathcal{A}U - b\|^2 \le \kappa\right\}. \tag{54}$$

We first observe that Proposition A.1, applied with $(q, \beta) = (0, \beta_0/2)$, implies that $\mathcal{P}_{\beta_0}(\kappa)$ is bounded for every $\kappa \in \mathbb{R}$. Our strategy for showing that the local sublevel sets are uniformly bounded is to find $\bar{\kappa} \in \mathbb{R}$, independent of t, such that

$$\mathcal{S}\big(Z_{t-1}; \mathcal{L}_{\beta_{t-1}}(\cdot; p_{t-1})\big) \subseteq \mathcal{P}_{\beta_0}(\bar{\kappa})$$

for every generated iteration index t.

Lemma 3.3. *For every iteration index $t \ge 1$, we have:*

$$\frac{\|p_t\|^2}{\beta_t} \le \frac{3}{\beta_0}\left(\|p_0\|^2 + \|p_*\|^2\right) + 4\varepsilon_p C_1 \tau_*, \tag{55}$$

$$\mathcal{L}_{\beta_t}(Z_t; p_t) \le \mathcal{L}_{\beta_0}(Z_0; p_0) + \frac{3\|p_0 - p_*\|^2}{\beta_0} + 6t\varepsilon_p C_1 \tau_*. \tag{56}$$

where C_1 and τ_* are defined in (24).

Proof. The triangle inequality, (34), and $\beta_t = 2\beta_{t-1}$ imply

$$\begin{aligned}\frac{\|p_t\|^2}{\beta_t} &\le \frac{\|p_*\|^2 + \|p_t - p_*\|^2}{\beta_{t-1}} \\ &\overset{(34)}{\le} \frac{\|p_*\|^2 + \|p_0 - p_*\|^2}{\beta_{t-1}} + 4\varepsilon_p C_1 \tau_* \\ &\le \frac{3(\|p_0\|^2 + \|p_*\|^2)}{\beta_0} + 4\varepsilon_p C_1 \tau_*,\end{aligned}$$

where the last inequality uses $\beta_{t-1} \ge \beta_0$ and $\|p_0 - p_*\|^2 \le 2(\|p_0\|^2 + \|p_*\|^2)$. This proves (55).

Relations (53), (18), (45), and (46) also imply

$$\begin{aligned}\mathcal{L}_{\beta_t}(Z_t; p_t) &\le \mathcal{L}_{\beta_{t-1}}(Z_{t-1}; p_{t-1}) + \frac{3\beta_{t-1}}{2}\|r(Z_t)\|^2 \\ &\overset{(34)}{\le} \mathcal{L}_{\beta_{t-1}}(Z_{t-1}; p_{t-1}) + \frac{3\|p_0 - p_*\|^2}{2\beta_{t-1}} + 6\varepsilon_p C_1 \tau_*.\end{aligned} \tag{57}$$

Induction using (57) and (23) yields

$$\mathcal{L}_{\beta_t}(Z_t; p_t) \le \mathcal{L}_{\beta_0}(Z_0; p_0) + \frac{3\|p_0 - p_*\|^2}{\beta_0}(1 - 2^{-t}) + 6t\varepsilon_p C_1 \tau_*,$$

which implies (56). □

Lemma 3.4. *For every $t \ge 0$, we have*

$$\mathcal{S}(Z_t; \mathcal{L}_{\beta_t}(\cdot; p_t)) \subseteq \mathcal{P}_{\beta_0}(\bar{\kappa}), \tag{58}$$

where $\mathcal{S}(\cdot;\cdot)$ is as in (5)*, and $\bar{\kappa}$ is as in* (48).

Proof. Assume that $U \in \mathcal{S}(Z_t; \mathcal{L}_{\beta_t}(\cdot; p_t))$, which clearly implies that $\mathcal{L}_{\beta_t}(U; p_t) \leq \mathcal{L}_{\beta_t}(Z_t; p_t)$, in view of the definition of $\mathcal{S}(\cdot;\cdot)$ in (5). This observation together with (18), the fact that $\beta_t \geq \beta_0$, and Young's inequality, then imply that

$$\begin{aligned}\langle C, U\rangle + \frac{\beta_0}{4}\|r(U)\|^2 &\leq \langle C, U\rangle + \frac{\beta_t}{4}\|r(U)\|^2 \overset{(18)}{=} \mathcal{L}_{\beta_t}(U; p_t) - \langle p_t, r(U)\rangle - \frac{\beta_t}{4}\|r(U)\|^2 \\ &\leq \mathcal{L}_{\beta_t}(Z_t; p_t) - \langle p_t, r(U)\rangle - \frac{\beta_t}{4}\|r(U)\|^2 \leq \mathcal{L}_{\beta_t}(Z_t; p_t) + \frac{\|p_t\|^2}{\beta_t} \leq \bar{\kappa}, \end{aligned} \tag{59}$$

where the last inequality is due to (55), (56), and the definition of $\bar{\kappa}$ in (48). The conclusion of the lemma now follows from (54). □

Proof of Theorem 3.1. Lemma 3.2(b) implies that X-AL is an instance of the AL framework. Hence, Theorem 2.4 implies that it terminates at an outer iteration index $\bar{t} \leq T$ with a pair satisfying (15). For every $1 \leq t \leq \bar{t}$, Lemma 3.4 at index $t-1$ and definitions (51) and (49) imply $\tau_t \leq \bar{\tau}$. Thus, induction using (52) and $\beta_{t-1} = 2\beta_{t-2}$ shows that the first t outer iterations perform at most

$$2t + 12\bar{\tau}^2\|\mathcal{A}\|^2\left(\frac{1}{\varepsilon_c} + \frac{2}{\varepsilon_d}\right)(2\beta_{t-1} - \beta_0) \tag{60}$$

inner iterations. Applying (60) with $t = \bar{t}$ and using $\bar{t} \leq T$ and (26) yields (50). □

The following lemma bounds the total decrease of the AL objectives over the inner solves. It will be used to bound the ADAP-AIPP work of HALLaR+.

Lemma 3.5. *Suppose that the iterates generated by the AL framework satisfy* (53) *for every* $t \geq 1$. *Then the total decrease of the AL objectives over all inner solves is at most*

$$\bar{\kappa} + \|b\|\|p_*\|, \tag{61}$$

where $\bar{\kappa}$ *is as in* (48).

Proof. Let $t \geq 1$ be an iteration index. The multiplier update, penalty doubling, and (18) imply

$$\begin{aligned}\mathcal{L}_{\beta_t}(Z_t; p_t) + \frac{3\|p_t - p_*\|^2}{2\beta_t} &= \mathcal{L}_{\beta_{t-1}}(Z_t; p_{t-1}) + \frac{3\|p_{t-1} - p_*\|^2}{2\beta_{t-1}} \\ &\quad + 3\langle p_t - p_*, r(Z_t)\rangle - \frac{3\|p_t - p_*\|^2}{4\beta_{t-1}} \\ &\leq \mathcal{L}_{\beta_{t-1}}(Z_t; p_{t-1}) + \frac{3\|p_{t-1} - p_*\|^2}{2\beta_{t-1}} + 3\varepsilon_p C_1 \tau_*. \end{aligned} \tag{62}$$

The last inequality follows from (29), (33), and (24). Moreover, (14) and completion of the square imply

$$\mathcal{L}_{\beta_t}(Z_t; p_t) + \frac{3\|p_t - p_*\|^2}{2\beta_t} \geq \langle b, p_*\rangle + \frac{\|p_t - p_*\|^2}{\beta_t} \geq \langle b, p_*\rangle. \tag{63}$$

Applying (62) successively over the outer iterations and using (63) and $\bar{t} \leq T$ bounds the total decrease over the inner solves by

$$\mathcal{L}_{\beta_0}(Z_0; p_0) + \frac{3\|p_0 - p_*\|^2}{2\beta_0} + 3T\varepsilon_p C_1\tau_* - \langle b, p_*\rangle \overset{(48)}{\leq} \bar{\kappa} + \|b\|\|p_*\|,$$

where the last inequality uses $\|p_0 - p_*\|^2 \leq 2(\|p_0\|^2 + \|p_*\|^2)$ and Cauchy–Schwarz. □

4 HALLaR+ in the U-Space

This section presents HALLaR+, the factorized implementation of the X-AL method of Section 3. It contains two subsections. Subsection 4.1 states the HALLaR+ method and presents its main complexity result, namely Theorem 4.1. Subsection 4.2 is devoted to the proof of Theorem 4.1.

4.1 Description of HALLaR+ and Main Theorem

We start by stating the HALLaR+ method.

HALLaR+ Method

Input: Initial pair $(Y_0, p_0) \in \mathbb{R}^{n\times r_0} \times \mathbb{R}^m$ for some $r_0 \geq 1$, tolerance triple $(\varepsilon_p, \varepsilon_c, \varepsilon_d) \in \mathbb{R}^3_{++}$, initial penalty parameter $\beta_0 \in \mathbb{R}_{++}$, and hybrid low-rank (HLR) parameters $\eta_0 > 0$, $M_0 > 1/2$, and $\bar{\rho} > 0$.
Output: Pair $(\bar{Y}, \bar{p})$ such that $(\bar{Y}\bar{Y}^\top, \bar{p})$ satisfies (15).

0. Set $t \leftarrow 1$.

1. Perform the following U-space HLR (U-HLR) call to approximately minimize t-th AL subproblem:

1.a) Set $(\tilde{U}_0, \hat{r}_1) \leftarrow (Y_{t-1}, r_{t-1})$ and $k \leftarrow 1$.

1.b) Call ADAP-AIPP in Appendix B.2 with input

$$W_0 = \tilde{U}_{k-1}, \quad g(U) = \mathcal{L}_{\beta_{t-1}}(UU^\top; p_{t-1}), \quad \Omega = \mathbb{R}^{n\times \hat{r}_k}, \quad \eta_0 = \eta_0, \quad M_0 = M_0, \quad \bar{\rho} = \bar{\rho} \tag{64}$$

to obtain output pair $(U_k, R_k) \in \mathbb{R}^{n\times\hat{r}_k} \times \mathbb{R}^{n\times\hat{r}_k}$ and set

$$S_k \leftarrow C - \mathcal{A}^*[p_{t-1} + \beta_{t-1} r(U_k U_k^\top)].$$

1.c) Compute a minimum eigenpair $(\lambda_{\min}(S_k), v_{\min}(S_k))$ of S_k with $\|v_{\min}(S_k)\| = 1$, and set

$$y_k^F \;:=\; \begin{cases} v_{\min}(S_k), & \lambda_{\min}(S_k) < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{65}$$

1.d) **If**

$$\lambda_{\min}(S_k) \;\geq\; -\varepsilon_d, \qquad |\langle S_k U_k, U_k\rangle| \;\leq\; \varepsilon_c, \tag{66}$$

then **set** $(Y_t, r_t) \leftarrow (U_k, \hat{r}_k)$ and go to **Step 2**.

1.e) Compute

$$(\alpha_k, \omega_k) \in \arg\min \left\{ \mathcal{L}_{\beta_{t-1}}\left(\alpha U_k U_k^\top + \omega y_k^F (y_k^F)^\top; p_{t-1}\right) : \alpha \geq 0,\ \omega \geq 0 \right\} \tag{67}$$

and set

$$(\tilde{U}_k, \hat{r}_{k+1}) \;:=\; \begin{cases} (\sqrt{\omega_k}\, y_k^F,\ 1), & \alpha_k = 0, \\ \left(\left[\sqrt{\alpha_k}\, U_k,\ \sqrt{\omega_k}\, y_k^F\right],\ \hat{r}_k + 1\right), & \alpha_k > 0. \end{cases} \tag{68}$$

1.f) Set $k \leftarrow k + 1$ and return to Step **1.b)**.

2. Set

$$p_t \leftarrow p_{t-1} + \beta_{t-1} r(Y_t Y_t^\top). \tag{69}$$

3. **If** $\|r(Y_t Y_t^\top)\| \leq \varepsilon_p$, then **output** $(\bar{Y}, \bar{p}) \leftarrow (Y_t, p_t)$ and **stop**.

4. Set

$$\beta_t \leftarrow 2\beta_{t-1}, \qquad t \leftarrow t + 1, \tag{70}$$

and return to Step **1**.

Theorem 4.1. *Set $\varepsilon := (\varepsilon_p, \varepsilon_c, \varepsilon_d)$. Let C_1, τ_*, and $\bar{\beta}(\varepsilon)$ be as in* (24)*, let T be as in* (25)*, and let $\bar{\kappa}$ and $\bar{\tau}$ be as in* (48) *and* (49)*, respectively. Define*

$$\bar{C}(\varepsilon) := 24\bar{\beta}(\varepsilon)\|\mathcal{A}\|^2(3\sqrt{2\bar{\tau}} + 1)^2 + 8\left[\|C\| + \|\mathcal{A}\|\left(\|p_*\| + \sqrt{\|p_0 - p_*\|^2 + 4\varepsilon_p C_1 \bar{\beta}(\varepsilon)\tau_*} + \bar{\beta}(\varepsilon)\|b\|\right)\right]. \tag{71}$$

The following statements about HALLaR+ hold:

(a) it returns a pair $(\bar{Y}, \bar{p})$ such that $(\bar{Y}\bar{Y}^\top, \bar{p})$ is an $(\varepsilon_p, \varepsilon_c, \varepsilon_d)$-solution of (11)–(12) in at most T outer iterations and at most

$$2T + 24\bar{\tau}^2\|\mathcal{A}\|^2\bar{\beta}(\varepsilon)\left(\frac{1}{\varepsilon_c} + \frac{2}{\varepsilon_d}\right) \tag{72}$$

inner iterations/minimum eigenvector (MEV) computations;

(b) the total number of ADAP-AIPP iterations/ADAP-FISTA calls it performs in its Step **1.b)** *is at most*

$$\left(1 + \left\lceil \log_2^+\left(\eta_0\bar{C}(\varepsilon)\right)\right\rceil\right)\left[2T + 24\bar{\tau}^2\|\mathcal{A}\|^2\bar{\beta}(\varepsilon)\left(\frac{1}{\varepsilon_c} + \frac{2}{\varepsilon_d}\right)\right] + \frac{C_\sigma \max\left\{\eta_0^{-1}, \bar{C}(\varepsilon)\right\}}{\bar{\rho}^2}\left(\bar{\kappa} + \|b\|\|p_*\|\right), \tag{73}$$

where η_0 and $\bar{\rho}$ are inputs to HALLaR+ and C_σ is as in (175).

4.2 Proof of Theorem 4.1

We first state a key proposition, without proof, which will be instrumental in proving Theorem 4.1. Due to the length of its proof, we defer the proof to Subsection 4.2.1.

Proposition 4.2. *Consider the t-th iteration of HALLaR+ and let*

$$C(\tau_t, \beta_{t-1}) := 24\beta_{t-1}\|\mathcal{A}\|^2(3\sqrt{2\tau_t} + 1)^2 + 8\|C - \mathcal{A}^*(p_{t-1} + \beta_{t-1}b)\| \tag{74}$$

where τ_t is as in (51) *with $Z_{t-1} := Y_{t-1}Y_{t-1}^\top$. The following statements about the U-HLR call in Step* **1** *hold:*

(a) it performs at most

$$\left\lfloor 2 + 12\tau_t^2\beta_{t-1}\|\mathcal{A}\|^2\left(\frac{1}{\varepsilon_c} + \frac{2}{\varepsilon_d}\right)\right\rfloor \tag{75}$$

iterations/minimum eigenvector (MEV) computations to find a pair (Y_t, r_t) satisfying

$$\lambda_{\min}\left(\nabla\mathcal{L}_{\beta_{t-1}}(Y_tY_t^\top; p_{t-1})\right) \geq -\varepsilon_d, \qquad \left|\left\langle\nabla\mathcal{L}_{\beta_{t-1}}(Y_tY_t^\top; p_{t-1})Y_t, Y_t\right\rangle\right| \leq \varepsilon_c, \tag{76}$$

$$\mathcal{L}_{\beta_{t-1}}(Y_tY_t^\top; p_{t-1}) \leq \mathcal{L}_{\beta_{t-1}}(Y_{t-1}Y_{t-1}^\top; p_{t-1}), \tag{77}$$

where $(\varepsilon_c, \varepsilon_d)$ are input tolerances;

(b) the total number of ADAP-AIPP iterations/ADAP-FISTA calls it performs is at most

$$\left(1 + \left\lceil \log_2^+\left(\eta_0 C(\tau_t, \beta_{t-1})\right)\right\rceil\right)\left(2 + 12\tau_t^2\beta_{t-1}\|\mathcal{A}\|^2\left(\frac{1}{\varepsilon_c} + \frac{2}{\varepsilon_d}\right)\right)$$
$$+ \frac{C_\sigma \max\left\{\eta_0^{-1}, C(\tau_t, \beta_{t-1})\right\}}{\bar{\rho}^2}\left[\mathcal{L}_{\beta_{t-1}}(Y_{t-1}Y_{t-1}^\top; p_{t-1}) - \mathcal{L}_{\beta_{t-1}}(Y_tY_t^\top; p_{t-1})\right]$$

where η_0 and $\bar{\rho}$ are inputs to HALLaR+ and C_σ and $C(\tau_t, \beta_{t-1})$ are as in (175) *and* (74)*, respectively.*

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. For every generated iteration index $t \geq 0$, set $Z_t := Y_tY_t^\top$.

(a) Let $t \geq 1$ be a generated iteration index. Proposition 4.2(a) implies that the U-HLR call in Step **1** terminates with a matrix Y_t satisfying (76) and (77). Moreover, relations (22) and (69) imply that

$$\nabla_X\mathcal{L}_{\beta_{t-1}}(Z_t; p_{t-1}) = C - \mathcal{A}^*\left[p_{t-1} + \beta_{t-1}r(Z_t)\right] = C - \mathcal{A}^*p_t = S_t,$$

and the identity $Z_t = Y_tY_t^\top$ implies that $\langle S_tY_t, Y_t\rangle = \langle S_t, Z_t\rangle$. Hence, (76) implies that (Z_t, p_t) satisfies (21), and (77) is precisely (53). Thus, Step **1** of HALLaR+ implements Step **1** of the AL framework, and Steps **2**–**4** of HALLaR+ coincide with Steps **2**–**4** of the AL framework with $Z_t = Y_tY_t^\top$. It then follows from Theorem 2.4 that HALLaR+ terminates at an outer iteration index $\bar{t} \leq T$ with a pair $(\bar{Y}, \bar{p})$ such that $(\bar{Y}\bar{Y}^\top, \bar{p})$ is an $(\varepsilon_p, \varepsilon_c, \varepsilon_d)$-solution of (11)–(12), that $\bar{\beta} = \beta_{\bar{t}-1} \leq \bar{\beta}(\varepsilon)$, and that every multiplier p_t with $t \geq 1$ satisfies (27).

The proofs of Lemmas 3.3 and 3.4 use only (21), (53), and the outer updates. Hence, they apply with $Z_t = Y_tY_t^\top$ and imply $\tau_t \leq \bar{\tau}$. Applying the induction used in (60) to (75) and using $\bar{t} \leq T$ and (26) yields (72). Since every inner iteration performs one MEV computation, this proves (a).

(b) For every $1 \le t \le \bar{t}$, the triangle inequality, (27), and (26) imply

$$\begin{aligned}\|C - \mathcal{A}^*(p_{t-1} + \beta_{t-1}b)\| &\le \|C\| + \|\mathcal{A}\|\left(\|p_{t-1}\| + \beta_{t-1}\|b\|\right)\\ &\le \|C\| + \|\mathcal{A}\|\left(\|p_*\| + \sqrt{\|p_0 - p_*\|^2 + 4\varepsilon_p C_1 \bar{\beta}(\varepsilon)\tau_*} + \bar{\beta}(\varepsilon)\|b\|\right),\end{aligned}$$

where, for $t = 1$, the last inequality uses $\|p_0\| \le \|p_*\| + \|p_0 - p_*\|$. Together with $\tau_t \le \bar{\tau}$ and definitions (71) and (74), this implies

$$C(\tau_t, \beta_{t-1}) \le \bar{C}(\varepsilon). \tag{78}$$

Lemma 3.5, applied with $Z_t = Y_t Y_t^\top$, bounds the total AL decrease by (61). Combining this bound with Proposition 4.2(b), (78), and part (a) yields (73). □

4.2.1 Proof of Proposition 4.2

This subsection is devoted to proving Proposition 4.2, which establishes the iteration complexity of the U-HLR call performed in Step **1** of the t-th iteration of HALLaR+. Accordingly, throughout this subsection, we fix an arbitrary iteration index $t \ge 1$ and consider just the t-th iteration of HALLaR+. We now define several quantities which arise in the technical complexity analysis of the U-HLR call. Let

$$\phi_t(\cdot) := \mathcal{L}_{\beta_{t-1}}(\cdot; p_{t-1}), \quad g_t(U) := \phi_t(UU^\top). \tag{79}$$

For every iteration index $k \ge 1$ of the U-HLR call in Step 1, define

$$\widetilde{\mathcal{D}}_k := 6D(\tilde{U}_{k-1}; g_t) + 1, \quad \widetilde{(\mathrm{Lip})}_k := [\mathrm{LIP}](\widetilde{\mathcal{D}}_k; g_t), \tag{80}$$

$$\widetilde{M}_k := 24\beta_{t-1}\|\mathcal{A}\|^2\widetilde{\mathcal{D}}_k^2 + 8\|C - \mathcal{A}^*(p_{t-1} + \beta_{t-1}b)\| \tag{81}$$

where $D(\cdot;\cdot)$ is as in (6), and $[\mathrm{LIP}](\cdot;\cdot)$ is as in (9).

We now start off with a lemma that shows that the call to ADAP-AIPP in Step **1.b)** is made with inputs that satisfy assumptions **(C1)**, **(C2)**, and **(C4)** in Appendix B.2. It is easy to see that assumption **(C3)** is always satisfied.

Lemma 4.3. *Let $k \ge 1$ and consider the k-th iteration of the U-HLR call in Step **1**. The input of the ADAP-AIPP call made in Step **1.b)** of U-HLR satisfies*

(a) the function $g_t(U) = \phi_t(UU^\top)$ is differentiable with locally Lipschitz continuous gradient. Moreover,

$$\widetilde{(\mathrm{Lip})}_k \le 6\beta_{t-1}\|\mathcal{A}\|^2\widetilde{\mathcal{D}}_k^2 + 2\|C - \mathcal{A}^*(p_{t-1} + \beta_{t-1}b)\|; \tag{82}$$

(b) $g_t(U)$ is bounded below on $\mathbb{R}^{n\times\hat{r}_k}$;

(c) the sublevel set $\mathcal{S}(\tilde{U}_{k-1}; g_t)$ has finite diameter $D(\tilde{U}_{k-1}; g_t)$, where $\mathcal{S}(\cdot;\cdot)$ and $D(\cdot;\cdot)$ are as in (5) and (6), respectively.

Proof. (a) It is easy to see that $g_t(U) = \phi_t(UU^\top)$ is differentiable. To show $g_t(U)$ has locally Lipschitz continuous gradient, let $R > 0$ be an arbitrary scalar and $B[R]$ be a ball centered at 0 with radius R. It suffices to show that $g_t(U)$ has Lipschitz continuous gradient on $B[R]$. Thus, let U and V be in $B[R]$. First, it is easy to see that

$$\begin{aligned}\|UU^\top - VV^\top\| &= \|U(U-V)^\top - (V-U)V^\top\|\\ &\le \|U\|\|U-V\| + \|V\|\|U-V\| \le 2R\|U-V\|.\end{aligned} \tag{83}$$

Moreover, it follows from the fact that $\phi_t(\cdot)$ is $(\beta_{t-1}\|\mathcal{A}\|^2)$-smooth that

$$\begin{aligned}\|\nabla\phi_t(VV^\top)\| = \|\nabla\phi_t(VV^\top) - \nabla\phi_t(0) + \nabla\phi_t(0)\| &\le \beta_{t-1}\|\mathcal{A}\|^2\|VV^\top\| + \|\nabla\phi_t(0)\|\\ &\le \beta_{t-1}\|\mathcal{A}\|^2R^2 + \|\nabla\phi_t(0)\|.\end{aligned} \tag{84}$$

It then follows from relations (83) and (84), the fact that ϕ_t is $(\beta_{t-1}\|\mathcal{A}\|^2)$-smooth, and the fact that $\nabla g_t(U) = 2\nabla\phi_t(UU^\top)U$ that

$$\|\nabla g_t(U) - \nabla g_t(V)\| \le \|2\nabla\phi_t(UU^\top)U - 2\nabla\phi_t(VV^\top)U + 2\nabla\phi_t(VV^\top)U - 2\nabla\phi_t(VV^\top)V\|$$

$$
\begin{aligned}
&\le 2\|U\|\|\nabla\phi_t(UU^\top) - \nabla\phi_t(VV^\top)\| + 2\|\nabla\phi_t(VV^\top)\|\|U-V\| \\
&\le 2\beta_{t-1}\|\mathcal{A}\|^2\|U\|\|UU^\top - VV^\top\| + 2\|\nabla\phi_t(VV^\top)\|\|U-V\| \\
&\overset{(83),(84)}{\le} 4\beta_{t-1}\|\mathcal{A}\|^2R^2\|U-V\| + 2(\beta_{t-1}\|\mathcal{A}\|^2R^2 + \|\nabla\phi_t(0)\|)\|U-V\| \\
&= (6\beta_{t-1}\|\mathcal{A}\|^2R^2 + 2\|\nabla\phi_t(0)\|)\|U-V\|. \qquad (85)
\end{aligned}
$$

Hence, it follows that $\nabla g_t(U)$ is $(6\beta_{t-1}\|\mathcal{A}\|^2R^2 + 2\|\nabla\phi_t(0)\|)$-Lipschitz continuous on $B[R]$. Since R was arbitrary, it follows that $\nabla g_t(U)$ is locally Lipschitz continuous. Moreover, it follows from the definitions of $\widetilde{(\mathrm{Lip})}_k$ and $[\mathrm{LIP}](\cdot;\cdot)$ in (80) and (9), respectively, and replacing R in (85) with $\widetilde{\mathcal{D}}_k$ that $\widetilde{(\mathrm{Lip})}_k \le 6\beta_{t-1}\|\mathcal{A}\|^2\widetilde{\mathcal{D}}_k^2 + 2\|\nabla\phi_t(0)\|$. The bound in (82) then follows from this conclusion and the fact that the definition of ϕ_t in (79) implies that $\nabla\phi_t(0) = C - \mathcal{A}^*(p_{t-1} + \beta_{t-1}b)$.

(b) It is easy to see that

$$
\inf_{U\in\mathbb{R}^{n\times\hat{r}_k}} g_t(U) = \inf_{U\in\mathbb{R}^{n\times\hat{r}_k}} \phi_t(UU^\top) = \inf_{X\succeq 0,\,\mathrm{rank}(X)\le\hat{r}_k} \phi_t(X) \ge \inf_{X\succeq 0} \phi_t(X) > -\infty. \qquad (86)
$$

Hence, it follows from (86) that g_t is bounded below on $\mathbb{R}^{n\times\hat{r}_k}$.

(c) Let $U \in \mathcal{S}(\tilde{U}_{k-1}; g_t)$. It follows by the definition of $\mathcal{S}(\cdot;\cdot)$ in (5) that $g_t(U) \le g_t(\tilde{U}_{k-1})$, which by the definition of g_t in (79) implies that $\phi_t(UU^\top) \le \phi_t(\tilde{U}_{k-1}\tilde{U}_{k-1}^\top)$. Thus, $UU^\top \in \mathcal{S}(\tilde{U}_{k-1}\tilde{U}_{k-1}^\top; \phi_t)$. Now observe that it follows from Proposition A.1 and the definition of ϕ_t in (79) that the sublevel sets of ϕ_t are bounded. Hence $\mathcal{S}(\tilde{U}_{k-1}\tilde{U}_{k-1}^\top; \phi_t)$ is bounded. Thus it follows that $\|UU^\top\| \le M$ for some scalar $M > 0$. Now observe from the Cauchy-Schwarz inequality that

$$
\|U\|^2 = \langle U, U\rangle = \langle I_n, UU^\top\rangle \le \sqrt{n}\|UU^\top\| \le \sqrt{n}M.
$$

Thus, we have $\|U\| \le n^{1/4}\sqrt{M}$. Since U was arbitrary, it follows that $\mathcal{S}(\tilde{U}_{k-1}; g_t)$ is bounded with finite diameter $D(\tilde{U}_{k-1}; g_t)$. □

The following lemma establishes properties of the ADAP-AIPP call made in Step **1.b)**.

Lemma 4.4. *Consider the k-th iteration of the U-HLR call in Step 1. The following statements about the ADAP-AIPP call in Step **1.b)** of U-HLR hold:*

(a) it terminates with a point $U_k \in \mathbb{R}^{n\times\hat{r}_k}$ satisfying

$$
R_k = 2\nabla\phi_t(U_kU_k^\top)U_k, \quad \|R_k\| \le \bar{\rho} \qquad (87)
$$

$$
\phi_t(U_kU_k^\top) \le \phi_t(\tilde{U}_{k-1}\tilde{U}_{k-1}^\top); \qquad (88)
$$

(b) its total number of ADAP-AIPP iterations/ADAP-FISTA calls performed within the ADAP-AIPP call is bounded by

$$
1 + \frac{C_\sigma \max\left\{\eta_0^{-1}, \widetilde{M}_k\right\}}{\bar{\rho}^2}\left[\phi_t(\tilde{U}_{k-1}\tilde{U}_{k-1}^\top) - \phi_t(U_kU_k^\top)\right] + \left\lceil \log_2^+\left(\eta_0\widetilde{M}_k\right)\right\rceil
$$

where C_σ is as in (175), $\widetilde{M}_k$ is as in (81), and η_0 and $\bar{\rho}$ are inputs to ADAP-AIPP.

Proof. (a) Let $k \ge 1$ and consider the k-th iteration of the U-HLR call performed in Step **1**. By Lemma 4.3(a)–(c), and because Step **1.b)** calls ADAP-AIPP with $\Omega = \mathbb{R}^{n\times\hat{r}_k}$, a closed convex set with easy projection, the inputs of this call satisfy assumptions **(C1)**–**(C4)** of Appendix B.2. It then follows from Proposition B.13(a), the fact that the call to ADAP-AIPP is made with initial point $W_0 = \tilde{U}_{k-1}$ and function $g_t(U) = \phi_t(UU^\top)$ as in (79), and the fact that ADAP-AIPP terminates with $(\bar{W}, \bar{R}) = (U_k, R_k)$ that the pair (U_k, R_k) satisfies

$$
R_k \in \nabla g_t(U_k) + \mathcal{N}_{\mathbb{R}^{n\times\hat{r}_k}}(U_k), \qquad \|R_k\| \le \bar{\rho} \qquad (89)
$$

$$
\phi_t(U_kU_k^\top) \le \phi_t(\tilde{U}_{k-1}\tilde{U}_{k-1}^\top). \qquad (90)
$$

Relations (87) and (88) then follow immediately from (89), (90), and the facts that $\nabla g_t(U_k) = 2\nabla\phi_t(U_kU_k^\top)U_k$ and $\mathcal{N}_{\mathbb{R}^{n\times\hat{r}_k}}(U_k) = \{0\}$.

(b) Let $k \ge 1$ and consider the k-th iteration of the U-HLR call performed in Step **1** of HALLaR+. It follows from Lemma 4.3 that the input of the ADAP-AIPP call satisfies assumptions **(C1)-(C4)** in Appendix B.2. It then

follows from this observation, Proposition B.13(b), and the definition of $\widetilde{(\mathrm{Lip})}_k$ in (80) that the total number of ADAP-AIPP iterations/ADAP-FISTA calls performed within the ADAP-AIPP call is upper bounded by

$$1 + \frac{C_\sigma \max\left\{\eta_0^{-1},\, 4\widetilde{(\mathrm{Lip})}_k\right\}}{\bar{\rho}^2}\left[\phi_t(\tilde{U}_{k-1}\tilde{U}_{k-1}^\top) - \phi_t(U_k U_k^\top)\right] + \left\lceil \log_2^+\left(4\eta_0\widetilde{(\mathrm{Lip})}_k\right)\right\rceil.$$

The complexity bound in part (b) then follows immediately from this bound and the fact that the bound on $\widetilde{(\mathrm{Lip})}_k$ in Lemma 4.3(a) and the definition of $\widetilde{M}_k$ in (81) imply that $4\widetilde{(\mathrm{Lip})}_k \le \widetilde{M}_k$. □

Lemma 4.5. *The following statements about the U-HLR call performed in Step **1** hold:*

(a) it is an instance of the RUFW framework of Appendix A.2 applied to

$$\min_{X\in\mathbb{S}^n_+} \phi_t(X) \tag{91}$$

with initial point $\tilde{X}_0 = Y_{t-1}Y_{t-1}^\top$ and tolerance pair $(\epsilon,\delta) = (\varepsilon_d, \varepsilon_c)$, where ϕ_t is as in (79);

*(b) for every iteration index $k \ge 1$ at which Step **1.e)** is performed, it holds that*

$$\phi_t(\tilde{U}_k\tilde{U}_k^\top) \le \phi_t(U_kU_k^\top) \le \phi_t(\tilde{U}_{k-1}\tilde{U}_{k-1}^\top). \tag{92}$$

Hence, it follows that for every iteration index $k \ge 1$, $\phi_t(\tilde{U}_{k-1}\tilde{U}_{k-1}^\top) \le \phi_t(\tilde{U}_0\tilde{U}_0^\top)$.

Proof. (a) Let $k \ge 1$ be a generated iteration index of the U-HLR call and associate with the U-HLR iterates the matrices

$$X_k \;:=\; U_kU_k^\top, \qquad X_k^F \;:=\; y_k^F(y_k^F)^\top. \tag{93}$$

Set $\tilde{X}_0 := \tilde{U}_0\tilde{U}_0^\top = Y_{t-1}Y_{t-1}^\top$, where the last identity is due to Step **1.a)**, and whenever Step **1.e)** is executed, also define $\tilde{X}_k := \tilde{U}_k\tilde{U}_k^\top$. Relation (88) gives $\phi_t(X_k) \le \phi_t(\tilde{X}_{k-1})$, which is (111). Moreover, $S_k = C - \mathcal{A}^*(p_{t-1} + \beta_{t-1}r(X_k)) = \nabla_X\phi_t(X_k)$ and $\langle S_kU_k, U_k\rangle = \langle S_k, X_k\rangle$. Thus, Steps **1.c)**–**1.d)** of the U-HLR call coincide with Steps 2 and 3 of the RUFW framework with $G_k = S_k$ and tolerance pair $(\epsilon,\delta) = (\varepsilon_d,\varepsilon_c)$. Relation (67) is the RUFW conic search (114), and direct multiplication in (68) yields

$$\tilde{X}_k = \alpha_kX_k + \omega_kX_k^F,$$

which is (115). Hence the U-HLR call is an instance of the RUFW framework applied to ϕ_t with initial point $\tilde{X}_0 = Y_{t-1}Y_{t-1}^\top$ and tolerance pair $(\epsilon,\delta) = (\varepsilon_d,\varepsilon_c)$.

(b) By part (a), Lemma A.4 applies to the iterates $X_k = U_kU_k^\top$, $X_k^F = y_k^F(y_k^F)^\top$, and $\tilde{X}_k = \tilde{U}_k\tilde{U}_k^\top$. Thus, (118) gives $\phi_t(\tilde{U}_k\tilde{U}_k^\top) \le \phi_t(U_kU_k^\top) \le \phi_t(\tilde{U}_{k-1}\tilde{U}_{k-1}^\top)$, which is (92). Applying these relations recursively implies the last conclusion of the lemma. □

Lemma 4.6. *For any iteration index $k \ge 1$ of the U-HLR call in Step 1, it holds that*

$$\widetilde{\mathcal{M}}_k \le C(\tau_t, \beta_{t-1}) \tag{94}$$

where $C(\tau_t,\beta_{t-1})$ and $\widetilde{M}_k$ are as in (74) and (81), respectively.

Proof. Let $k \ge 1$ be an iteration index of the U-HLR call in Step 1. We claim that

$$\widetilde{\mathcal{D}}_k \le 3\sqrt{2\tau_t} + 1 \tag{95}$$

where τ_t and $\widetilde{\mathcal{D}}_k$ are as in (51) and (80), respectively. Hence, consider $U \in \mathcal{S}(\tilde{U}_{k-1}; g_t)$ where $\mathcal{S}(\cdot;\cdot)$ is as in (5). It then follows from this inclusion, the definition of g_t in (79), the last conclusion of Lemma 4.5(b), and the fact that $\tilde{U}_0 = Y_{t-1}$ in Step 1.a) that

$$\phi_t(UU^\top) \le \phi_t(\tilde{U}_{k-1}\tilde{U}_{k-1}^\top) \le \phi_t(\tilde{U}_0\tilde{U}_0^\top) = \phi_t(Y_{t-1}Y_{t-1}^\top)$$

which implies that $UU^\top \in \mathcal{S}(Z_{t-1};\phi_t)$ with $Z_{t-1} := Y_{t-1}Y_{t-1}^\top$. It then follows from this inclusion and the definition of τ_t in (51) that $\|U\|^2 = \mathrm{tr}(UU^\top) \le \tau_t/2$. This bound together with the definition of $\widetilde{\mathcal{D}}_k$ in (80) gives

$$\widetilde{\mathcal{D}}_k \le 6\sqrt{\tau_t/2} + 1 = 3\sqrt{2\tau_t} + 1$$

which proves (95). It then follows from (95) and the definitions of $C(\tau_t,\beta_{t-1})$ and $\widetilde{M}_k$ in (74) and (81), respectively, that relation (94) must hold. □

We are now ready to prove Proposition 4.2.

Proof of Proposition 4.2. (a) It follows from Lemma 4.5(a) that the U-HLR call in Step 1 is an instance of the RUFW framework applied to ϕ_t with initial point $\tilde{X}_0 = Y_{t-1}Y_{t-1}^\top = Z_{t-1}$ and tolerance pair $(\epsilon, \delta) = (\varepsilon_d, \varepsilon_c)$. Relations (18) and (22) imply that ϕ_t satisfies RUFW assumption **(A1)** with $L = \beta_{t-1}\|\mathcal{A}\|^2$, while Proposition A.1 implies that ϕ_t satisfies RUFW assumption **(A2)**. Moreover, the definitions of $\bar{\tau}$ in (116) and τ_t in (51), together with (5), imply that $\bar{\tau} = \tau_t$. It then follows from Proposition A.3(b) with $L = \beta_{t-1}\|\mathcal{A}\|^2$ and $\bar{\tau} = \tau_t$ that the U-HLR call terminates in at most (75) iterations with an iterate $X_k = U_k U_k^\top$ satisfying (109) and (110) with $\phi = \phi_t$ and $\tilde{X}_0 = Y_{t-1}Y_{t-1}^\top$. Since Step **1.d)** sets $Y_t = U_k$, and since $S_k = \nabla\phi_t(U_kU_k^\top)$ and $\langle S_kU_k, U_k\rangle = \langle S_k, U_kU_k^\top\rangle$, relations (109) and (110) are precisely (76) and (77). Statement (a) now follows from the fact that every iteration of the U-HLR call performs exactly one MEV computation in Step **1.c)**.

(b) Let $k \geq 1$ be an iteration index of the U-HLR call in Step 1. It follows from the complexity bound in Lemma 4.4(b) and the bound on $\widetilde{M}_k$ in (94) that the ADAP-AIPP call in Step **1.b)** of the k-th iteration of U-HLR performs at most

$$1 + \frac{C_\sigma \max\left\{\eta_0^{-1},\, C(\tau_t, \beta_{t-1})\right\}}{\bar{\rho}^2}\left[\phi_t(\tilde{U}_{k-1}\tilde{U}_{k-1}^\top) - \phi_t(U_kU_k^\top)\right] + \left\lceil \log_2^+\left(\eta_0 C(\tau_t, \beta_{t-1})\right)\right\rceil \tag{96}$$

iterations/ADAP-FISTA calls.

Let $\hat{k}$ denote the final iteration index of the U-HLR call. Relation (92) implies that the total decrease produced by its ADAP-AIPP calls is at most

$$\phi_t(Y_{t-1}Y_{t-1}^\top) - \phi_t(Y_tY_t^\top).$$

Combining this bound with (96), (75), and the definition of ϕ_t in (79) proves (b). □

5 Numerical Results

In this section we report computational results for cuHALLaR+, together with comparisons with existing SDP solvers. The problem classes are maximum stable set, matrix completion, quadratic assignment problem (QAP), and general SDPs from the Mittelmann benchmark collection [43]. We first describe the implementations and the comparison criteria.

Implementation details The GPU implementation cuHALLaR+ uses C/CUDA. It is compiled for the NVIDIA H200 GPU architecture with compute capability 9.0 using CMake 3.30.2, GCC 12.3.0, and NVIDIA `nvcc` 12.6.68 (CUDA Toolkit 12.6.1), and linked against OpenBLAS 0.3.23. The GPU experiments use one NVIDIA H200 with 142 GB of GPU memory on a PACE node equipped with an Intel Xeon Platinum 8469C CPU.

For maximum stable set, cuHALLaR+ uses hybrid sparse low-rank (HSLR) input [1]. All cuHALLaR baselines are the on-the-fly results reported in [2], using the settings reported there. These implementations, as well as the QAP implementations of cuHALLaR+ and HALLaR, evaluate CU, $\mathcal{A}(UU^\top)$, and $\mathcal{A}^*(p)U$ directly from the problem data without assembling the SDP coefficient matrices. cuLoRADS, CARDAL, and CSDP use SDPA input [64]. All solvers use this format for the Mittelmann collection.

Competing methods and comparison criteria The GPU experiments compare cuHALLaR+ with cuHALLaR, cuLoRADS v1.0.1 [23] and the single-GPU implementation of CARDAL v0.1.0 [37].

Given positive tolerances $(\varepsilon_p, \varepsilon_g, \varepsilon_d)$, cuHALLaR+ stops at $(\bar{X}, \bar{p})$, with $\bar{X} \succeq 0$ and $\bar{S} := C - \mathcal{A}^*\bar{p}$, when

$$\frac{\|\mathcal{A}\bar{X} - b\|_2}{1 + \|b\|_2} \leq \varepsilon_p, \qquad \frac{|\langle C, \bar{X}\rangle - \langle b, \bar{p}\rangle|}{1 + |\langle C, \bar{X}\rangle| + |\langle b, \bar{p}\rangle|} \leq \varepsilon_g, \qquad \frac{\max\{0, -\widehat{\lambda}_{\min}(\bar{S})\}}{1 + |\langle C, \bar{X}\rangle| + |\langle b, \bar{p}\rangle|} \leq \varepsilon_d, \tag{97}$$

where $\widehat{\lambda}_{\min}$ is the computed smallest eigenvalue. We set $\varepsilon_p = \varepsilon_g = \varepsilon_d = \epsilon$. HALLaR and cuHALLaR use the trace-constrained stopping tests of [44] and [2], respectively. The comparison target ϵ is 10^{-4} for QAP and 10^{-5} for every other problem class. Solver-specific stopping tests and phase settings are retained. The time limit is one hour for the Mittelmann SDPs, six hours for the QAPLIB instances, and eight hours for the random QAP instances.

Times are in seconds, and an entry time/rank also reports the final rank. The cuLoRADS v1.0.1 times include its dual-infeasibility evaluations, which cuLoRADS performs on the CPU after each phase and reports separately from its solver time, whereas the Mittelmann benchmark site[1] reports only the solver time. The dual residual is

[1] `https://plato.asu.edu/ftp/sparse_sdp.html`

part of the stopping test (97), and the cuHALLaR+ times include every evaluation of it. Therefore, we include the corresponding cuLoRADS dual evaluation time within its final solve time. We use TL to denote an instance that hit the time-limit and ** to denote a crash or an instance too large for the solver. An entry $*/N$ denotes an instance with reported maximum relative error N.

5.1 Maximum stable set

Given a graph $\mathcal{G} = ([n], E)$, the maximum stable set problem consists of finding a subset of vertices of largest cardinality such that no two vertices are connected by an edge. Lovász [39] introduced a constant, the ϑ-function, which upper bounds the value of the maximum stable set. The ϑ-function is the value of the SDP

$$\max \quad \{ee^\top \bullet X \ : \ X_{ij} = 0, \ ij \in E, \ \mathrm{tr}X = 1, \ X \succeq 0, \quad X \in \mathbb{S}^n(\mathbb{R})\}$$

where $e = (1, 1, \ldots, 1) \in \mathbb{R}^n$ is the all ones vector. It was shown in [21] that the ϑ-function agrees exactly with the stable set number for perfect graphs.

Table 1 compares cuHALLaR+ with cuHALLaR from [2], cuLoRADS, and CARDAL on the Hamming instances.

Both cuHALLaR+ and cuHALLaR solve all 14 Hamming instances and return rank-two solutions throughout. cuHALLaR+ is faster on 12 instances; cuHALLaR is faster on $H_{18,2}$ and $H_{19,2}$. cuLoRADS and CARDAL report solves on the first five instances, and cuHALLaR+ is faster than both on each of these. CARDAL is faster than cuLoRADS only on $H_{10,2}$. On the largest graph, with over eight million vertices, cuHALLaR+ takes 44.13 seconds, compared with 83.905 seconds for cuHALLaR.

Graph $(n; \lvert E\rvert)$	cuHALLaR+	cuHALLaR	cuLoRADS	CARDAL	cuLoRADS/ cuHALLaR+	CARDAL/ cuHALLaR+
$H_{10,2}$(1,024; 5,120)	0.552/2	1.735/2	1.05/14	0.83 / 19	**1.90**	**1.50**
$H_{11,2}$(2,048; 11,264)	0.48/2	1.627/2	1.23/16	1.96 / 21	**2.56**	**4.08**
$H_{12,2}$(4,096; 24,576)	1.57/2	2.033/2	1.91/17	7.01 / 23	**1.22**	**4.46**
$H_{13,2}$(8,192; 53,248)	0.95/2	3.106/2	5.56/19	54.72 / 24	**5.85**	**57.60**
$H_{14,2}$(16,384; 114,688)	0.93/2	3.170/2	15.44/20	539.45 / 26	**16.60**	**580.05**
$H_{15,2}$(32,768; 245,760)	2.52/2	3.223/2	**	**	–	–
$H_{16,2}$(65,536; 524,288)	3.69/2	4.457/2	**	**	–	–
$H_{17,2}$(131,072; 1,114,112)	2.25/2	3.904/2	**	**	–	–
$H_{18,2}$(262,144; 2,359,296)	3.05/2	2.627/2	**	**	–	–
$H_{19,2}$(524,288; 4,980,736)	7.81/2	5.281/2	**	**	–	–
$H_{20,2}$(1,048,576; 10,485,760)	5.19/2	8.109/2	**	**	–	–
$H_{21,2}$(2,097,152; 22,020,096)	8.73/2	16.966/2	**	**	–	–
$H_{22,2}$(4,194,304; 46,137,344)	14.91/2	37.145/2	**	**	–	–
$H_{23,2}$(8,388,608; 96,468,992)	44.13/2	83.905/2	**	**	–	–

Table 1: Runtimes for Hamming stable-set SDPs.

Table 2 reports the GSET results. The cuHALLaR runs from [2] use trace bound $\tau = 1$. Ratios are shown only for cuLoRADS runs that pass the accuracy check.

Both cuHALLaR+ and cuHALLaR solve all 28 GSET instances, compared with 26 for CARDAL and 19 for cuLoRADS. cuHALLaR+ is faster than cuHALLaR on 24 instances. On the largest graph, G81, cuHALLaR+ returns a rank-two solution in 48.91 seconds, compared with a rank-three solution in 64.20 seconds for cuHALLaR.

Graph $(n; \lvert E\rvert)$	cuHALLaR+	cuHALLaR	cuLoRADS	CARDAL	cuLoRADS/ cuHALLaR+
G1(800; 19,176)	46.97/101	45.19/98	49.29/108	1232.91 / 121	**1.05**
G10(800; 19,176)	44.81/101	40.52/97	$*/2.388 \times 10^{-5}$	1153.66 / 96	–
G11(800; 1,600)	4.36/2	4.23/2	113.20/32	1.36 / 15	**25.96**
G12(800; 1,600)	3.19/2	2.59/2	1.68/14	0.79 / 15	0.53
G14(800; 4,694)	124.21/58	214.73/74	25.25/32	4.47 / 17	0.20
G20(800; 4,672)	165.73/170	359.40/166	$*/2.646 \times 10^{-5}$	260.89 / 19	–
G43(1000; 9,990)	27.30/63	35.84/60	41.15/72	968.02 / 68	**1.51**
G51(1,000; 5,909)	174.10/134	273.84/128	67.89/72	14.39 / 19	0.39
G23(2,000; 19,990)	35.91/79	52.59/76	119.17/81	$*/6.536 \times 10^{-6}$	**3.32**
G31(2,000; 19,990)	33.94/68	58.23/76	138.56/81	$*/1.393 \times 10^{-5}$	**4.08**
G32(2,000; 4,000)	22.89/3	25.34/3	137.63/36	2.79 / 19	**6.01**
G34(2,000; 4,000)	22.43/3	25.29/3	7.50/16	2.52 / 17	0.33
G35(2,000; 11,778)	203.53/117	254.68/108	127.86/54	83.46 / 19	0.63
G41(2,000; 11,785)	231.95/148	286.82/139	83.58/36	23.15 / 20	0.36
G48(3,000; 6,000)	8.31/2	18.95/2	19.43/17	4.80 / 19	**2.34**
G55(5,000; 12,498)	64.68/48	81.11/45	445.35/62	127.89 / 24	**6.89**
G56(5,000; 12,498)	67.33/43	80.83/45	428.62/41	426.77 / 22	**6.37**
G57(5,000; 10,000)	3.94/2	42.55/2	$*/1.647 \times 10^{4}$	16.77 / 19	–
G58(5,000; 29,570)	381.04/127	504.55/130	$*/1.918 \times 10^{-4}$	3249.83 / 23	–
G59(5,000; 29,570)	379.55/125	539.28/121	$*/2.218 \times 10^{-4}$	3575.02 / 24	–
G60(7,000; 17,148)	68.41/48	96.89/50	647.88/41	637.34 / 25	**9.47**
G62(7,000; 14,000)	7.03/3	25.18/3	2017.50/168	59.86 / 20	**286.98**
G64(7,000; 41,459)	295.47/55	334.12/51	$*/2.316 \times 10^{-4}$	3235.03 / 22	–
G66(9,000; 18,000)	22.38/2	27.91/3	1787.71/29	74.35 / 23	**79.88**
G67(10,000; 20,000)	4.36/2	5.83/2	1815.13/29	207.55 / 20	**416.31**
G72(10,000; 20,000)	5.03/2	5.81/2	$*/7.960 \times 10^{-4}$	157.89 / 20	–
G77(14,000; 28,000)	9.54/2	11.81/2	$*/3.825 \times 10^{-4}$	636.46 / 21	–
G81(20,000; 40,000)	48.91/2	64.20/3	$*/3.724 \times 10^{-1}$	929.70 / 22	–

Table 2: Runtimes for GSET stable-set SDPs.

5.2 Matrix completion

Consider the problem of retrieving a low rank matrix $M \in \mathbb{R}^{n_1 \times n_2}$, where $n_1 \leq n_2$, by observing a subset of its entries: M_{ij}, $ij \in \mathcal{I}$. A standard approach to tackle this problem is by considering the nuclear norm relaxation:

$$\min_{Y} \quad \left\{ \|Y\|_* \quad : \quad Y_{ij} = M_{ij},\ \forall ij \in \mathcal{I}, \quad Y \in \mathbb{R}^{n_1 \times n_2} \right\}$$

The above problem can be rephrased as the following SDP:

$$\min_{X} \quad \left\{ \frac{1}{2}\mathrm{tr}(X) \quad : \quad X = \begin{pmatrix} W_1 & Y \\ Y^\top & W_2 \end{pmatrix} \succeq 0, \quad Y_{i,j} = M_{i,j}\ \forall ij \in \mathcal{I}, \quad X \in \mathbb{S}^{n_1+n_2}(\mathbb{R}) \right\}$$

The nuclear norm relaxation was introduced in [19]. It was shown in [11] it provably completes the matrix when $m = |\mathcal{I}|$ is sufficiently large and the indices of the observations are independent and uniform.

Matrix completion instances are generated randomly using the protocol in [2]: given $r \leq n_1 \leq n_2$, the hidden solution matrix is $M = M_1 M_2^\top$, where $M_1 \in \mathbb{R}^{n_1 \times r}$ and $M_2 \in \mathbb{R}^{n_2 \times r}$ have independent standard Gaussian entries. Then $m = \lceil r^2(n_1+n_2)\log(n_1+n_2) \rceil$ entries are sampled uniformly at random; the oversampling ratio is $r\log(n_1+n_2)$.

In Table 3, both cuHALLaR+ and cuHALLaR solve all 26 instances, while cuLoRADS and CARDAL solve 25 and 18, respectively. cuHALLaR+ is faster than cuHALLaR on all 26 instances and faster than cuLoRADS and CARDAL on every instance for which they report a solve. The final ranks of both cuHALLaR+ and cuHALLaR agree with the planted rank on 24 instances and exceed it by one on the remaining two. On the largest instance, with over 261 million observations, cuHALLaR+ takes 30.82 seconds, compared with 53.3 seconds for cuHALLaR and 4,124.44 seconds for cuLoRADS.

$(n_1, n_2; m)$	r	cuHALLaR+	cuHALLaR	cuLoRADS	CARDAL	cuLoRADS/ cuHALLaR+
3,000, 7,000; 828,931	3	0.53/3	0.77/3	3.12/19	0.79 / 28	**5.89**
3,000, 7,000; 828,931	3	0.64/3	0.70/3	3.14/19	$*/1.137 \times 10^{-4}$	**4.91**
3,000, 7,000; 2,302,586	5	0.83/5	1.28/5	3.79/19	1.15 / 35	**4.57**
3,000, 7,000; 2,302,586	5	0.79/5	1.09/5	3.42/19	1.08 / 35	**4.33**
10,000, 21,623; 2,948,996	3	1.04/3	1.22/3	9.36/21	2.92 / 30	**9.00**
10,000, 21,623; 2,948,996	3	1.02/3	1.11/3	9.21/21	2.90 / 30	**9.03**
10,000, 21,623; 8,191,654	5	1.68/5	2.05/5	12.64/21	487.26 / 40	**7.52**
10,000, 21,623; 8,191,654	5	1.73/5	1.95/5	10.92/21	8.40 / 37	**6.31**
25,000, 50,000; 3,367,574	2	1.17/2	1.20/2	16.72/23	5.90 / 31	**14.29**
25,000, 50,000; 7,577,040	3	1.41/3	1.82/3	21.14/23	11.72 / 35	**14.99**
30,000, 70,000; 4,605,171	2	1.59/2	1.63/2	25.79/24	8.59 / 31	**16.22**
30,000, 70,000; 10,361,633	3	2.01/3	2.57/3	27.13/24	19.06 / 34	**13.50**
50,000, 100,000; 7,151,035	2	1.42/2	1.89/2	40.20/24	933.59 / 33	**28.31**
50,000, 100,000; 16,089,828	3	2.57/3	3.68/3	40.13/24	2672.44 / 37	**15.61**
80,000, 120,000; 9,764,859	2	2.09/2	2.45/2	50.41/25	27.80 / 35	**24.12**
80,000, 120,000; 21,970,931	3	4.86/3	5.02/3	38.35/25	63.03 / 37	**7.89**
120,000, 180,000; 34,051,152	3	5.43/4	7.35/4	87.41/26	127.07 / 38	**16.10**
160,000, 240,000; 46,437,192	3	4.85/4	8.79/4	409.53/26	253.83 / 37	**84.44**
240,000, 360,000; 71,845,299	3	10.74/3	12.82/3	744.89/27	2797.54 / 37	**69.36**
320,000, 480,000; 97,865,043	3	14.43/3	17.41/3	1187.79/28	$*/9.668 \times 10^{-1}$	**82.31**
400,000, 600,000; 124,339,596	3	18.13/3	22.31/3	2037.88/28	$*/3.377 \times 10^{-5}$	**112.40**
480,000, 720,000; 151,176,587	3	24.41/3	28.13/3	$*/1.555 \times 10^{-6}$	$*/3.338 \times 10^{-1}$	–
560,000, 840,000; 178,314,984	3	30.69/3	34.72/3	460.49/29	$*/9.782 \times 10^{-1}$	**15.00**
640,000, 960,000; 205,711,405	3	30.36/3	39.25/3	3102.16/29	$*/9.802 \times 10^{-1}$	**102.18**
720,000, 1,080,000; 233,333,416	3	30.59/3	48.85/3	482.13/29	$*/9.951 \times 10^{-1}$	**15.76**
800,000, 1,200,000; 261,155,840	3	30.82/3	53.3/3	4124.44/30	$*/9.999 \times 10^{-1}$	**133.82**

Table 3: Runtimes for matrix-completion SDPs. The cuHALLaR results are reproduced from [2, Table 2], using tolerance 10^{-5} and a three-hour time limit.

5.3 Quadratic assignment problem

Given matrices $A, B \in \mathbb{S}^\nu$, the quadratic assignment problem (QAP) consists of minimizing $\mathrm{tr}(APBP^\top)$ over the $\nu \times \nu$ permutation matrices P. With column-wise vectorization, the identity

$$\mathrm{tr}(APBP^\top) = \mathrm{vec}(P)^\top (B \otimes A)\,\mathrm{vec}(P)$$

motivates the following SDP relaxation:

$$\min_{X} \quad \left\{ (B \otimes A) \bullet X \;:\; \begin{array}{l} \operatorname{tr}_1 X = I_\nu, \quad \operatorname{tr}_2 X = I_\nu, \\ e_{\nu^2} e_{\nu^2}^\top \bullet X = \nu^2, \quad X \succeq 0, \quad X \in \mathbb{S}^{\nu^2}(\mathbb{R}) \end{array} \right\}. \tag{98}$$

Here $e_{\nu^2} \in \mathbb{R}^{\nu^2}$ is the all-ones vector. The index (i,j) denotes the entry $i + (j-1)\nu$ of a vectorized $\nu \times \nu$ matrix, and the partial traces are defined by

$$[\operatorname{tr}_1 X]_{ij} = \sum_{\ell=1}^{\nu} X_{(i,\ell),(j,\ell)}, \qquad [\operatorname{tr}_2 X]_{ij} = \sum_{\ell=1}^{\nu} X_{(\ell,i),(\ell,j)}.$$

For every permutation matrix P, the rank-one matrix $X = \operatorname{vec}(P)\operatorname{vec}(P)^\top$ is feasible for (98) and has the same objective value. Thus, (98) is a relaxation of QAP and is equivalent to that in [44, Section 4.6], up to the ordering of the Kronecker factors. Its matrix dimension is $n = \nu^2$. For $\nu \geq 2$, we omit the redundant final diagonal equation of $\operatorname{tr}_2 X = I_\nu$, leaving $m = \nu(\nu+1)$ independent scalar constraints. The partial-trace constraints imply $\operatorname{tr} X = \nu$, but HALLaR+ does not require this trace bound as an input. HALLaR uses the trace bound $\tau = \nu$.

For nonsymmetric QAPLIB data, the objective matrix is replaced by $0.5(B \otimes A + B^\top \otimes A^\top)$. This symmetrization preserves the quadratic form and leaves the constraints in (98) unchanged.

Table 4 covers 32 QAPLIB instances, including the 11 `tai` instances in [44, Table 10]. All methods use a six-hour time limit.

cuHALLaR+ starts from $U_0 = e_{\nu^2}/\sqrt{\nu}$ and the zero multiplier. The objective is scaled by $1/(\|A\|_F \|B\|_F)$, and the same scaling is used in the stopping test (97). For all instances after the first 11, CSDP receives the independent equality constraints and the redundant constraint $\operatorname{tr} X + s = \nu$, $s \geq 0$.

CSDP reaches the target accuracy on 23 of the 32 instances, including the nine smallest `tai` instances. HALLaR reaches the target on seven instances. The final errors reported by cuLoRADS exceed the target on all 32 instances.

In Table 5, CSDP reaches the target on 15 of 18 instances, while HALLaR does so on four. cuLoRADS finishes above the target on all 18 instances. On the two largest instances, CSDP takes about 10,400 seconds on `esc64a` and has no completed result for `esc128`.

CSDP reaches the target on 18 of the 31 instances in Table 6, while cuLoRADS does so only on `sko64`. CSDP misses the accuracy target on all six `sko100` instances. On `tho150`, the largest instance in this group, cuHALLaR+ reaches the time limit.

Table 7 reports results at the 11 random QAP dimensions in [44, Table 12]. For our runs, A and B are symmetric with zero diagonal; each strict upper-triangular entry is obtained by rounding an independent uniform draw on $[0, 10]$. We generate one instance per size using Julia 1.10.1 and seed 0, and solve (98) with a tolerance of 10^{-4} and an eight-hour solver limit. HALLaR and cuHALLaR+ use the same Gaussian unit-norm initial factor.

HALLaR, CSDP, and cuLoRADS record no solves at the target accuracy on the random instances. At $\nu = 400$, the SDP has dimension 160,000 and the reported HALLaR error is 2.5×10^{-3}. For cuLoRADS, the $\nu = 200$ instance with $n = 40{,}000$ exhausts GPU memory on the H200 during solver initialization.

Instance	n	HALLaR	CSDP	cuHALLaR+	cuLoRADS
tai12a	144	$*/4.6 \times 10^{-4}$	0.333	2.753/17	$*/1.464 \times 10^{-2}$
tai15a	225	96.376/3	0.858	7.582/21	$*/1.254 \times 10^{4}$
tai20a	400	188.321/2	3.979	6.391/28	$*/9.061 \times 10^{-3}$
tai25a	625	$*/3.5 \times 10^{-4}$	13.998	9.228/35	$*/4.462 \times 10^{-2}$
tai30a	900	695.807/3	43.371	6.541/42	$*/3.809 \times 10^{-2}$
tai35a	1,225	$*/4.4 \times 10^{-4}$	121.273	9.114/49	$*/5.694 \times 10^{-2}$
tai40a	1,600	$*/8.9 \times 10^{-4}$	335.869	12.812/56	$*/4.236 \times 10^{-2}$
tai50a	2,500	$*/1.9 \times 10^{-4}$	1,894.414	18.169/70	$*/3.573 \times 10^{-2}$
tai60a	3,600	$*/4.7 \times 10^{-3}$	6,653.269	31.262/85	$*/4.454 \times 10^{-2}$
tai80a	6,400	$*/6.0 \times 10^{-3}$	$*/5.260 \times 10^{-1}$	117.579/113	$*/1.581 \times 10^{-2}$
tai100a	10,000	$*/6.9 \times 10^{-3}$	$*/5.721$	174.861/141	$*/4.399 \times 10^{-2}$
esc32a	1,024	$*/9.3 \times 10^{-4}$	150.310	6.423/45	$*/2.285 \times 10^{-2}$
esc32b	1,024	$*/3.8 \times 10^{-3}$	$*/2.021 \times 10^{-3}$	2.406/45	$*/2.651 \times 10^{-1}$
esc32c	1,024	$*/2.9 \times 10^{-3}$	171.951	3.735/45	$*/7.398 \times 10^{-2}$
esc32d	1,024	$*/8.6 \times 10^{-4}$	175.683	6.130/45	$*/3.270 \times 10^{-1}$
esc32e	1,024	$*/2.3 \times 10^{-3}$	$*/2.990 \times 10^{-4}$	3.022/45	$*/7.810 \times 10^{-3}$
esc32g	1,024	$*/2.3 \times 10^{-3}$	$*/1.856 \times 10^{-4}$	2.565/45	$*/9.339 \times 10^{-3}$
esc32h	1,024	$*/3.8 \times 10^{-4}$	185.051	3.408/45	$*/8.705 \times 10^{-2}$
kra30a	900	$*/1.1 \times 10^{-4}$	$*/3.935 \times 10^{-4}$	11.682/42	$*/9.852 \times 10^{-2}$
kra30b	900	$*/5.4 \times 10^{-4}$	46.562	8.673/42	$*/1.164 \times 10^{-1}$
kra32	1,024	$*/1.3 \times 10^{-3}$	$*/2.050 \times 10^{-4}$	14.626/45	$*/2.988 \times 10^{-1}$
lipa30a	900	$*/1.8 \times 10^{-3}$	38.018	6.417/42	$*/1.371 \times 10^{-2}$
lipa30b	900	2,748.815/24	39.653	7.326/42	$*/9.520 \times 10^{-3}$
nug21	441	$*/6.0 \times 10^{-4}$	5.230	6.462/29	$*/2.180 \times 10^{-2}$
nug22	484	$*/4.3 \times 10^{-4}$	7.235	9.317/31	$*/3.750 \times 10^{-2}$
nug24	576	$*/2.0 \times 10^{-4}$	11.002	14.059/34	$*/9.982 \times 10^{-1}$
nug25	625	1,732.228/35	$*/1.495 \times 10^{-4}$	4.876/35	$*/1.938 \times 10^{-2}$
nug27	729	$*/2.1 \times 10^{-3}$	23.242	18.794/38	$*/2.282 \times 10^{-2}$
nug28	784	2,696.431/39	$*/1.246 \times 10^{-4}$	20.518/39	$*/1.407$
nug30	900	5,652.199/42	41.891	30.640/42	$*/1.492 \times 10^{-2}$
tai25b	625	$*/2.8 \times 10^{-3}$	14.694	12.814/35	$*/3.355 \times 10^{-1}$
tho30	900	$*/9.9 \times 10^{-4}$	41.601	16.543/42	$*/5.090 \times 10^{-2}$

Table 4: QAPLIB SDP results.

Instance	n	HALLaR	CSDP	cuHALLaR+	cuLoRADS
chr12b	144	$*/7.7 \times 10^{-4}$	$*/2.539 \times 10^{-4}$	2.088/17	$*/1.832 \times 10^{-1}$
chr12c	144	$*/4.5 \times 10^{-4}$	0.319	2.854/17	$*/2.103$
chr15a	225	$*/2.3 \times 10^{-4}$	0.815	3.569/21	$*/8.525 \times 10^{-2}$
chr15b	225	$*/5.0 \times 10^{-4}$	0.789	4.608/21	$*/7.327 \times 10^{-2}$
chr15c	225	462.838/18	0.792	8.163/21	$*/4.291 \times 10^{-2}$
chr18a	324	$*/8.0 \times 10^{-4}$	2.084	4.675/25	$*/5.505 \times 10^{-2}$
chr18b	324	$*/1.1 \times 10^{-3}$	$*/2.776 \times 10^{-4}$	4.298/25	$*/1.070 \times 10^{-1}$
chr20a	400	$*/1.0 \times 10^{-3}$	3.786	6.732/28	$*/4.751 \times 10^{-2}$
chr20b	400	$*/1.5 \times 10^{-4}$	3.379	3.979/28	$*/3.675 \times 10^{-2}$
chr20c	400	$*/1.9 \times 10^{-4}$	3.851	5.675/28	$*/1.228 \times 10^{-1}$
els19	361	$*/8.0 \times 10^{-4}$	3.076	5.099/27	$*/8.363 \times 10^{-2}$
esc64a	4,096	$*/1.2 \times 10^{-3}$	10,400.143	328.053/90	$*/6.617 \times 10^{-2}$
esc128	16,384	$*/7.7 \times 10^{-3}$	**	96.435/181	$*/9.880 \times 10^{-1}$
had12	144	81.567/6	0.348	4.673/17	$*/4.059 \times 10^{-2}$
had14	196	94.103/2	0.614	6.275/20	$*/6.711 \times 10^{-2}$
had16	256	193.668/8	1.324	5.277/22	$*/6.728 \times 10^{-2}$
had18	324	$*/1.0 \times 10^{-4}$	2.297	5.746/25	$*/6.602 \times 10^{-2}$
had20	400	$*/2.0 \times 10^{-3}$	3.973	7.221/28	$*/2.495 \times 10^{-2}$

Table 5: Remaining QAPLIB instances of [44, Table 11], part 1 of 2.

Instance	n	HALLaR	CSDP	cuHALLaR+	cuLoRADS
nug12	144	$*/2.8 \times 10^{-4}$	0.312	2.720/17	$*/2.045 \times 10^{-2}$
nug14	196	$*/3.9 \times 10^{-4}$	0.618	2.683/20	$*/3.877 \times 10^{-2}$
nug15	225	151.787/7	0.817	2.162/21	$*/4.266 \times 10^{-2}$
nug16a	256	$*/3.9 \times 10^{-4}$	1.269	4.111/22	$*/8.999 \times 10^{-2}$
nug16b	256	$*/2.9 \times 10^{-4}$	1.199	4.089/22	$*/7.426 \times 10^{-2}$
nug17	289	254.685/7	1.538	3.864/24	$*/8.621 \times 10^{-2}$
nug18	324	417.714/14	2.097	3.865/25	$*/8.741 \times 10^{-2}$
nug20	400	$*/6.5 \times 10^{-4}$	3.641	6.658/28	$*/1.245 \times 10^{-1}$
rou12	144	70.956/3	0.329	1.825/17	$*/5.214 \times 10^{1}$
rou15	225	$*/1.8 \times 10^{-4}$	0.800	6.946/21	$*/6.922 \times 10^{-2}$
rou20	400	194.810/4	3.818	6.890/28	$*/4.727 \times 10^{-2}$
scr12	144	$*/1.1 \times 10^{-3}$	$*/8.053 \times 10^{-4}$	6.117/17	$*/5.735 \times 10^{-1}$
scr15	225	$*/6.5 \times 10^{-4}$	$*/1.031 \times 10^{-3}$	3.267/21	$*/4.563 \times 10^{-1}$
scr20	400	$*/2.3 \times 10^{-3}$	$*/8.460 \times 10^{-4}$	4.588/28	$*/4.714 \times 10^{-1}$
sko42	1,764	$*/7.3 \times 10^{-3}$	425.000	20.370/59	$*/1.270 \times 10^{-2}$
sko49	2,401	$*/3.0 \times 10^{-3}$	1,446.965	27.557/69	$*/3.462 \times 10^{-2}$
sko56	3,136	$*/2.5 \times 10^{-3}$	3,845.140	30.136/79	$*/1.000$
sko64	4,096	$*/5.1 \times 10^{-3}$	12,333.661	70.466/90	134.090
sko100a	10,000	$*/6.8 \times 10^{-3}$	$*/2.184 \times 10^{4}$	202.732/141	$*/2.454 \times 10^{-2}$
sko100b	10,000	$*/5.4 \times 10^{-3}$	$*/3.144 \times 10^{3}$	205.820/143	$*/2.401 \times 10^{-2}$
sko100c	10,000	$*/9.7 \times 10^{-3}$	$*/1.955 \times 10^{4}$	209.715/142	$*/5.477 \times 10^{-2}$
sko100d	10,000	$*/9.7 \times 10^{-3}$	$*/2.773 \times 10^{3}$	216.082/142	$*/8.624 \times 10^{-3}$
sko100e	10,000	$*/9.7 \times 10^{-3}$	$*/2.625 \times 10^{3}$	215.098/141	$*/1.416 \times 10^{-2}$
sko100f	10,000	$*/9.7 \times 10^{-3}$	$*/2.873 \times 10^{3}$	217.526/143	$*/3.214 \times 10^{-2}$
ste36a	1,296	$*/1.4 \times 10^{-3}$	139.864	170.791/51	$*/9.751 \times 10^{-2}$
ste36b	1,296	$*/2.5 \times 10^{-4}$	$*/1.985 \times 10^{-4}$	100.028/51	$*/6.575 \times 10^{-2}$
ste36c	1,296	5,445.863/33	141.673	100.777/51	$*/1.368 \times 10^{-1}$
tho40	1,600	$*/2.9 \times 10^{-3}$	$*/1.669 \times 10^{-4}$	13.467/56	$*/4.046$
tho150	22,500	$*/2.1 \times 10^{-3}$	**	TL	$*/3.218 \times 10^{-1}$
wil50	2,500	$*/4.3 \times 10^{-3}$	1,874.664	202.683/70	$*/3.859 \times 10^{-2}$
wil100	10,000	$*/7.4 \times 10^{-3}$	$*/2.975 \times 10^{4}$	220.789/141	$*/2.453 \times 10^{-2}$

Table 6: Remaining QAPLIB instances of [44, Table 11], part 2 of 2.

ν	n	HALLaR	CSDP	cuHALLaR+	cuLoRADS
100	10,000	$*/3.6 \times 10^{-3}$	$*/1.159 \times 10^{2}$	210.226/141	$*/2.103 \times 10^{-2}$
120	14,400	$*/3.1 \times 10^{-3}$	$*/1.452 \times 10^{5}$	199.461/169	$*/2.972 \times 10^{-2}$
150	22,500	$*/2.9 \times 10^{-3}$	**	425.046/192	$*/1.000$
175	30,625	$*/4.8 \times 10^{-3}$	**	605.946/227	$*/1.000$
200	40,000	$*/3.5 \times 10^{-3}$	**	826.315/263	**
250	62,500	$*/3.6 \times 10^{-3}$	**	1,381.325/253	**
275	75,625	$*/3.6 \times 10^{-3}$	**	1,724.889/289	**
300	90,000	$*/2.6 \times 10^{-3}$	**	2,107.618/324	**
350	122,500	$*/2.5 \times 10^{-3}$	**	3,011.551/395	**
375	140,625	$*/2.0 \times 10^{-3}$	**	3,529.608/330	**
400	160,000	$*/2.5 \times 10^{-3}$	**	4,096.032/465	**

Table 7: Random QAP SDP results.

5.4 General SDP benchmark

Table 8 reports results for the same 27 Mittelmann SDP instances [43] considered in [23, Table 7]. CARDAL solves 14 of the 27 instances and reaches the one-hour limit on the remaining 13. cuLoRADS has reported runtimes on 15 instances.

Instance	n	m	cuHALLaR+	cuLoRADS	CARDAL
1dc.1024	1024	24064	43.92	$*/2.485 \times 10^{-6}$	1513.56
1et.2048	2048	22529	65.62	$*/4.858 \times 10^{-6}$	908.64
1tc.2048	2048	18945	44.28	180.96	763.98
1zc.1024	1024	16641	19.95	$*/1.970 \times 10^{-6}$	276.82
AlH	5990	7230	6.60	$*/1.621 \times 10^{-5}$	$*/6.011 \times 10^{-3}$
BH2	2166	1743	7.12	$*/9.283 \times 10^{-5}$	$*/1.017 \times 10^{-3}$
cancer_100	569	10469	42.79	$*/1.578 \times 10^{-6}$	624.05
CH2	2166	1743	67.89	$*/2.434 \times 10^{-5}$	$*/1.306 \times 10^{-3}$
checker_1.5	3970	3970	7.60	28.90	210.92
cphil12	364	12376	40.69	$*/4.428 \times 10^{-6}$	$*/2.969 \times 10^{-3}$
fap09	28224	30276	242.79	548.29	$*/2.506 \times 10^{-2}$
foot	2208	2209	95.39	74.28	$*/1.150 \times 10^{-6}$
G40_mb	2000	2001	15.12	59.20	1456.10
G48_mb	3000	3001	4.82	15.35	542.90
G60_mb	7000	7001	35.51	141.42	$*/2.775 \times 10^{-5}$
H3O	3162	2964	36.95	$*/1.642 \times 10^{-5}$	$*/6.406 \times 10^{-4}$
hand	1296	1297	3.09	4.22	94.26
ice_2.0	8113	8113	51.47	72.39	236.10
NH2	2166	1743	127.89	$*/1.364 \times 10^{-5}$	$*/1.377 \times 10^{-3}$
NH3	3162	2964	126.95	$*/2.189 \times 10^{-5}$	$*/7.693 \times 10^{-4}$
NH4	4426	4743	46.20	$*/8.608 \times 10^{-6}$	$*/2.636 \times 10^{-2}$
p_auss2_3.0	9115	9115	28.75	68.08	215.86
rendl1	2000	2001	10.39	11.18	$*/3.668 \times 10^{-6}$
sensor_1000	6528	5549	54.97	131.24	$*/4.356 \times 10^{-2}$
theta12	600	17979	29.86	32.28	919.99
theta102	500	37467	15.28	17.54	1560.72
theta123	600	90020	46.37	52.91	2428.30

Table 8: Runtimes for Mittelmann SDPs.

6 Concluding remarks

This paper develops HALLaR+, a low-rank inexact augmented Lagrangian method for SDPs with possibly unbounded feasible regions. A key practical advantage is that the method does not require a user-specified trace bound τ. Choosing such a bound can be difficult: a value that is too small may exclude all optimal solutions, whereas an overly conservative value may substantially increase the computational effort. Under the stated assumptions, we establish a uniform bound on the diameters of the relevant augmented Lagrangian sublevel sets and use it to derive iteration-complexity guarantees for obtaining approximate primal–dual solutions.

The numerical experiments demonstrate strong computational performance on several classes of SDPs, with substantial speedups over existing solvers on many instances. On the Mittelmann benchmarks, cuHALLaR+ achieves speedups of up to approximately 4.09× over cuLoRADS and 112.63× over CARDAL among instances successfully solved by the respective methods. It also solves the largest random QAP relaxation tested, with matrix dimension $160,000$, in approximately 68 minutes.

Future work includes extending HALLaR+ to SDPs with inequality constraints and multiple positive semidefinite blocks, as well as to doubly nonnegative programs. For the latter, an important challenge is to handle the entrywise nonnegativity constraints while preserving the memory advantages of the low-rank factorization.

A X-Space and Convex-Analysis Machinery

This appendix collects the technical convex-analysis facts and describes the relaxed unbounded Frank–Wolfe method used in our X-space analysis.

A.1 Technical Convex-Analysis Facts

This subsection collects the convex-analysis facts used in the X-AL analysis of Section 3.

The next proposition establishes the boundedness property used for the AL sublevel sets in the X-AL analysis.

Proposition A.1. *Assume that Assumption 2.1 holds. For every $q \in \mathbb{R}^m$, $\beta > 0$, and $\kappa \in \mathbb{R}$, the set*

$$\left\{X \in \mathbb{S}^n_+ : \mathcal{L}_\beta(X; q) \le \kappa\right\} \tag{99}$$

is bounded.

Proof. The proof can be found in [49, Theorem 8.4, p. 64] (see also [26, Proposition A.2.2.3, p. 40]). □

For every $\tau > 0$, let

$$\Delta_\tau \;:=\; \left\{X \in \mathbb{S}^n_+ :\; \mathrm{tr}(X) \le \tau\right\}. \tag{100}$$

For every $G \in \mathbb{S}^n$, the spectral decomposition of G gives

$$\min_{X \in \Delta_\tau} \langle G, X\rangle \;=\; \tau \min\left\{\lambda_{\min}(G), 0\right\}. \tag{101}$$

Proposition A.2. *Let $\tau > 0$, $X \in \mathbb{S}^n_+$, and $G \in \mathbb{S}^n$, and define*

$$\epsilon_\tau(X; G) \;:=\; \max_{Y \in \Delta_\tau} \langle G, X - Y\rangle. \tag{102}$$

Then the following statements hold:

(a) We have

$$\epsilon_\tau(X; G) \;\le\; |\langle G, X\rangle| + \tau \max\{\lambda_{\max}(-G), 0\}. \tag{103}$$

(b) If $X \in \Delta_{\tau/2}$, then

$$3\epsilon_\tau(X; G) \;\ge\; |\langle G, X\rangle| + \tau \max\{\lambda_{\max}(-G), 0\}. \tag{104}$$

Proof. a) Relations (101) and (102) imply that

$$\epsilon_\tau(X; G) = \langle G, X\rangle - \tau \min\{\lambda_{\min}(G), 0\} = \langle G, X\rangle + \tau \max\{\lambda_{\max}(-G), 0\}. \tag{105}$$

and hence that (103) holds.

b) Let $X \in \Delta_{\tau/2}$ be given. Since 0 and $2X$ are both in Δ_τ due to the definition of Δ_τ in (100), it follows from (102) with $Y = 0$ and $Y = 2X$ that

$$\epsilon_\tau(X; G) \ge \max\left\{\langle G, X - 0\rangle, \langle G, X - 2X\rangle\right\} = |\langle G, X\rangle|. \tag{106}$$

Combining (105) and (106), we obtain

$$\tau \max\{\lambda_{\max}(-G), 0\} = \epsilon_\tau(X; G) - \langle G, X\rangle \le \epsilon_\tau(X; G) + |\langle G, X\rangle| \le 2\epsilon_\tau(X; G).$$

Adding this inequality to (106) proves (104). □

A.2 X-Space Relaxed Unbounded Frank–Wolfe Material

Given an initial point $\tilde{X}_0 \in \mathbb{S}^n_+$, this subsection presents the Relaxed Unbounded Frank-Wolfe (RUFW) framework for solving

$$\phi_* \;:=\; \min\{\phi(X) :\; X \in \mathbb{S}^n_+\}, \tag{107}$$

where the function $\phi : \mathbb{S}^n_+ \to \mathbb{R}$ is assumed to satisfy the following conditions:

(A1) ϕ is convex and differentiable on $\mathbb{S}^n_+$, and its gradient is L-Lipschitz continuous for some $L > 0$, i.e.,

$$\|\nabla\phi(X') - \nabla\phi(X)\| \le L\|X' - X\| \qquad \forall X, X' \in \mathbb{S}^n_+;$$

(A2) the sublevel set $\mathcal{S}(\tilde{X}_0;\phi)$ is bounded, where $\mathcal{S}(\cdot;\cdot)$ is defined as in (5) with $\Omega = \mathbb{S}^n_+$.

Assumption **(A1)** implies that

$$\phi(X') \;\le\; \phi(X) + \langle \nabla\phi(X), X' - X\rangle + \frac{L}{2}\|X' - X\|^2 \qquad \forall\, X, X' \in \mathbb{S}^n_+. \tag{108}$$

Here, the inner product is the Froebnius one and the norm is the one induced by the Frobenius inner product, i.e., $\langle U, V\rangle = \mathrm{tr}(U^T V)$ and $\|U\|^2 = \mathrm{tr}(U^T U)$.

Under **(A1)**-**(A2)**, it is also easy to see the minimum in (107) is attained since **(A1)**–**(A2)** imply that ϕ is a continuous coercive function. Likewise, these assumptions imply that step 4 in the RUFW framework below is well-defined. We now describe the type of solution that RUFW aims to find.

Problem: Given a function ϕ and initial point $\tilde{X}_0 \in \mathbb{S}^n_+$ satisfying **(A1)**–**(A2)** and a tolerance pair $(\epsilon, \delta) \in \mathbb{R}^2_{++}$, RUFW aims to find a point $\bar{X} \in \mathbb{S}^n_+$ satisfying

$$\lambda_{\min}\left(\nabla\phi(\bar{X})\right) \ge -\epsilon, \qquad \left|\left\langle \nabla\phi(\bar{X}), \bar{X}\right\rangle\right| \le \delta; \tag{109}$$

$$\phi(\bar{X}) \le \phi(\tilde{X}_0). \tag{110}$$

The formal description of the relaxed unbounded Frank–Wolfe (RUFW) framework is given below.

Relaxed Unbounded Frank–Wolfe (RUFW) Framework

Input: tolerance pair $(\epsilon, \delta) \in \mathbb{R}^2_{++}$, function $\phi : \mathbb{S}^n_+ \to \mathbb{R}$, and initial point $\tilde{X}_0 \in \mathbb{S}^n_+$.

Step 0. Set $k \leftarrow 1$.

Step 1. Choose $X_k \in \mathbb{S}^n_+$ such that

$$\phi(X_k) \;\le\; \phi(\tilde{X}_{k-1}). \tag{111}$$

Step 2. Set $G_k \leftarrow \nabla\phi(X_k)$. Compute a minimum eigenpair $(\lambda_k^{\min}(G_k), v_k^{\min}(G_k))$ of G_k with $\|v_k^{\min}(G_k)\| = 1$, and set

$$y_k^F := \begin{cases} v_k^{\min}(G_k), & \text{if } \lambda_k^{\min}(G_k) < 0, \\ 0, & \text{if } \lambda_k^{\min}(G_k) \ge 0, \end{cases} \qquad X_k^F \;:=\; y_k^F (y_k^F)^\top; \tag{112}$$

Step 3. If

$$\lambda_k^{\min}(G_k) \;\ge\; -\epsilon, \qquad |\langle G_k, X_k\rangle| \;\le\; \delta, \tag{113}$$

then set $\bar{X} \leftarrow X_k$, output $\bar{X}$, and **stop**.

Step 4. Compute

$$(\alpha_k, \omega_k) \;\in\; \arg\min\left\{\phi(\alpha X_k + \omega X_k^F) :\; \alpha \ge 0,\; \omega \ge 0\right\} \tag{114}$$

and set

$$\tilde{X}_k \;:=\; \alpha_k X_k + \omega_k X_k^F; \tag{115}$$

Step 5. Set $k \leftarrow k + 1$ and return to Step **1**.

Step 1 allows any choice of $X_k \in \mathbb{S}^n_+$ whose objective value does not exceed that of the current iterate $\tilde{X}_{k-1}$. Step 2 evaluates the gradient $G_k = \nabla\phi(X_k)$ and computes a minimum eigenpair of G_k to construct X_k^F and check the stopping conditions (113) in Step 3. If these conditions are not satisfied, Step 4 determines the coefficients (α_k, ω_k) by solving the two-dimensional optimization problem (114), following an approach similar to that of Laue [35]. The next iterate $\tilde{X}_k$ is then formed as the conic combination of X_k and X_k^F given in (115). Since $(\alpha, \omega) = (1, 0)$ is feasible in (114), Step 4 does not increase the objective value. Together with (111), this ensures that $\phi(X_k) \le \phi(\tilde{X}_0)$ for every k. Thus, whenever RUFW terminates in Step 3, its output $\bar{X}$ satisfies both (109) and (110).

The following proposition states the main convergence result of the RUFW framework. Its proof is given in the following subsubsection. In contrast to classical Frank–Wolfe complexity bounds, the bound in (117) below does not depend on the initial objective gap $\phi(\tilde{X}_0) - \phi_*$. The initial point $\tilde{X}_0$ enters the bound only through the radius $\bar{\tau}$ of its sublevel set.

Proposition A.3. *Assume that **(A1)**–**(A2)** hold and define*

$$\bar{\tau} \;:=\; \max\left\{1,\; 2\sup\left\{\mathrm{tr}(X) : X \in \mathbb{S}^n_+,\; \phi(X) \le \phi(\tilde{X}_0)\right\}\right\}, \qquad \epsilon_k(\bar{\tau}) \;:=\; \max_{X \in \Delta_{\bar{\tau}}} \langle G_k, X_k - X\rangle. \tag{116}$$

Then, the following statements hold for RUFW:

(a) for every iteration index $k \ge 1$, it holds that $\phi(X_k) \le \phi(\tilde{X}_0)$;

(b) RUFW terminates with a point $\bar{X}$ satisfying (109) *and* (110) *in at most*

$$\left\lfloor 2 + 12\bar{\tau}^2 L\left(\frac{1}{\delta} + \frac{2}{\epsilon}\right)\right\rfloor \tag{117}$$

iterations.

A.2.1 Proof of Proposition A.3

This subsubsection proves Proposition A.3. Recall the trace-bounded spectrahedron Δ_τ and its linear-oracle value from (100) and (101), respectively. Let X_* be a minimizer of (107). Since $\phi(X_*) \le \phi(\tilde{X}_0)$, it follows from **(A2)** and (116) that $X_* \in \Delta_{\bar{\tau}/2} \subseteq \Delta_{\bar{\tau}}$. We now begin with a lemma describing key properties of the RUFW iterates.

Lemma A.4. *For every iteration index $k \ge 1$, the following statements hold:*

(a) it holds

$$\phi(\tilde{X}_k) \le \phi(X_k) \le \phi(\tilde{X}_{k-1}), \qquad \mathrm{tr}(X_k) \le \bar{\tau}/2, \qquad \mathrm{tr}(\tilde{X}_k) \le \bar{\tau}/2; \tag{118}$$

(b) the FW gap $\epsilon_k(\bar{\tau})$ satisfies

$$|\langle G_k, X_k\rangle| \;\le\; \epsilon_k(\bar{\tau}), \qquad \lambda_k^{\min}(G_k) \;\ge\; -\frac{2\epsilon_k(\bar{\tau})}{\bar{\tau}}. \tag{119}$$

Proof. (a) Relation (111) gives $\phi(X_k) \le \phi(\tilde{X}_{k-1})$. Moreover, the choice of (α_k, ω_k) in (114) and the update (115) give $\phi(\tilde{X}_k) \le \phi(X_k)$. Applying these inequalities recursively yields

$$\phi(X_k) \le \phi(\tilde{X}_0), \qquad \phi(\tilde{X}_k) \le \phi(\tilde{X}_0).$$

The trace bounds now follow from the definition of $\bar{\tau}$ in (116).

(b) By part (a) and Step 1, $X_k \in \Delta_{\bar{\tau}/2}$. Moreover, the definition of the FW gap and (102) imply that

$$\epsilon_k(\bar{\tau}) = \max_{X \in \Delta_{\bar{\tau}}} \langle G_k, X_k - X\rangle = \epsilon_{\bar{\tau}}(X_k; G_k).$$

Hence, the argument proving Proposition A.2*(b)*, applied with $(\tau, X, G) = (\bar{\tau}, X_k, G_k)$, and the identity $\lambda_k^{\min}(G_k) = -\lambda_{\max}(-G_k)$ yield (119). □

The following lemma characterizes key properties of $\epsilon_k(\bar{\tau})$ and X_k^F.

Lemma A.5. *For every iteration index $k \ge 1$, the following relations hold:*

$$\bar{\tau} X_k^F \;\in\; \arg\min_{X \in \Delta_{\bar{\tau}}} \langle G_k, X\rangle, \qquad \epsilon_k(\bar{\tau}) \;=\; \langle G_k, X_k - \bar{\tau} X_k^F\rangle, \tag{120}$$

$$\epsilon_k(\bar{\tau}) \;\ge\; \phi(X_k) - \phi_*. \tag{121}$$

Proof. It is easy to see from (112), the unit eigenvector convention, and (101) with $\tau = 1$ that $X_k^F \in \arg\min_{X \in \Delta_1} \langle G_k, X\rangle$: if $\lambda_k^{\min} < 0$, then $X_k^F = v_k^{\min}(G_k)(v_k^{\min}(G_k))^\top$ attains $\lambda_k^{\min}(G_k)$, while if $\lambda_k^{\min}(G_k) \ge 0$, then $X_k^F = 0$ attains the zero minimum. Scaling gives the first relation in (120). The second relation in (120) follows immediately from the first relation and the definition of $\epsilon_k(\bar{\tau})$ in (116). Finally, convexity, $X_* \in \Delta_{\bar{\tau}}$, and (116) give $\phi_* = \phi(X_*) \ge \phi(X_k) + \langle G_k, X_* - X_k\rangle = \phi(X_k) - \langle G_k, X_k - X_*\rangle \ge \phi(X_k) - \epsilon_k(\bar{\tau})$, which is (121). □

The following proposition gives the one-step estimate on which the analysis is based. It uses the fact that, for every $\alpha \in [0, 1]$, the conic search in Step 4 includes the point $(1-\alpha)X_k + \alpha\bar{\tau}X_k^F$.

Proposition A.6. *For every iteration index $k \geq 1$ and every $\alpha \in [0,1]$, we have*

$$\phi(\tilde{X}_k) \;\leq\; \phi(X_k) - \alpha\,\epsilon_k(\bar{\tau}) + \frac{3\bar{\tau}^2 L}{2}\,\alpha^2, \tag{122}$$

where $\epsilon_k(\bar{\tau})$ is as in (116).

Proof. Let $\alpha \in [0,1]$ be given. Since the pair $(1-\alpha, \alpha\bar{\tau})$ is feasible in (114), relations (114)–(115) imply $\phi(\tilde{X}_k) \leq \phi((1-\alpha)X_k + \alpha\bar{\tau}X_k^F) = \phi(X_k + \alpha(\bar{\tau}X_k^F - X_k))$. Observe also that the first relation in (120) and (118) imply that

$$\|\bar{\tau}X_k^F - X_k\| \leq \|\bar{\tau}X_k^F\| + \|X_k\| \leq \frac{3\bar{\tau}}{2}. \tag{123}$$

It then follows from the above observations, (123), $G_k = \nabla\phi(X_k)$, and (108) with $X' = X_k + \alpha(\bar{\tau}X_k^F - X_k) \in \mathbb{S}_+^n$ that

$$\begin{aligned}\phi(\tilde{X}_k) \leq \phi\left(X_k + \alpha(\bar{\tau}X_k^F - X_k)\right) &\leq \phi(X_k) + \alpha\langle \nabla\phi(X_k), \bar{\tau}X_k^F - X_k\rangle + \frac{L\alpha^2}{2}\|\bar{\tau}X_k^F - X_k\|^2 \\ &\overset{(123)}{\leq} \phi(X_k) + \alpha\langle G_k, \bar{\tau}X_k^F - X_k\rangle + \frac{9L\alpha^2\bar{\tau}^2}{8} \overset{(120)}{=} \phi(X_k) - \alpha\epsilon_k(\bar{\tau}) + \frac{9L\alpha^2\bar{\tau}^2}{8}.\end{aligned}$$

Relation (122) then follows immediately from the above inequalities and the fact that $9/8 \leq 3/2$. □

The next lemma establishes $\mathcal{O}(1/k)$ bounds on the objective gap and the minimum Frank–Wolfe gap in (121), both independent of the initial objective gap $\phi(\tilde{X}_0) - \phi_*$. By the two relations in (119), the latter bound also controls the quantities appearing in the stopping conditions (113).

Lemma A.7. *Let $k \geq 1$ be an iteration index such that Step 4 is performed at iterations $1, \ldots, k$. Then the following statements hold:*

(a) we have

$$\phi(\tilde{X}_k) - \phi_* \;\leq\; \frac{6\bar{\tau}^2 L}{k+3}; \tag{124}$$

(b) if $k \geq 2$, then

$$\min_{1\leq j\leq k} \epsilon_j(\bar{\tau}) \;\leq\; \frac{12\bar{\tau}^2 L}{k}. \tag{125}$$

Proof. (a) Let $j \in \{1, \ldots, k\}$ and $\alpha \in [0,1]$ be given. Using relations (122) and (121), the fact that $1-\alpha \geq 0$, and the second inequality in (118), we obtain

$$\begin{aligned}\phi(\tilde{X}_j) - \phi_* &\overset{(122)}{\leq} \phi(X_j) - \phi_* - \alpha\,\epsilon_j(\bar{\tau}) + \frac{3\bar{\tau}^2 L}{2}\alpha^2 \overset{(121)}{\leq} (1-\alpha)\left[\phi(X_j) - \phi_*\right] + \frac{3\bar{\tau}^2 L}{2}\alpha^2 \\ &\overset{(118)}{\leq} (1-\alpha)\left[\phi(\tilde{X}_{j-1}) - \phi_*\right] + \frac{3\bar{\tau}^2 L}{2}\alpha^2.\end{aligned} \tag{126}$$

We now prove (124) by induction. Relation (126) with $j = 1$ and $\alpha = 1$ gives

$$\phi(\tilde{X}_1) - \phi_* \;\leq\; \frac{3\bar{\tau}^2 L}{2} \;=\; \frac{6\bar{\tau}^2 L}{1+3},$$

which is (124) with $k = 1$. Observe that this bound does not depend on $\tilde{X}_0$ since the coefficient $1-\alpha$ multiplying $\phi(\tilde{X}_0) - \phi_*$ in (126) vanishes. Assume now that $2 \leq j \leq k$ and that $\phi(\tilde{X}_{j-1}) - \phi_* \leq 6\bar{\tau}^2 L/(j+2)$. Relation (126) with $\alpha = 2/(j+2) \in [0,1]$ and this assumption then imply that

$$\phi(\tilde{X}_j) - \phi_* \;\leq\; \frac{j}{j+2}\cdot\frac{6\bar{\tau}^2 L}{j+2} + \frac{3\bar{\tau}^2 L}{2}\cdot\frac{4}{(j+2)^2} \;=\; \frac{6\bar{\tau}^2 L(j+1)}{(j+2)^2} \;\leq\; \frac{6\bar{\tau}^2 L}{j+3},$$

where the last inequality is due to the fact that $(j+1)(j+3) \leq (j+2)^2$. Relation (124) then follows by induction.

(b) Let $k \geq 2$ be given and note that $\lfloor k/2 \rfloor \geq 1$. Let $\alpha \in (0,1]$ and $j \in \{1, \ldots, k\}$ be given. Relation (122) and the second inequality in (118) imply that

$$\alpha\,\epsilon_j(\bar{\tau}) \overset{(122)}{\leq} \phi(X_j) - \phi(\tilde{X}_j) + \frac{3\bar{\tau}^2 L}{2}\alpha^2 \overset{(118)}{\leq} \phi(\tilde{X}_{j-1}) - \phi(\tilde{X}_j) + \frac{3\bar{\tau}^2 L}{2}\alpha^2. \tag{127}$$

Summing (127) over $j = \lfloor k/2 \rfloor + 1, \ldots, k$, a range with $\lceil k/2 \rceil$ indices, and using the fact that $\phi(\tilde{X}_k) \geq \phi_*$ and (124) with k replaced by $\lfloor k/2 \rfloor$, we obtain

$$\alpha \left\lceil \frac{k}{2} \right\rceil \min_{1 \leq j \leq k} \epsilon_j(\bar{\tau}) \leq \alpha \sum_{j=\lfloor k/2 \rfloor + 1}^{k} \epsilon_j(\bar{\tau}) \overset{(127)}{\leq} \phi(\tilde{X}_{\lfloor k/2 \rfloor}) - \phi(\tilde{X}_k) + \left\lceil \frac{k}{2} \right\rceil \frac{3\bar{\tau}^2 L}{2} \alpha^2$$
$$\overset{(124)}{\leq} \frac{6\bar{\tau}^2 L}{\lfloor k/2 \rfloor + 3} + \left\lceil \frac{k}{2} \right\rceil \frac{3\bar{\tau}^2 L}{2} \alpha^2.$$

Taking $\alpha = 2/(\lfloor k/2 \rfloor + 3) \in (0, 1]$ in the above inequality and dividing the result by $\alpha \lceil k/2 \rceil$, we conclude that

$$\min_{1 \leq j \leq k} \epsilon_j(\bar{\tau}) \ \leq \ \frac{3\bar{\tau}^2 L}{\lceil k/2 \rceil} + \frac{3\bar{\tau}^2 L}{\lfloor k/2 \rfloor + 3} \ \leq \ \frac{6\bar{\tau}^2 L}{k} + \frac{6\bar{\tau}^2 L}{k} \ = \ \frac{12\bar{\tau}^2 L}{k},$$

where the last inequality is due to the fact that both $\lceil k/2 \rceil$ and $\lfloor k/2 \rfloor + 3$ are bounded below by $k/2$. □

We are now ready to prove Proposition A.3.

Proof of Proposition A.3. (a) Applying the first relation in (118) recursively implies that $\phi(X_k) \leq \phi(\tilde{X}_0)$ for every iteration index $k \geq 1$.

(b) Let k denote the integer in (117). Then $k \geq 2$ and $k > 12\bar{\tau}^2 L(1/\delta + 2/\epsilon)$, and hence

$$\frac{12\bar{\tau}^2 L}{k} \ < \ \left(\frac{1}{\delta} + \frac{2}{\epsilon} \right)^{-1} \ \leq \ \min\left\{ \delta, \frac{\epsilon}{2} \right\}. \tag{128}$$

Suppose now for contradiction, that RUFW does not terminate at any iteration index $j \leq k$, so that it performs Step 4 at iterations $1, \ldots, k$ and the stopping test (113) fails at every iteration index $j \leq k$. Both relations in (119), the fact that $\bar{\tau} \geq 1$, and (128) imply that there exists an iteration index $i \in \{1, \ldots, k\}$ such that

$$|\langle G_i, X_i \rangle| \overset{(119)}{\leq} \epsilon_i(\bar{\tau}) \overset{(125)}{\leq} \frac{12\bar{\tau}^2 L}{k} \overset{(128)}{<} \delta,$$
$$\lambda_i^{\min}(G_i) \overset{(119)}{\geq} \frac{-2\epsilon_i(\bar{\tau})}{\bar{\tau}} \overset{(125)}{\geq} -\frac{24\bar{\tau} L}{k} \ \geq \ -2 \cdot \frac{12\bar{\tau}^2 L}{k} \overset{(128)}{>} -\epsilon.$$

Thus, the stopping test (113) holds at the iteration index $i \leq k$, which contradicts the assumption that RUFW does not terminate at any iteration index $j \leq k$. Hence, RUFW terminates at some iteration index $j \leq k$ with output $\bar{X} = X_j$, which satisfies (109) in view of the stopping test (113) and satisfies (110) in view of part (a). □

B U-Space and Factorized Machinery

This appendix collects the material used to implement the descent step (111) of the RUFW framework of Appendix A.2 in the factorized U-space. It is divided into two parts. Subsection B.1 presents the ADAP-FISTA method while Subsection B.2 presents the ADAP-AIPP method, whose subproblems are solved by ADAP-FISTA.

B.1 ADAP-FISTA

Let $\mathbb{E}$ be a finite-dimensional inner product space and let $\Omega \subseteq \mathbb{E}$ be a nonempty closed convex set. This section presents the ADAP-FISTA method for the problem

$$\min\{\psi(U) : U \in \Omega\} \tag{129}$$

where ψ and Ω are assumed to satisfy the following assumptions:

(B1) ψ is a real-valued function that is differentiable on $\mathbb{E}$ and whose gradient is locally Lipschitz continuous in the sense of (8);

(B2) Ω is closed, possibly unbounded, convex set that admits an easily computable projection;

(B3) a point $X_0 \in \Omega$ is given such that the sublevel set $\mathcal{S}(X_0; \psi)$ has finite diameter $D(X_0; \psi)$, where $\mathcal{S}(\cdot;\cdot)$ and $D(\cdot;\cdot)$ are as in (5) and (6), respectively.

Problem: Given (ψ, Ω, X_0) satisfying **(B1)-(B3)**, the goal of ADAP-FISTA is to find a pair $(\bar{Y}, \bar{V}) \in \Omega \times \mathbb{E}$ such that

$$\bar{V} \in \nabla\psi(\bar{Y}) + \mathcal{N}_\Omega(\bar{Y}), \qquad \|\bar{V}\| \leq \sigma\|\bar{Y} - X_0\|, \qquad \psi(\bar{Y}) \leq \psi(X_0). \tag{130}$$

The formal description of ADAP-FISTA is now given below.

ADAP-FISTA Method

Universal Parameters: $\sigma > 0$ and $\chi \in (0, 1)$.
Input: initial point $\bar{Y}_0 \in \Omega$, pair $\mu > 0$, $L_0 > \mu$, and a function ψ and a set Ω as in **(B1)-(B3)**

0. set $Y_0 = X_0 = \bar{Y}_0$, $A_0 = 0$, $\tau_0 = 1$, and $i = 1$;

1. set $L = L_{i-1}$;

2. compute

$$a = \frac{\tau_{i-1} + \sqrt{\tau_{i-1}^2 + 4\tau_{i-1}A_{i-1}(L-\mu)}}{2(L-\mu)}, \quad \tilde{X} = \frac{A_{i-1}Y_{i-1} + aX_{i-1}}{A_{i-1} + a}, \tag{131}$$

$$Y := \operatorname*{arg\,min}_{U \in \Omega} \left\{ \ell_\psi(U; \tilde{X}) + \frac{L}{2}\|U - \tilde{X}\|^2 \right\}. \tag{132}$$

If the inequality

$$\ell_\psi(Y; \tilde{X}) + \frac{(1-\chi)L}{2}\|Y - \tilde{X}\|^2 \geq \psi(Y) \tag{133}$$

holds then set

$$a_{i-1} = a, \quad \tilde{X}_{i-1} = \tilde{X}, \quad Y_i = Y, \quad L_i = L \tag{134}$$

and go to step 3; else set $L \leftarrow 2L$ and repeat step 2;

3. compute

$$\begin{aligned} &A_i = A_{i-1} + a_{i-1}, \quad \tau_i = \tau_{i-1} + a_{i-1}\mu, \\ &X_i = \frac{1}{\tau_i}\left[\mu a_{i-1}Y_i + \tau_{i-1}X_{i-1} - a_{i-1}(L_i - \mu)(\tilde{X}_{i-1} - Y_i)\right]; \end{aligned} \tag{135}$$

4. if the following three inequalities hold

$$\|Y_i - \bar{Y}_0\|^2 \geq \chi A_i L_i \|Y_i - \tilde{X}_{i-1}\|^2, \quad \psi(Y_i) \leq \psi(\bar{Y}_0) \tag{136}$$

$$\|Y_i - X_i\| \leq \|Y_i - \bar{Y}_0\| \tag{137}$$

hold, then go to step 5; otherwise, stop with **failure**;

5. compute

$$V_i = \nabla\psi(Y_i) - \nabla\psi(\tilde{X}_{i-1}) + L_i(\tilde{X}_{i-1} - Y_i); \tag{138}$$

if the inequality

$$\|V_i\| \leq \sigma\|Y_i - \bar{Y}_0\| \tag{139}$$

holds then stop with **success** and output $(\bar{Y}, \bar{V}, \bar{L}) := (Y_i, V_i, L_i)$; otherwise, $i \leftarrow i + 1$ and go to step 1.

It is easy to see that the update for Y_i in (148) is equivalent to projecting the point $\tilde{X}_{i-1} - (1/L_i)\nabla\psi(\tilde{X}_{i-1})$ onto the set Ω. ADAP-FISTA takes a parameter $\mu > 0$ as input, even when we do not know in advance whether ψ is μ-strongly convex. The method therefore checks the inequalities in (136) and (137) at each iteration. If any of these inequalities is violated, the method stops with failure. When ψ is μ-strongly convex, these inequalities always hold, and ADAP-FISTA returns a triple satisfying (139) within a logarithmic number of iterations. To accommodate objectives with locally Lipschitz continuous gradients, we use different inequalities from those in earlier adaptive variants of FISTA [38, 54].

Before stating the main complexity result of ADAP-FISTA, we introduce the quantities which appear in the complexity result:

$$D_0 := 6D(X_0; \psi) + 1, \quad G_0 := \nabla(D_0/2; \psi), \quad (\text{lip})_0 := [\text{LIP}](D_0; \psi) \tag{140}$$

where $D(\cdot;\cdot)$, $\nabla(\cdot;\cdot)$, and [LIP] are as in (6), (10), and (9), respectively.

The quantities above are finite. Indeed, **(B3)** implies that D_0 is finite. Since $\nabla\psi$ is continuous, it follows that $G_0 < \infty$. Since $\nabla\psi$ is locally Lipschitz, a finite-cover argument on the compact ball $B[D_0]$ yields that the Lipschitz constant $(\text{lip})_0$ is finite. The main proposition which states ADAP-FISTA's main complexity result is now given. Its proof is given in the next two subsections.

Proposition B.1. *Let*

$$N_0 := \max\{L_0, G_0, (\text{lip})_0\}. \tag{141}$$

where L_0 is input to ADAP-FISTA and G_0 and $(\text{lip})_0$ are as in (140)*. The following statements about ADAP-FISTA hold:*

(a) it always stops (with either success or failure) in at most

$$\mathcal{O}_1\left(\sqrt{\frac{N_0}{\mu}}\log_2^+(N_0) + \log_2^+\left(\frac{N_0}{L_0}\right)\right)$$

iterations/resolvent evaluations;

(b) if it stops successfully, then the pair $(\bar{Y}, \bar{V}) \in \Omega \times \mathbb{E}$ output by ADAP-FISTA satisfies the goal (130)*, i.e.,*

$$\bar{V} \in \nabla\psi(\bar{Y}) + \mathcal{N}_\Omega(\bar{Y}), \quad \|\bar{V}\| \le \sigma\|\bar{Y} - \bar{Y}_0\|, \tag{142}$$

$$\psi(\bar{Y}) \le \psi(\bar{Y}_0), \tag{143}$$

and the Lipschitz estimate $\bar{L}$ that it outputs satisfies

$$L_0 \le \bar{L} \le \max\left\{L_0, 12G_0, \frac{2(\text{lip})_0}{1-\chi}\right\}; \tag{144}$$

(c) if ψ is μ-strongly convex on $B[D_0]$, then ADAP-FISTA always terminates with success.

We now make one brief remark about the above proposition. In general, the assumption in (c) that ψ is μ-strongly convex on $B[D_0]$ is not directly verifiable since we do not know D_0. However, we will see in the next section that it will play an important role in guaranteeing that an inner loop of our other method ADAP-AIPP always terminates.

B.1.1 Proof of Proposition B.1(a)-(b)

This subsection is dedicated to proving Proposition B.1(a)-(b). We first define several quantities which arise in our technical analysis:

$$\zeta := (\text{lip})_0 + \max\left\{L_0, 12G_0, \frac{2(\text{lip})_0}{1-\chi}\right\}, \quad Q := \sqrt{\frac{4}{\mu}}\sqrt{\max\left\{L_0, 12G_0, \frac{2(\text{lip})_0}{1-\chi}\right\}}. \tag{145}$$

Also note that since $\nabla\psi$ is $(\text{lip})_0$-Lipschitz on $B[D_0]$ it follows from that

$$\psi(X'') \;\le\; \psi(X') + \langle\nabla\psi(X'), X'' - X'\rangle + \frac{(\text{lip})_0}{2}\|X'' - X'\|^2 \qquad \forall\, X', X'' \in B[D_0]. \tag{146}$$

We start off with a key lemma which characterizes properties of the output of the loop in step 2 of ADAP-FISTA.

Lemma B.2. *Let $i \ge 1$ be an iteration index of ADAP-FISTA. If the line-search in the loop in step 2 terminates, then the following statements then hold:*

(a)

$$a_{i-1} = \frac{\tau_{i-1} + \sqrt{\tau_{i-1}^2 + 4\tau_{i-1}A_{i-1}(L_i - \mu)}}{2(L_i - \mu)}, \quad \tilde{X}_{i-1} = \frac{A_{i-1}Y_{i-1} + a_{i-1}X_{i-1}}{A_{i-1} + a_{i-1}}, \tag{147}$$

$$Y_i := \arg\min_{U\in\Omega}\left\{\ell_\psi(U;\tilde{X}_{i-1}) + \frac{L_i}{2}\|U - \tilde{X}_{i-1}\|^2\right\}; \tag{148}$$

(b) it also holds that

$$\tau_{i-1} = 1 + A_{i-1}\mu, \quad \frac{\tau_{i-1}A_i}{a_{i-1}^2} = L_i - \mu \tag{149}$$

$$A_i L_i \geq \max\left\{\frac{i^2}{4}, \left(1+Q^{-1}\right)^{2(i-1)}\right\}. \tag{150}$$

Proof. (a) The proof of (a) is immediate from relations (131) and (132) and the way a_{i-1}, $\tilde{X}_{i-1}$, Y_i, and L_i are set in (134) at the end of the loop in step 2 of ADAP-FISTA.

(b) The first equality in (149) follows directly from both of the relations in (135). The second equality in (149) follows immediately from the definition of a_{i-1} in (147) and the first relation in (135). The proof of the lower bound on A_iL_i in (150) follows from a near identical argument to the proof of Lemma A.4 in [53]. □

We now start off with a key technical lemma which is instrumental for showing that the line-search performed in step 2 of ADAP-FISTA always terminates.

Lemma B.3. *Let $\theta > 0$, D_0 be as in* (140)*, and $\hat{X} \in B[D_0/2]$. For $\hat{L} > 0$, define*

$$Y_{\hat{L}}^+ := \arg\min_{U\in\Omega}\left\{\ell_\psi(U;\hat{X}) + \frac{\hat{L}}{2}\|U-\hat{X}\|^2\right\}. \tag{151}$$

If

$$\hat{L} \;\geq\; \max\left\{6\|\nabla\psi(\hat{X})\|,\;\frac{(\mathrm{lip})_0}{\theta}\right\} \geq \max\left\{\frac{6\|\nabla\psi(\hat{X})\|}{D_0},\;\frac{(\mathrm{lip})_0}{\theta}\right\}, \tag{152}$$

then $Y_{\hat{L}}^+ \in B[D_0]$ and

$$\ell_\psi(Y_{\hat{L}}^+;\hat{X}) + 0.5\theta\hat{L}\|Y_{\hat{L}}^+ - \hat{X}\|^2 \;\geq\; \psi(Y_{\hat{L}}^+). \tag{153}$$

Proof. It follows from the fact that $X_0 \in \mathcal{S}(X_0;\psi)$, **(B3)**, and the definition of D_0 in (140) that $\|X_0\| \leq D_0/6$. It is easy to see from this fact, (151), the nonexpansivity of the projection operator defined in (4), the assumption on $\hat{L}$ in (152), the fact that $D_0 \geq 1$, and the fact that $\hat{X} \in B[D_0/2]$ that:

$$\begin{aligned}\|Y_{\hat{L}}^+\| &\overset{(151)}{=} \left\|\mathrm{proj}_\Omega\left(\hat{X} - \frac{1}{\hat{L}}\nabla\psi(\hat{X})\right) - \mathrm{proj}_\Omega(X_0) + \mathrm{proj}_\Omega(X_0)\right\| \leq \left\|\hat{X} - \frac{1}{\hat{L}}\nabla\psi(\hat{X}) - X_0\right\| + \|X_0\| \\ &\leq \|\hat{X} - X_0\| + \frac{1}{\hat{L}}\left\|\nabla\psi(\hat{X})\right\| + \|X_0\| \overset{(152)}{\leq} \frac{4D_0}{6} + \frac{D_0}{6} + \frac{D_0}{6} = D_0.\end{aligned}$$

Hence, $Y_{\hat{L}}^+ \in B[D_0]$. Since $\hat{X}$ and $Y_{\hat{L}}^+$ belong to $B(D_0)$, it follows from (146) with $X'' = Y_{\hat{L}}^+$ and $X' = \hat{X}$ that

$$\psi(Y_{\hat{L}}^+) \leq \ell_\psi(Y_{\hat{L}}^+;\hat{X}) + \frac{(\mathrm{lip})_0}{2}\|Y_{\hat{L}}^+ - \hat{X}\|^2$$

which in view of the assumption on $\hat{L}$ in (152) implies the second conclusion of the lemma. □

The following lemma shows that the line-search in step 2 of ADAP-FISTA always terminates and provides a lower and upper bound on the sequence of Lipschitz estimates $\{L_i\}$.

Lemma B.4. *For any iteration index $i \geq 1$ generated by ADAP-FISTA, the line-search in step 2 always terminates, the iterates X_{i-1}, Y_{i-1}, and $\tilde{X}_{i-1}$ all lie in $B[D_0/2]$, and*

$$L_0 \leq L_i \leq \max\left\{L_0, 12G_0,\;\frac{2(\mathrm{lip})_0}{1-\chi}\right\}. \tag{154}$$

Proof. We proceed by induction. We do not include the argument for the $i = 1$ case since it follows from the same argument as below and by using the fact that $X_0 = Y_0 \in B[D_0/6]$.

Assume then that the statement holds for an iteration index $i \geq 1$ generated by ADAP-FISTA. We show that these facts hold also for the $(i+1)$-st iteration index. Since the method generates the $(i+1)$-st iteration index, the inequalities checked in step 4 of the i-th iteration are satisfied, i.e.,

$$\psi(Y_i) \leq \psi(X_0), \quad \|Y_i - X_i\| \leq \|Y_i - X_0\|. \tag{155}$$

Hence, $Y_i \in \mathcal{S}(X_0;\psi)$ and by (140), $\|Y_i\| \leq D_0/6$. Using this fact, (155), and $\|X_0\| \leq D_0/6$, we obtain

$$\|X_i\| \leq \|Y_i\| + \|Y_i - X_i\| \overset{(155)}{\leq} \|Y_i\| + \|Y_i - X_0\| \leq 2\|Y_i\| + \|X_0\| \leq \frac{D_0}{2}. \tag{156}$$

Thus, $X_i, Y_i \in B[D_0/2]$, and every tentative point $\tilde{X}$ computed in (131) of the $(i+1)$-st iteration also lies in $B[D_0/2]$.

This fact, the way Y is computed in (132), the last conclusion in Lemma B.3, and the check in (133) imply that if the Lipschitz estimate L generated in step 2 of ADAP-FISTA becomes large enough, then the line-search in step 2 of the $(i+1)$-st iteration must terminate. This fact and (134) imply that $a_i, \tilde{X}_i, Y_{i+1}$, and L_{i+1} are all generated and that $\tilde{X}_i$ lies in $B[D_0/2]$. We now show that the inequalities in (154) hold for L_{i+1}. It is immediate from the way that L is updated in step 2 and the way L_{i+1} is set in (134) that $L_0 \leq L_{i+1}$. We now show by contradiction that the second relation in (154) holds for L_{i+1}. Suppose then that the line search in step 2 of ADAP-FISTA's $(i+1)$-st iteration terminates with L_{i+1} that satisfies

$$L_{i+1} > \max\left\{L_0, 12G_0, \ \frac{2(\mathrm{lip})_0}{1-\chi}\right\}. \tag{157}$$

It then follows that (133) was not satisfied with $L = L_{i+1}/2$ and the $\tilde{X}$ and Y in (131) and (132) corresponding to this L. It then follows from (157), the fact that $\tilde{X} \in B[D_0/2]$, and the definition of G_0 in (140) that

$$L = \frac{L_{i+1}}{2} > \max\left\{6G_0, \ \frac{(\mathrm{lip})_0}{1-\chi}\right\} \overset{(140)}{\geq} \max\left\{6\|\nabla\psi(\tilde{X})\|, \ \frac{(\mathrm{lip})_0}{1-\chi}\right\}.$$

Hence, it follows from this inclusion, the update for Y in (132), and Lemma B.3 with $\theta = (1-\chi)$, $\hat{X} = \tilde{X}$, $\hat{L} = L$, and $Y_{\hat{L}}^{+} = Y$ that $Y \in B[D_0]$ and

$$\ell_\psi(Y;\tilde{X}) + \frac{(1-\chi)L}{2}\|Y - \tilde{X}\|^2 \geq \psi(Y) \tag{158}$$

which contradicts that (133) was not satisfied with $L = L_{i+1}/2$, $\tilde{X}$, and Y. □

Lemma B.5. *For any iteration index $i \geq 1$ where ADAP-FISTA does not fail in its step 4, it holds that*

$$Y_i \in B[D_0] \tag{159}$$

$$V_i \in \nabla\psi(Y_i) + \mathcal{N}_\Omega(Y_i), \quad \|V_i\| \leq \zeta\|Y_i - \tilde{X}_{i-1}\|. \tag{160}$$

Proof. Let $i \geq 1$ be an iteration index where ADAP-FISTA does not fail in its step 4. It follows that the second relation in (136) must hold. This observation and relation (148) imply that $Y_i \in \mathcal{S}(X_0;\psi)$. It then follows from this fact and (140) that $Y_i \in B[D_0/6] \subseteq B[D_0]$, which shows (159).

It follows from the fact that ADAP-FISTA did not fail in its step 4 that it reaches step 5 and V_i in (138) is computed. It follows from (148) and its associated optimality condition that $0 \in \nabla\psi(\tilde{X}_{i-1}) + \mathcal{N}_\Omega(Y_i) - L_i(\tilde{X}_{i-1} - Y_i)$, which in view of the definition of V_i in (138) implies the inclusion in (160).

Lemma B.4 and (159) imply that $\tilde{X}_{i-1}$ and Y_i both lie in $B[D_0]$. It follows from these observations, triangle inequality, the fact that $\nabla\psi$ is $(\mathrm{lip})_0$-continuous, the bound (154) on L_i, and the definition of ζ in (145) that

$$\begin{aligned}\|V_i\| &\overset{(138)}{\leq} \|\nabla\psi(Y_i) - \nabla\psi(\tilde{X}_{i-1})\| + L_i\|\tilde{X}_{i-1} - Y_i\| \\ &\leq (\mathrm{lip})_0\|Y_i - \tilde{X}_{i-1}\| + L_i\|Y_i - \tilde{X}_{i-1}\| \overset{(154)}{\leq} \zeta\|Y_i - \tilde{X}_{i-1}\|\end{aligned}$$

which immediately implies that the inequality in (160) holds. □

The following proposition states that ADAP-FISTA terminates with success or failure in a logarithmic number of iterations.

Proposition B.6. *ADAP-FISTA always stops (with either success or failure) and does so by performing at most*

$$\left\lceil (1+Q)\log_2^+\left(\frac{\zeta^2}{\chi\sigma^2}\right) + 1\right\rceil + \left\lceil \log_2^+\left(\max\left\{\frac{6G_0}{L_0}, \ \frac{(\mathrm{lip})_0}{(1-\chi)L_0}\right\}\right)\right\rceil \tag{161}$$

iterations/resolvent evaluations, where D_0, G_0, and $(\mathrm{lip})_0$ are as in (140) and ζ and Q are as in (145)

Proof. Let l denote the first quantity in (161). Using this definition and the inequality $\log(1+\alpha) \geq \alpha/(1+\alpha)$ for any $\alpha \geq 0$, it is easy to verify that

$$\left(1+Q^{-1}\right)^{2(l-1)} \geq \frac{\zeta^2}{\chi\sigma^2}. \tag{162}$$

We claim that ADAP-FISTA terminates with success or failure in at most l iterations. Indeed, it suffices to show that if ADAP-FISTA has not stopped with success or failure before the l-th iteration and does not fail during the l-th iteration, then it must stop successfully at the l-th iteration. So, assume that ADAP-FISTA has not stopped with failure up to (and including) the l-th iteration. In view of step 4 of ADAP-FISTA, it follows that (136) holds with $i = l$.

This observation, inequality in (160) with $i = l$, the first inequality in (136) with $i = l$, inequality (150) with $i = l$, and (162) then imply that

$$\|Y_l - \bar{Y}_0\|^2 \overset{(136)}{\geq} \chi A_l L_l \|Y_l - \tilde{X}_{l-1}\|^2 \overset{(160)}{\geq} \frac{\chi}{\zeta^2} A_l L_l \|V_l\|^2 \overset{(150)}{\geq} \frac{\chi}{\zeta^2}\left(1+Q^{-1}\right)^{2(l-1)} \|V_l\|^2 \overset{(162)}{\geq} \frac{1}{\sigma^2}\|V_l\|^2, \tag{163}$$

and hence that (139) is satisfied. In view of Step 5 of ADAP-FISTA, the method must successfully stop at the end of the l-th iteration. We have thus shown that the above claim holds. Moreover, in view of (154), it follows that the second term in (161) is a bound on the total number of times L_i is multiplied by 2 and step 2 is repeated. Since exactly one resolvent evaluation occurs every time step 2 is executed, the desired conclusion follows. □

Proof of Proposition B.1(a)-(b). (a) It follows from the definition of ζ in (145) that $\mathcal{O}(\log_2^+(\zeta^2)) = \mathcal{O}(\log_2^+(\zeta)) = \mathcal{O}\left(\log_2^+\left(\max\left\{L_0, (\mathrm{lip})_0, G_0\right\}\right)\right)$. The result immediately follows from this observation, Proposition B.6, and the definitions of Q and N_0 in (145) and (141), respectively.

(b) It follows immediately from the termination criterion (139) in step 5 of ADAP-FISTA and the inclusion in (160) that (142) must hold if ADAP-FISTA stops successfully.

To see (143), first observe that since ADAP-FISTA stops successfully, the second relation in (136) held in ADAP-FISTA's last iteration. Hence, this observation together with relation (148) imply that $Y \in \Omega$ and $\psi(\bar{Y}) \leq \psi(\bar{Y}_0)$. The bound (144) follows immediately from the fact that relation (154) holds for all iteration indices $i \geq 1$ generated by ADAP-FISTA. □

B.1.2 Proof of Proposition B.1(c)

This subsection proves Proposition B.1(c). Throughout, we assume that ψ is μ-strongly convex on $B[D_0]$. The following lemma proves key facts of the i-th iteration of ADAP-FISTA.

Lemma B.7. *For every iteration $i \geq 1$ of ADAP-FISTA, the following statements hold:*

(a) for every $X \in \Omega \cap B[D_0]$, we have

$$\left(A_{i-1}[\psi(Y_{i-1}) - \psi(X)] + \frac{\tau_{i-1}}{2}\|X - X_{i-1}\|^2\right) - \left(A_i[\psi(Y_i) - \psi(X)] + \frac{\tau_i}{2}\|X - X_i\|^2\right) \geq \frac{\chi A_i L_i}{2}\|Y_i - \tilde{X}_{i-1}\|^2;$$

(b) for every $X \in \Omega \cap B[D_0]$, we have

$$A_i[\psi(Y_i) - \psi(X)] + \frac{\tau_i}{2}\|X - X_i\|^2 \leq \frac{1}{2}\|X - \bar{Y}_0\|^2 - \frac{\chi}{2}\sum_{j=1}^{i} A_j L_j \|Y_j - \tilde{X}_{j-1}\|^2. \tag{164}$$

(c) the method does not fail at iteration i.

Proof. (a) For every iteration $i \geq 1$ of ADAP-FISTA and $X \in \Omega$, define

$$\gamma_i(X) := \ell_\psi(Y_i, \tilde{X}_{i-1}) + \langle (L_i - \mu)(\tilde{X}_{i-1} - Y_i), X - Y_i\rangle + \frac{\mu}{2}\|Y_i - \tilde{X}_{i-1}\|^2 + \frac{\mu}{2}\|X - Y_i\|^2. \tag{165}$$

It is now easy to see from similar arguments as in Lemma A.7 of [53] that for every iteration $i \geq 1$ and $X \in \Omega$, we have

$$\begin{aligned} A_{i-1}\gamma_i(Y_{i-1}) + a_{i-1}\gamma_i(X) + \frac{\tau_{i-1}}{2}\|X_{i-1} - X\|^2 - \frac{\tau_i}{2}\|X_i - X\|^2 \\ \geq A_i\psi(Y_i) + \frac{\chi A_i L_i}{2}\|Y_i - \tilde{X}_{i-1}\|^2. \end{aligned} \tag{166}$$

To get an inequality that telescopes, we now show that $\gamma_i(X) \leq \psi(X)$ for all $X \in \Omega \cap B[D_0]$. To show the claim for every iteration $i \geq 1$, define

$$\tilde{\gamma}_i(X) := \ell_\psi(X; \tilde{X}_{i-1}) + \frac{\mu}{2}\|X - \tilde{X}_{i-1}\|^2. \tag{167}$$

Now observe that Lemma B.4 implies that $\tilde{X}_{i-1} \in B[D_0/2] \subseteq B[D_0]$. It follows from this observation and the assumption that ψ is μ-strongly convex on $B[D_0]$ that $\tilde{\gamma}_i(X) \leq \psi(X)$ for all $X \in B[D_0]$. The rest of the argument showing that $\gamma_i(X) \leq \psi(X)$ for all $X \in \Omega \cap B(D_0)$ is identical to the one in Lemma A.8 in [53].

Lemma B.4 imply that $Y_{i-1} \in B[D_0/2] \cap \Omega \subseteq B[D_0] \cap \Omega$. This observation, the fact that $\gamma_i(X) \leq \psi(X)$ for all $X \in B[D_0] \cap \Omega$, and (166) imply that

$$\begin{aligned} A_{i-1}\psi(Y_{i-1}) + a_{i-1}\psi(X) + \frac{\tau_{i-1}}{2}\|X_{i-1} - X\|^2 - \frac{\tau_i}{2}\|X_i - X\|^2 \\ \geq A_i\psi(Y_i) + \frac{\chi A_i L_i}{2}\|Y_i - \tilde{X}_{i-1}\|^2 \end{aligned} \tag{168}$$

holds for all $X \in B[D_0] \cap \Omega$. The relation in part (a) then immediately follows from subtracting $A_i\psi(X)$ from both sides of (168) and using the fact that (135) implies $A_i = A_{i-1} + a_{i-1}$.

(b) It follows from summing the inequality of in part (a) from 1 to i, and using the facts that $X_0 = \bar{Y}_0$, $A_0 = 0$ and $\tau_0 = 1$ that

$$A_i[\psi(Y_i) - \psi(X)] + \frac{\tau_i}{2}\|X - X_i\|^2 \leq \frac{1}{2}\|X - \bar{Y}_0\|^2 - \frac{\chi}{2}\sum_{j=1}^{i} A_j L_j \|Y_j - \tilde{X}_{j-1}\|^2.$$

which immediately implies that (164) holds.

(c) To show that ADAP-FISTA does not fail at its iteration i, we show that all three inequalities checked in (136) and (137) in its step 4 must hold. We first show that $\psi(Y_i) \leq \psi(\bar{Y}_0)$. First observe that it is easy to see that $\bar{Y}_0 \in B[D_0/6] \cap \Omega \subseteq B[D_0] \cap \Omega$ in view of the definition of D_0 in (140). Thus, using (164) with $X = \bar{Y}_0$ and the facts that the sequences τ_i, A_i, and L_i are nonnegative, we see that

$$A_i[\psi(Y_i) - \psi(X_0)] \leq \frac{1}{2}\|\bar{Y}_0 - \bar{Y}_0\|^2 = 0$$

Hence, diving both sides by A_i, we see that $\psi(Y_i) \leq \psi(\bar{Y}_0)$. To conclude the proof, we now show that

$$\|Y_i - X_i\| \leq \|Y_i - X_0\|, \quad \|Y_i - X_0\|^2 \geq \chi A_i L_i \|Y_i - \tilde{X}_{i-1}\|^2. \tag{169}$$

It follows from above conclusion, (148), and the definition of D_0 in (140) that $Y_i \in B[D_0] \cap \Omega$. Hence, using (164) with $X = Y_i$ and the fact that $\tau_i \geq 1$, we see that $\|Y_i - X_i\| \leq \|Y_i - \bar{Y}_0\|$. Likewise also using (164) with $X = Y_i$ but keeping the last term on the right-hand side of (164), we see that $\|Y_i - \bar{Y}_0\|^2 \geq \chi A_i L_i \|Y_i - \tilde{X}_{i-1}\|^2$. Hence, the three inequalities checked in (136) and (137) must hold at iteration i, which implies that ADAP-FISTA does not fail in its i-ith iteration. □

We are now ready to prove Proposition B.1(c).

Proof of Proposition B.1(c). It follows from Lemma B.7(c) that, for any iteration $i \geq 1$, ADAP-FISTA does not terminate with failure at step 4 when applied to a function ψ that is μ-strongly convex on $B[D_0]$ and satisfies assumptions **(B1)**–**(B3)**. Hence, it follows from Proposition B.1(a)-(b) that ADAP-FISTA must always terminate successfully and return a pair $(\bar{Y}, \bar{V}) \in \Omega \times \mathbb{E}$ satisfying (142) and (143) and a Lipschitz estimate $\bar{L}$ satisfying (144). □

B.2 ADAP-AIPP Method and Complexity

Let $\mathbb{E}$ be a finite-dimensional inner product space. This section considers

$$g_* := \min\{g(U) : \ U \in \Omega\},$$

under the following assumptions:

(C1) g is a real-valued function that is differentiable on $\mathbb{E}$ and whose gradient is locally Lipschitz continuous in the sense of (8);

(C2) g is bounded below on Ω;

(C3) Ω is closed, possibly unbounded, convex set that admits an easily computable projection;

(C4) a point $W_0 \in \Omega$ is given such that the sublevel set $\mathcal{S}(W_0; g)$ has finite diameter $D(W_0; g)$, where $\mathcal{S}(\cdot;\cdot)$ and $D(\cdot;\cdot)$ are as in (5) and (6), respectively.

Problem: Given (g, Ω, W_0) satisfying **(C1)-(C4)** and a tolerance $\bar{\rho} \in \mathbb{R}_{++}$, ADAP-AIPP seeks $(W, R) \in \Omega \times \mathbb{E}$ such that

$$R \in \nabla g(W) + \mathcal{N}_\Omega(W), \qquad \|R\| \leq \bar{\rho}.$$

At its j-th iteration, calls ADAP-FISTA several times until it sucessfully finds an approximate stationary point of

$$\min_{U \in \Omega} \left\{ \psi_\eta(U) := \eta g(U) + \tfrac{1}{2}\|U - W_{j-1}\|^2 \right\} \tag{170}$$

where $\eta > 0$ is a prox stepsize that is halved after an unsuccessful ADAP-FISTA call.

ADAP-AIPP Method

Universal Parameters (used by ADAP-FISTA): $\sigma \in (0, 1/2)$, $\chi \in (0, 1)$.

Input: an initial point $W_0 \in \Omega$, a function g satisfying **(C1)-(C4)**, a set Ω, prox stepsize $\eta_0 > 0$, an initial Lipschitz estimate M_0, and tolerance $\bar{\rho} > 0$.

Step 0. Set $j \leftarrow 1$ and $\eta \leftarrow \eta_0$.

Step 1. Call ADAP-FISTA with input

$$\bar{Y}_0 = W_{j-1}\,, \quad \mu = \frac{1}{2}\,, \quad L_0 = M_{j-1}\,, \quad \psi(\cdot) = \eta g(\cdot) + \frac{1}{2}\|\cdot - W_{j-1}\|^2, \quad \Omega = \Omega. \tag{171}$$

Step 2. If the above ADAP-FISTA call terminates successfully, then let (W_j, V_j, M_j) be its output and set

$$\eta_j \leftarrow \eta, \quad R_j \;\leftarrow\; \frac{V_j + W_{j-1} - W_j}{\eta_j}; \tag{172}$$

else, set $\eta \leftarrow \eta/2$ and return to Step **1**.

Step 3. If $\|R_j\| \leq \bar{\rho}$, then output $(\bar{W}, \bar{R}) \leftarrow (W_j, R_j)$ and **stop**; **else**, set $j \leftarrow j + 1$ and return to Step **1**.

We now define several quantities which arise in ADAP-AIPP's technical analysis and complexity result. Let

$$\mathcal{D}_j := 6D(W_{j-1}; g) + 1, \quad \mathcal{D}_0 := \mathcal{D}_1 \tag{173}$$

where $j \geq 1$ is an iteration index and $D(\cdot;\cdot)$ is as in (6). For every $j \geq 0$, we define

$$\nabla_j := \nabla(\mathcal{D}_j/2; g), \quad (\mathrm{Lip})_j := [\mathrm{LIP}](\mathcal{D}_j; g) \tag{174}$$

where $\nabla(\cdot;\cdot)$, and $[\mathrm{LIP}](\cdot;\cdot)$ are as in (10) and (9), respectively. We also let

$$C_\sigma = \frac{2(1-\sigma)^2}{(1-2\sigma)^2} \tag{175}$$

where $\sigma \in (0, 1/2)$ is an universal input to ADAP-AIPP.

The next lemma states useful facts about the sublevel sets of the prox function in (170).

Lemma B.8. *Let $j \geq 1$ be an iteration index generated by ADAP-AIPP and consider $\psi_\eta(\cdot)$ as in* (170)*, i.e., $\psi_\eta(\cdot) := \eta g(\cdot) + 0.5\|\cdot - W_{j-1}\|^2$ where $\eta > 0$ is a prox stepsize. The following facts then hold:*

(a) the diameter $D(W_{j-1}; \psi_\eta) \leq D(W_{j-1}; g)$;

(b) $\nabla\psi_\eta(\cdot)$ is $1 + \eta(\mathrm{Lip})_j$ Lipschitz continuous on $B[\mathcal{D}_j]$;

(c) it holds that

$$\sup\{\|\nabla\psi_\eta(U)\| :\; U \in B[\mathcal{D}_j/2]\} \leq \eta\nabla_j + \mathcal{D}_j. \tag{176}$$

Proof. (a) Let $U' \in \mathcal{S}(W_{j-1};\psi_\eta)$. Hence, it follows from the definition of $\mathcal{S}(\cdot;\cdot)$ in (5) that $\psi_\eta(U') \le \psi_\eta(W_{j-1})$. It then follows from the definition of $\psi_\eta(\cdot)$ that $\eta g(U') \le \eta g(U') + 0.5\|U' - W_{j-1}\|^2 \le \eta g(W_{j-1})$, which implies that $U' \in \mathcal{S}(W_{j-1};g)$. Thus, it follows that $\mathcal{S}(W_{j-1},\psi_\eta) \subseteq \mathcal{S}(W_{j-1};g)$ which implies that $D(W_{j-1};\psi_\eta) \le D(W_{j-1};g)$ in view of the definition of $D(W_{j-1};g)$ in (6).

(b) It is easy to see that

$$\nabla\psi_\eta(U) = \eta\nabla g(U) + (U - W_{j-1}). \tag{177}$$

The result then follows from (177) and the fact that ∇g is $(\mathrm{Lip})_j$-Lipschitz continuous on $B[\mathcal{D}_j]$.

(c) It is immediate from (177) and the definition of ∇_j in (174) that

$$\sup\{\|\nabla\psi_\eta(U)\| :\ U \in B[\mathcal{D}_j/2]\} \le \eta\nabla_j + \sup_{U\in B[\mathcal{D}_j/2]} \|U\| + \|W_{j-1}\| \tag{178}$$

$$\le \eta\nabla_j + \frac{\mathcal{D}_j}{2} + \frac{\mathcal{D}_j}{6} \le \eta\nabla_j + \mathcal{D}_j \tag{179}$$

where the second inequality follows from the definition of $\mathcal{D}_j$ in (173) and the fact that $W_{j-1} \in \mathcal{S}(W_{j-1};g)$. □

Lemma B.9. *Let $j \ge 1$ be an iteration index of ADAP-AIPP, and define*

$$\Theta_j := \max\{M_{j-1},\ \eta_0\nabla_j + \mathcal{D}_j,\ 1 + \eta_0(\mathrm{Lip})_j\}. \tag{180}$$

Then the following statements hold:

(a) the loop in steps 1 and 2 of the j-th iteration is repeated at most a logarithmic number of times before terminating with a triple $(W_j, V_j, M_j) \in \Omega \times \mathbb{E} \times \mathbb{R}_{++}$ and η_j satisfying

$$\eta_0 \ge \eta_j \ge \min\{\eta_0, 1/(4(\mathrm{Lip})_j)\}, \quad R_j \in \nabla g(W_j) + \mathcal{N}_\Omega(W_j), \tag{181}$$

$$\|V_j\| \le \sigma\|W_j - W_{j-1}\|, \quad \eta_j\,(g(W_{j-1}) - g(W_j)) \ge \frac{1}{2}\|W_j - W_{j-1}\|^2, \tag{182}$$

and

$$M_{j-1} \le M_j \le \max\left\{M_{j-1}, 12(\eta_j\nabla_j + \mathcal{D}_j), \frac{2(1+\eta_j(\mathrm{Lip})_j)}{1-\chi}\right\}; \tag{183}$$

(b) every ADAP-FISTA call made in step 1 within this loop terminates with either success or failure, in at most

$$\mathcal{O}_1\left(\sqrt{\Theta_j}\log_2^+(\Theta_j) + \log_2^+\left(\frac{\Theta_j}{M_{j-1}}\right)\right) \tag{184}$$

ACG iterations/resolvent evaluations;

(c) the inequality holds:

$$g(W_{j-1}) - g(W_j) \ \ge\ \frac{\eta_j}{C_\sigma}\|R_j\|^2. \tag{185}$$

where C_σ is as in (175).

Proof. (a) Let $j \ge 1$ be an iteration index of ADAP-AIPP and consider its j-th iteration. It follows from step 1 and Lemma B.8(b) that ADAP-AIPP calls ADAP-FISTA with a function $\psi(\cdot) = \eta g(\cdot) + \frac{1}{2}\|\cdot - W_{j-1}\|^2$ whose gradient is $1 + \eta(\mathrm{Lip})_j$ Lipschitz continuous on $B[\mathcal{D}_j]$. Thus, this function is $(1 - \eta(\mathrm{Lip})_j)$-strongly convex on $B[\mathcal{D}_j]$. Thus, it follows from this observation, the fact that ADAP-FISTA is called with $\mu = 1/2$, and Proposition B.1(c) that if ADAP-FISTA is called with $\psi(\cdot) = \eta g(\cdot) + \frac{1}{2}\|\cdot - W_{j-1}\|^2$ where $\eta \le 1/(2(\mathrm{Lip})_j)$, then the call to ADAP-FISTA must terminate successfully. The first conclusion of part (a) and the first relation in (181) then immediately follow from this observation, the fact that η is halved after every unsuccessful ADAP-FISTA call, and the way η_j is set in (172).

It then follows from Proposition B.1(b), the fact that ADAP-AIPP calls ADAP-FISTA with input as in (171), and the way η_j is set in (172) that ADAP-FISTA terminates with a triple (W_j, V_j, M_j) and η_j satisfying

$$V_j \in \eta_j\nabla g(W_j) + (W_j - W_{j-1}) + \mathcal{N}_\Omega(W_j), \quad \|V_j\| \le \sigma\|W_j - W_{j-1}\| \tag{186}$$

$$\eta_j\,(g(W_{j-1}) - g(W_j)) \ge \frac{1}{2}\|W_j - W_{j-1}\|^2 \tag{187}$$

which immediately implies both relations in (182). The second relation in (181) is immediate from the inclusion in (186) and the definition of R_j in (172). Relation (183) follows from the fact that ADAP-FISTA is called with input as in (171), relation (144), Lemma B.8(a)-(c), and the way that η_j is set in (172).

(b) Let $j \geq 1$ and consider $\psi_\eta := \eta g(\cdot) + 0.5\|\cdot - W_{j-1}\|^2$ where $\eta > 0$. It follows from Lemma B.8(a) and the definition of $\mathcal{D}_j$ in (173) that $D(W_{j-1};\psi_\eta) \leq D(W_{j-1};g) \leq \mathcal{D}_j$. This relation together with (176) then imply that

$$\sup\{\|\nabla\psi_\eta(U)\| :\ U \in B[D(W_{j-1};\psi_\eta)/2]\} \leq \sup\{\|\nabla\psi_\eta(U)\| :\ U \in B[\mathcal{D}_j/2]\} \leq \eta\nabla_j + \mathcal{D}_j.$$

It then follows from these conclusions, Proposition B.1(a), the fact that ADAP-FISTA is called with input as in (171), the fact that the way η is updated in step 2 implies that $\eta \leq \eta_0$ for all η generated by ADAP-AIPP, and the definition of Θ_j in (180) that every ADAP-FISTA call within the loop in steps 1 and 2 performs at most (184) ACG iterations/resolvent evaluations.

(c) Let $\tilde{\sigma} = \sigma/(1-\sigma)$ where $\sigma \in (0, 1/2)$ is an universal input to ADAP-AIPP. It follows from the first relation in (182) and the definition of R_j in (172) that $\|V_j\| \leq \sigma\|W_j - W_{j-1}\| \leq \sigma\|V_j\| + \sigma\|\eta_j R_j\|$. This relation together with the definition of $\tilde{\sigma}$ then imply that

$$\left\|\frac{V_j}{\eta_j}\right\| \leq \tilde{\sigma}\|R_j\|. \tag{188}$$

It then follows from the definition of R_j in (172), relation (188), and multiplying both sides of the second inequality in (182) by $2/\eta_j^2$ that

$$\begin{aligned}
\frac{2}{\eta_j}[g(W_{j-1}) - g(W_j)] &\overset{(182)}{\geq} \left\|\frac{W_{j-1} - W_j}{\eta_j}\right\|^2 \overset{(172)}{=} \left\|R_j - \frac{V_j}{\eta_j}\right\|^2 \\
&\overset{(188)}{\geq} \min_{\|z\| \leq \tilde{\sigma}\|R_j\|} \|R_j - z\|^2 \\
&= \min_{0 \leq t \leq \tilde{\sigma}\|R_j\|} (\|R_j\| - t)^2 \\
&= (1-\tilde{\sigma})^2\|R_j\|^2.
\end{aligned} \tag{189}$$

Relation (185) then follows from multiplying both sides of (189) by $\eta_j/2$, the fact that $\tilde{\sigma} = \sigma/(1-\sigma)$, and the definition of C_σ in (175). □

Lemma B.10. *The sequence $\{\mathcal{D}_j\}$ is nonincreasing, i.e., for any iteration index $j \geq 1$, it holds that*

$$\mathcal{D}_j \leq \mathcal{D}_{j-1} \leq \mathcal{D}_0 \tag{190}$$

where recall $\mathcal{D}_j$ is as in (173).

Proof. Let $j \geq 1$ be an iteration index of ADAP-AIPP and let $U' \in \mathcal{S}(W_j; g)$ where $\mathcal{S}(\cdot;\cdot)$ is as in (5). It follows from this inclusion, (185), and the facts that η_j, C_σ, and $\|R_j\|^2$ are all nonnegative quantities that

$$g(U') \leq g(W_j) \overset{(185)}{\leq} g(W_{j-1}) - \frac{\eta_j}{C_\sigma}\|R_j\|^2 \overset{(175)}{\leq} g(W_{j-1}).$$

Hence, it follows from the above relation that $U' \in \mathcal{S}(W_{j-1}; g)$ and thus $\mathcal{S}(W_j; g) \subseteq \mathcal{S}(W_{j-1}; g)$. It then follows from the definition of $D(\cdot;\cdot)$ in (6) that $D(W_j; g) \leq D(W_{j-1}; g)$. The first inequality in (190) then follows from this relation and the definition of $\mathcal{D}_j$ in (173). The second inequality in (190) then follows from the first inequality in (190), a simple induction argument, and the fact that $\mathcal{D}_1 = \mathcal{D}_0$. □

Lemma B.11. *For every iteration index $j \geq 1$, it holds that*

$$M_0 \leq M_j \leq \max\left\{M_0, 12(\eta_0\nabla_0 + \mathcal{D}_0), \frac{2(1 + \eta_0(\mathrm{Lip})_0)}{1-\chi}\right\}. \tag{191}$$

Proof. The first inequality in (191) is immediate from the first inequality in (183) together with a simple induction argument. To show the second inequality in (191), we also proceed by induction. We do not include the argument for the $j = 1$ case since it follows from a similar argument as below.

Suppose then that the second inequality in (191) holds for an iteration index $j \geq 1$ generated by ADAP-AIPP. We show that it also must hold for the $(j+1)$-st iteration index. It follows from the second inequality in (183) with

$j = j + 1$, the bounds on $\mathcal{D}_{j+1}$ and η_{j+1} in (190) and (181), respectively, and the definitions of ∇_j and $(\mathrm{Lip})_j$ in (174) that

$$\begin{aligned} M_{j+1} &\overset{(183)}{\leq} \max\left\{M_j, 12(\eta_{j+1}\nabla_{j+1} + \mathcal{D}_{j+1}), \frac{2(1+\eta_{j+1}(\mathrm{Lip})_{j+1})}{1-\chi}\right\} \\ &\overset{(181)}{\leq} \max\left\{M_j, 12(\eta_0\nabla_{j+1} + \mathcal{D}_{j+1}), \frac{2(1+\eta_0(\mathrm{Lip})_{j+1})}{1-\chi}\right\} \\ &\overset{(190)}{\leq} \max\left\{M_j, 12(\eta_0\nabla_0 + \mathcal{D}_0), \frac{2(1+\eta_0(\mathrm{Lip})_0)}{1-\chi}\right\}. \end{aligned} \tag{192}$$

The fact that (191) holds for all iteration indices $j \geq 1$ then follows immediately from (192) and the induction hypothesis. □

The following lemma uniformly bounds the quantity Θ_j in (180). This is useful for uniformly bounding the number of gradient evaluations made during each ADAP-FISTA call in step 1.

Lemma B.12. *Let*

$$\bar{\Theta} := \max\left\{M_0, \eta_0\nabla_0 + \mathcal{D}_0, 1 + \eta_0(\mathrm{Lip})_0\right\}. \tag{193}$$

For every iteration index $j \geq 1$, it holds that $\Theta_j = \mathcal{O}(\bar{\Theta})$.

Proof. Let $j \geq 1$ be an iteration index generated by ADAP-FISTA. It follows from (190) and the definitions of $(\mathrm{Lip})_j$ and ∇_j in (174) that $\mathcal{D}_j \leq \mathcal{D}_0$, $(\mathrm{Lip})_j \leq (\mathrm{Lip})_0$, and $\nabla_j \leq \nabla_0$. It is then immediate from this bound, the bound on M_j in (191), the fact that $\chi \in (0,1)$, and (193) that

$$\begin{aligned} \Theta_j &\overset{(180)}{=} \max\left\{M_{j-1},\ \eta_0\nabla_j + \mathcal{D}_j,\ 1 + \eta_0(\mathrm{Lip})_j\right\} \\ &\overset{(190)}{\leq} \max\left\{M_{j-1},\ \eta_0\nabla_0 + \mathcal{D}_0,\ 1 + \eta_0(\mathrm{Lip})_0\right\} \qquad (194) \\ &\overset{(191)}{\leq} \max\left\{M_0, 12\left(\eta_0\nabla_0 + \mathcal{D}_0\right), \frac{2\left(1+\eta_0(\mathrm{Lip})_0\right)}{1-\chi}\right\} \overset{(193)}{=} \mathcal{O}(\bar{\Theta}) \qquad (195) \end{aligned}$$

which immediately implies the result of the lemma. □

The following proposition now states the main complexity results of ADAP-AIPP.

Proposition B.13. *The following statements about ADAP-AIPP hold:*

(a) it terminates with a pair $(\bar{W}, \bar{R}) \in \Omega \times \mathbb{E}$ satisfying

$$\bar{R} \in \nabla g(\bar{W}) + \mathcal{N}_\Omega(\bar{W}), \qquad \|\bar{R}\| \leq \bar{\rho} \tag{196}$$

$$g(\bar{W}) \leq g(W_0); \tag{197}$$

(b) if ℓ denotes its final iteration index, then its total number of ADAP-AIPP iterations/ADAP-FISTA calls is bounded by

$$1 + \frac{C_\sigma \max\left\{\eta_0^{-1}, 4(\mathrm{Lip})_0\right\}}{\bar{\rho}^2}\left[g(W_0) - g(W_\ell)\right] + \left\lceil \log_2^+\left(4\eta_0(\mathrm{Lip})_0\right)\right\rceil \tag{198}$$

where $(\mathrm{Lip})_0$ and C_σ are as in (174) and (175), respectively, and η_0, $\bar{\rho}$, and W_0 are inputs to ADAP-AIPP.

Proof. (a) We first show that ADAP-AIPP must terminate and that its final iteration index $\ell \geq 1$ satisfies

$$\ell \leq 1 + \frac{C_\sigma \max\left\{\eta_0^{-1}, 4(\mathrm{Lip})_0\right\}}{\bar{\rho}^2}\left[g(W_0) - g(W_\ell)\right]. \tag{199}$$

Suppose by contradiction that ADAP-AIPP generates some index $\bar{j}$ such that

$$\bar{j} > 1 + \frac{C_\sigma \max\left\{\eta_0^{-1}, 4(\mathrm{Lip})_0\right\}}{\bar{\rho}^2}\left[g(W_0) - g(W_{\bar{j}})\right]. \tag{200}$$

As ADAP-AIPP generates $\bar{j}$ as an iteration index, it does not terminate at the $(\bar{j}-1)$-th iteration. In view of step 3 of ADAP-AIPP, this implies that $\|R_j\| > \bar{\rho}$ for every $j = 1, \ldots, \bar{j}-1$. Using this conclusion, summing inequality (185) from 1 to $\bar{j}$, and using the lower bound on η_j in (181), we conclude that

$$(\bar{j}-1)\bar{\rho}^2 < \sum_{j=1}^{\bar{j}-1} \|R_j\|^2 \leq \sum_{j=1}^{\bar{j}} \|R_j\|^2 \overset{(185)}{\leq} C_\sigma \sum_{j=1}^{\bar{j}} \frac{g(W_{j-1}) - g(W_j)}{\eta_j} \overset{(181)}{\leq} \frac{C_\sigma}{[\min\{\eta_0, 1/(4(\mathrm{Lip})_0)\}]} \left[g(W_0) - g(W_{\bar{j}})\right]$$

which contradicts (200). Hence, ADAP-AIPP only generates iteration indices $j \geq 1$ satisfying

$$j \leq 1 + \frac{C_\sigma \max\left\{\eta_0^{-1},\, 4(\mathrm{Lip})_0\right\}}{\bar{\rho}^2} \left[g(W_0) - g(W_j)\right]. \tag{201}$$

In particular, since $g(W_j) \geq g_*$ for every generated index j, we have

$$j \leq 1 + \frac{C_\sigma \max\left\{\eta_0^{-1},\, 4(\mathrm{Lip})_0\right\}}{\bar{\rho}^2} \left[g(W_0) - g_*\right].$$

Thus ADAP-AIPP cannot generate infinitely many iteration indices, and hence it must terminate. Applying (201) with $j = \ell$, where ℓ is ADAP-AIPP's final iteration index, gives (199).

The fact that ADAP-AIPP's output $(\bar{W}, \bar{R})$ satisfies (196) is immediate from ADAP-AIPP's termination criterion in step 3 and the fact that the inclusion in (181) holds for every generated iteration index $j \geq 1$. Moreover, it follows from (185) and a simple induction argument that $g(W_j) \leq g(W_0)$ for all iteration indices $j \geq 1$, which implies that output $(\bar{W}, \bar{R})$ satisfies (197).

(b) The bound on the total number of ADAP-FISTA calls in (198) is immediate from the bound on ADAP-AIPP's final iteration index ℓ in (199), Lemma B.9(a), and the fact that (190) implies that $4(\mathrm{Lip})_j \leq 4(\mathrm{Lip})_0$. □

Remark: Each ADAP-FISTA call performs at most

$$\mathcal{O}\left(\sqrt{\bar{\Theta}}\log_2^+(\bar{\Theta}) + \log_2^+(\bar{\Theta})\right). \tag{202}$$

gradient evaluations and projections onto Ω where $\bar{\Theta}$ is as in (193). ■

Proof. The result follows immediately from the complexity bound in (184) fact that Lemma B.12 implies that $\Theta_j = \mathcal{O}(\bar{\Theta})$, and the first inequality in (191). □

References

[1] J. Aguirre, D. Cifuentes, V. Guigues, R. D. Monteiro, V. H. Nascimento, and A. Sujanani, *A user manual for cuhallar: A gpu accelerated low-rank semidefinite programming solver*, arXiv preprint arXiv:2508.15951, (2025).

[2] J. M. Aguirre, D. Cifuentes, V. Guigues, R. D. Monteiro, V. H. Nascimento, and A. Sujanani, *cuHALLaR: A GPU accelerated low-rank augmented lagrangian method for large-scale semidefinite programming*, arXiv preprint arXiv:2505.13719, (2025).

[3] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results*, SIAM Journal on Optimization, 8 (1998), pp. 746–768.

[4] R. Angell and A. McCallum, *Fast, scalable, warm-start semidefinite programming with spectral bundling and sketching*, in Proceedings of the 41st International Conference on Machine Learning, vol. 235 of Proceedings of Machine Learning Research, PMLR, 2024, pp. 1579–1615.

[5] B. Borchers, *Csdp, ac library for semidefinite programming*, Optimization methods and Software, 11 (1999), pp. 613–623.

[6] D. Brosch, J. Schwiddessen, and A. Wiegele, *The augmented mixing method: Computing high-accuracy primal-dual solutions to large-scale SDPs via column updates*, Optimization Methods and Software, (2026).

[7] S. Burer and R. D. Monteiro, *A projected gradient algorithm for solving the maxcut SDP relaxation*, Optimization methods and Software, 15 (2001), pp. 175–200.

[8] ———, *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Mathematical programming, 95 (2003), pp. 329–357.

[9] ———, *Local minima and convergence in low-rank semidefinite programming*, Mathematical programming, 103 (2005), pp. 427–444.

[10] M. O. C. Helmberg and F. Rendl, *The spectral bundle method with second-order information*, Optimization Methods and Software, 29 (2014), pp. 855–876.

[11] E. Candes and B. Recht, *Exact matrix completion via convex optimization*, Communications of the ACM, 55 (2012), pp. 111–119.

[12] Q. Deng, Q. Feng, W. Gao, D. Ge, B. Jiang, Y. Jiang, J. Liu, T. Liu, C. Xue, Y. Ye, et al., *An enhanced admm-based interior point method for linear and conic optimization*, arXiv preprint arXiv:2209.01793, (2022).

[13] ———, *New developments of admm-based interior point methods for linear programming and conic programming*, arXiv preprint arXiv:2209.01793, (2022).

[14] L. Ding and B. Grimmer, *Revisiting spectral bundle methods: Primal-dual (sub)linear convergence rates*, SIAM Journal on Optimization, 33 (2023), pp. 1305–1332.

[15] L. Ding, H. Lu, and J. Yang, *New understandings and computation on augmented lagrangian methods for low-rank semidefinite programming*, arXiv preprint arXiv:2505.15775, (2025).

[16] L. Ding, A. Yurtsever, V. Cevher, J. A. Tropp, and M. Udell, *An optimal-storage approach to semidefinite programming using approximate complementarity*, SIAM Journal on Optimization, 31 (2021), pp. 2695–2725.

[17] J. C. Duchi, O. Hinder, A. Naber, and Y. Ye, *Conic descent and its application to memory-efficient optimization over positive semidefinite matrices*, in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 8308–8317.

[18] M. A. Erdogdu, A. Ozdaglar, P. A. Parrilo, and N. D. Vanli, *Convergence rate of block-coordinate maximization burer–monteiro method for solving large sdps*, Mathematical Programming, 195 (2022), pp. 243–281.

[19] M. Fazel, *Matrix rank minimization with applications*, PhD thesis, PhD thesis, Stanford University, 2002.

[20] M. Garstka, M. Cannon, and P. Goulart, *Cosmo: A conic operator splitting method for convex conic problems*, Journal of Optimization Theory and Applications, 190 (2021), pp. 779–810.

[21] M. Grötschel, L. Lovász, and A. Schrijver, *Polynomial algorithms for perfect graphs*, in North-Holland mathematics studies, vol. 88, Elsevier, 1984, pp. 325–356.

[22] Q. Han, C. Li, Z. Lin, C. Chen, Q. Deng, D. Ge, H. Liu, and Y. Ye, *A low-rank admm splitting approach for semidefinite programming*, INFORMS Journal on Computing, (2025).

[23] Q. Han, Z. Lin, H. Liu, C. Chen, Q. Deng, D. Ge, and Y. Ye, *Accelerating low-rank factorization-based semidefinite programming algorithms on gpu*, arXiv preprint arXiv:2407.15049, (2024).

[24] C. Helmberg and K. Kiwiel, *A spectral bundle method with bounds*, Math. Program., 93 (2002), pp. 173–194.

[25] C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM Journal on Optimization, 10 (2000), pp. 673–696.

[26] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*, Grundlehren Text Editions, Springer, Berlin, Heidelberg, 1 ed., 2001.

[27] S. Homer and M. Peinado, *Design and performance of parallel and distributed approximation algorithms for maxcut*, Journal of Parallel and Distributed Computing, 46 (1997), pp. 48–61.

[28] D. Hou, T. Tang, and K.-C. Toh, *A low-rank augmented Lagrangian method for polyhedral-SDP and moment-SOS relaxations of polynomial optimization*, Mathematical Programming, (2026).

[29] ———, *RiNNAL+: A riemannian ALM solver for SDP-RLT relaxations of mixed-binary quadratic programs*, Mathematical Programming Computation, 18 (2026), pp. 679–735.

[30] W. Huang, K. A. Gallivan, and X. Zhang, *Solving PhaseLift by low-rank Riemannian optimization methods for complex semidefinite constraints*, SIAM Journal on Scientific Computing, 39 (2017), pp. B840–B859.

[31] Y. Huang and D. F. Gleich, *Suboptimality bounds for trace-bounded SDPs enable a faster and scalable low-rank SDP solver SDPLR+*, 2024. Version 3, revised 1 June 2026.

[32] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre, *Low-rank optimization on the cone of positive semidefinite matrices*, SIAM Journal on Optimization, 20 (2010), pp. 2327–2351.

[33] S. Kang, X. Jiang, and H. Yang, *Local linear convergence of the alternating direction method of multipliers for semidefinite programming under strict complementarity*, arXiv preprint arXiv:2503.20142, (2025).

[34] B. Kulis, A. C. Surendran, and J. C. Platt, *Fast low-rank semidefinite programming for embedding and clustering*, in Artificial Intelligence and Statistics, PMLR, 2007, pp. 235–242.

[35] S. Laue, *A hybrid algorithm for convex semidefinite optimization*, in Proceedings of the 29th International Conference on Machine Learning, 2012, pp. 1083–1090.

[36] B. Li and G. B. Giannakis, *Conic descent redux for memory-efficient optimization*, in 2023 57th Asilomar Conference on Signals, Systems, and Computers, IEEE, 2023, pp. 594–601.

[37] H. Li, H. Liu, D. Ge, and Y. Ye, *A curvature-aware rank-adaptive distributed augmented-lagrangian solver for large-scale sdps*, 2026.

[38] Q. Lin and L. Xiao, *An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization*, Comput. Optim. Appl., 60 (2015), pp. 633–674.

[39] L. Lovász, *On the Shannon capacity of a graph*, IEEE Transactions on Information theory, 25 (1979), pp. 1–7.

[40] R. Madani, A. Kalbat, and J. Lavaei, *Admm for sparse semidefinite programming with applications to optimal power flow problem*, in 2015 54th IEEE Conference on Decision and Control (CDC), IEEE, 2015, pp. 5932–5939.

[41] A. Majumdar, G. Hall, and A. A. Ahmadi, *Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics*, Annual Review of Control, Robotics, and Autonomous Systems, 3 (2020), pp. 331–360.

[42] S. Mei, T. Misiakiewicz, A. Montanari, and R. I. Oliveira, *Solving SDPs for synchronization and MaxCut problems via the G rothendieck inequality*, in Conference on learning theory, PMLR, 2017, pp. 1476–1515.

[43] H. D. Mittelmann, *An independent benchmarking of sdp and socp solvers*, Mathematical Programming, 95 (2003), pp. 407–430.

[44] R. D. Monteiro, A. Sujanani, and D. Cifuentes, *A low-rank augmented lagrangian method for large-scale semidefinite programming based on a hybrid convex-nonconvex approach*, Mathematical Programming, (2026). Published online 29 April 2026.

[45] R. D. C. Monteiro, *Primal–dual path-following algorithms for semidefinite programming*, SIAM Journal on Optimization, 7 (1997), pp. 663–678.

[46] F. Oustry, *A second-order bundle method to minimize the maximum eigenvalue function*, Math. Program., 89 (2000), pp. 1–33.

[47] B. O'donoghue, E. Chu, N. Parikh, and S. Boyd, *Conic optimization via operator splitting and homogeneous self-dual embedding*, Journal of Optimization Theory and Applications, 169 (2016), pp. 1042–1068.

[48] ———, *Conic optimization via operator splitting and homogeneous self-dual embedding*, Journal of Optimization Theory and Applications, 169 (2016), pp. 1042–1068.

[49] R. T. Rockafellar, *Convex Analysis*, vol. 28 of Princeton Mathematical Series, Princeton University Press, Princeton, NJ, 1970.

[50] D. M. Rosen, *Scalable low-rank semidefinite programming for certifiably correct machine perception*, in Algorithmic Foundations of Robotics XIV: Proceedings of the Fourteenth Workshop on the Algorithmic Foundations of Robotics 14, Springer, 2021, pp. 551–566.

[51] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, *SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group*, The International Journal of Robotics Research, 38 (2019), pp. 95–125.

[52] N. Shinde, V. Narayanan, and J. Saunderson, *Memory-efficient structured convex optimization via extreme point sampling*, SIAM Journal on Mathematics of Data Science, 3 (2021), pp. 787–814.

[53] A. Sujanani and R. D. Monteiro, *An adaptive superfast inexact proximal augmented lagrangian method for smooth nonconvex composite optimization problems*, Journal of Scientific Computing, 97 (2023), p. 34.

[54] ———, *Efficient parameter-free restarted accelerated gradient methods for convex and strongly convex optimization*, Journal of Optimization Theory and Applications, 206 (2025), p. 52.

[55] T. Tang and K.-C. Toh, *A feasible method for general convex low-rank sdp problems*, SIAM Journal on Optimization, 34 (2024), pp. 2169–2200.

[56] ———, *A feasible method for solving an sdp relaxation of the quadratic knapsack problem*, Mathematics of Operations Research, 49 (2024), pp. 19–39.

[57] ———, *Solving graph equipartition sdps on an algebraic variety*, Mathematical programming, 204 (2024), pp. 299–347.

[58] ———, *A preconditioned augmented Lagrangian method for solving semidefinite programming problems*, 2026.

[59] M. J. Todd, K. C. Toh, and R. H. Tütüncü, *On the nesterov–todd direction in semidefinite programming*, SIAM Journal on Optimization, 8 (1998), pp. 769–796.

[60] J. Wang and L. Hu, *Solving Low-Rank Semidefinite Programs via Manifold Optimization*, Available on arXiv:2303.01722, (2023).

[61] J. Wang, L. Hu, and B. Xia, *A dual riemannian admm algorithm for low-rank sdps with unit diagonal*, arXiv preprint arXiv:2512.04406, (2025).

[62] Y. Wang, K. Deng, H. Liu, and Z. Wen, *A decomposition augmented lagrangian method for low-rank semidefinite programming*, SIAM Journal on Optimization, 33 (2023), pp. 1361–1390.

[63] Z. Wen, D. Goldfarb, and W. Yin, *Alternating direction augmented lagrangian methods for semidefinite programming*, Mathematical Programming Computation, 2 (2010), pp. 203–230.

[64] M. Yamashita, K. Fujisawa, M. Fukuda, K. Kobayashi, K. Nakata, and M. Nakata, *Latest Developments in the SDPA Family for Solving Large-Scale SDPs*, Springer US, New York, NY, 2012, pp. 687–713.

[65] L. Yang, D. Sun, and K.-C. Toh, *SDPNAL+: A majorized semismooth newton-cg augmented lagrangian method for semidefinite programming with nonnegative constraints*, Mathematical Programming Computation, 7 (2015), pp. 331–366.

[66] A. Yurtsever, O. Fercoq, and V. Cevher, *A conditional-gradient-based augmented lagrangian framework*, in International Conference on Machine Learning, PMLR, 2019, pp. 7272–7281.

[67] A. Yurtsever, Y.-P. Hsieh, and V. Cevher, *Scalable convex methods for phase retrieval*, in 2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), IEEE, 2015, pp. 381–384.

[68] A. Yurtsever, J. A. Tropp, O. Fercoq, M. Udell, and V. Cevher, *Scalable semidefinite programming*, SIAM Journal on Mathematics of Data Science, 3 (2021), pp. 171–200.

[69] X.-Y. Zhao, D. Sun, and K.-C. Toh, *A newton-cg augmented lagrangian method for semidefinite programming*, SIAM Journal on Optimization, 20 (2010), pp. 1737–1765.

[70] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, *Fast admm for semidefinite programs with chordal sparsity*, in 2017 American Control Conference (ACC), IEEE, 2017, pp. 3335–3340.