End-To-End Resource Analysis for Quantum Interior-Point Methods and Portfolio Optimization

Alexander M. Dalzell,[1,2,*] B. David Clader,[3,†] Grant Salton,[1,2,4,‡] Mario Berta,[1,2,5,6] Cedric Yen-Yu Lin,[7] David A. Bader,[3,8] Nikitas Stamatopoulos,[3] Martin J. A. Schuetz,[1,4] Fernando G. S. L. Brandão,[1,2] Helmut G. Katzgraber,[1,4,9] and William J. Zeng[3]

[1] *AWS Center for Quantum Computing, Pasadena, California 91106, USA*
[2] *California Institute of Technology, Pasadena, California 91125, USA*
[3] *Goldman Sachs, New York, New York 10282, USA*
[4] *Amazon Quantum Solutions Lab, Seattle, Washington 98121, USA*
[5] *Department of Computing, Imperial College London, London SW7 2AZ, United Kingdom*
[6] *Institute for Quantum Information, RWTH Aachen University, 52056 Aachen, Germany*
[7] *AWS Quantum Technologies, Seattle, WA 98170, USA 98170*
[8] *New Jersey Institute of Technology, Newark, New Jersey 07102, USA*
[9] *University of Washington, Seattle, Washington 98195, USA*

(Received 17 January 2023; accepted 25 August 2023; published 13 November 2023)

We study quantum interior-point methods (QIPMs) for second-order cone programming (SOCP), guided by the example use case of portfolio optimization (PO). We provide a complete quantum circuit-level description of the algorithm from problem input to problem output, making several improvements to the implementation of the QIPM. We report the number of logical qubits and the quantity and/or depth of non-Clifford T gates needed to run the algorithm, including constant factors. The resource counts we find depend on instance-specific parameters, such as the condition number of certain linear systems within the problem. To determine the size of these parameters, we perform numerical simulations of small PO instances, which lead to concrete resource estimates for the PO use case. Our numerical results do not probe large enough instance sizes to make conclusive statements about the asymptotic scaling of the algorithm. However, already at small instance sizes, our analysis suggests that, due primarily to large constant prefactors, poorly conditioned linear systems, and a fundamental reliance on costly quantum state tomography, fundamental improvements to the QIPM are required for it to lead to practical quantum advantage.

DOI: 10.1103/PRXQuantum.4.040325

I. OVERVIEW

A. Introduction

The practical utility of finding optimal solutions to well-posed optimization problems has been known since the days of antiquity, with Euclid considering the minimal distance between two points using a line. In the modern era, optimization algorithms for business and financial use cases continue to be ubiquitous. Partly as a result of this utility, algorithmic techniques for optimization problems have been well studied since even before the invention of the computer, including a famous dispute between Legendre and Gauss on who was responsible for the invention of least-squares fitting [1]. With the advent of the quantum era, there has been great interest in developing quantum algorithms that solve optimization problems with provable speed-ups over classical algorithms. Some of the earliest proposals rely on quantum annealing [2] or more recent work in variational algorithms [3,4] to solve combinatorial optimization problems. Quantum algorithms have also been developed that allow for more efficient convex optimization, including algorithms for semidefinite, second-order cone, and linear programs [5–14], as well as algorithms for solving systems of linear equations [15–19],

*dalzel@amazon.com
†dave.clader@bqpadvisors.com (Current affiliation: BQP Advisors, LLC)
‡saltg@amazon.com

which can be used for quantum data fitting [20]. Using these techniques, specific financial use cases such as solving the portfolio-optimization problem have been studied [21–24].

Unfortunately, it can be difficult to evaluate whether these quantum algorithms will be *practically* useful. In some cases, the algorithms are heuristic and their performance can only be measured empirically once it is possible to run them on actual quantum hardware. In other cases, the difficulty in evaluating practicality stems from the inherent complexity of combining many distinct ingredients, each with their own caveats and bottlenecks. To make an apples-to-apples comparison and quantify advantages of a quantum algorithm, a truly end-to-end resource analysis that accounts for all costs from problem input to problem output must be performed.

In this work, we perform such an end-to-end analysis for a *quantum interior-point method* (QIPM) for solving second-order cone programs (SOCPs), which was originally proposed in Ref. [13], based on earlier QIPMs for semidefinite and linear programs [10]. In particular, we focus on a concrete use case with very broad application but of primary interest in the financial services sector: *portfolio optimization* (PO). In general, PO is the task of determining the optimal resource allocation to a collection of possible classes, so as to optimize a given objective. In finance, one seeks to determine the optimal allocation of funds across a set of possible assets that maximizes returns and minimizes risk, subject to constraints. Importantly, many variants of the PO problem can be cast as an SOCP and subsequently solved with a classical interior point method (CIPM) or QIPM. Indeed, CIPMs are efficient not only in theory but also in practice; they are the method of choice within fast numerical solvers for SOCPs and other conic programs (see, e.g., Ref. [25]), which encompass a large variety of optimization problems that appear in industry. Notably, QIPMs structurally mirror CIPMs and seek improvements by replacing certain subroutines with quantum primitives. Thus, compared to other proposed quantum algorithms for conic programs not based on widely used classical techniques (e.g., solvers that leverage the multiplicative weights update method [5–8]), QIPMs are uniquely positioned to provide not only a theoretical asymptotic advantage but also a practical quantum solution for this common class of problem.

However, the QIPM is a complex algorithm that delicately combines some purely classical steps with multiple distinct quantum subroutines. The runtime of the QIPM is stated in terms of several parameters that can only be evaluated once a particular use case has been specified; depending on how these parameters scale, an asymptotic speed-up may or may not be achievable. Additionally, any speed-up is contingent on access to a large quantum random access memory (QRAM), an ingredient that in prior asymptotic-focused analyses has typically been assumed to exist without much further justification or cost analysis.

Our resource analysis is detailed and takes care to study all aspects of the end-to-end pipeline, including the QRAM component. We report our results in terms of relevant problem parameters and then we perform numerical experiments to determine the size and scaling of these parameters for actual randomly chosen instances of the PO problem, based on historical stock data. This approach allows us to estimate the exact resource cost of the QIPM for an example PO problem, including a detailed breakdown of costs by various subroutines. This estimate incorporates several optimizations to the underlying subroutines and technical improvements to how they are integrated into the QIPM. Consequently, our analysis allows us to evaluate the prospect that the algorithm could exhibit a practical quantum advantage and it clearly reveals the computational bottlenecks within the algorithm that are most in need of further improvement.

While we focus on the QIPM and in particular on its application to the PO problem, our work has more general takeaways for quantum algorithms and for quantum computing applications. First, our results emphasize the importance of end-to-end analysis when evaluating a proposed application. Furthermore, our modular treatment of the underlying algorithmic primitives produces quantitative and qualitative takeaways that would be relevant for end-to-end treatments of a large number of other algorithms that also rely on these subroutines, especially those in the area of machine learning, where data access via QRAM and quantum linear-algebra techniques are often required [26].

B. Results

Our resource analysis focuses on three central quantities that determine the overall cost of algorithms implemented on fault-tolerant quantum computers: the number of *logical* qubits, the total number of T gates (the "T-count"), and the number of parallel layers of T gates (the "T-depth") needed to construct quantum circuits for solving the problem. The T-depth acts as a proxy for the overall runtime of the algorithm, whereas the T-count and the number of logical qubits are important for determining how many physical qubits would be required for a full fault-tolerant implementation. We justify the focus on T gates by pointing out that, in many prominent approaches to fault-tolerant quantum computation (such as lattice surgery [27–30]), quantum circuits are decomposed into Clifford gates and T gates and the cost of implementing the circuit is dominated by the number and depth of the T gates. The fault-tolerant Clifford gates can be performed transversally or even in software, whereas the T gates require the expensive process of magic state distillation [31,32]. We stop short of a full analysis of the algorithm at the physical level, as

TABLE I. The asymptotic leading-order contributions to the total quantum resources for an end-to-end PO (including constant factors), in terms of the number of assets in the portfolio (n), the desired precision to which the portfolio should be optimized (ϵ), the maximum Frobenius condition number of matrices encountered by the QIPM (κ_F), and the minimum tomographic precision necessary for the algorithm to succeed (ξ). The T-depth and T-count expressions represent the cumulative cost of $\mathcal{O}(\xi^{-2}n^{1.5}\log(n)\log(\epsilon^{-1}))$ individual quantum circuits performed serially, a quantity that we estimate evaluates to 6×10^{12} circuits at $n=100$ (for a detailed accounting, see Table X). The right-hand column uses a numerical simulation of the quantum algorithm (see Sec. VI) to compute the instance-specific parameters in the resource expression and estimate the resource cost at $n=100$ and $\epsilon=10^{-7}$.

Resource	QIPM complexity	Estimated at $n=100$
Number of logical qubits	$800n^2$	8×10^6
T-depth	$(1\times10^{10})\kappa_F n^{1.5}\xi^{-2}\log_2(n)\log_2(\epsilon^{-1})\log_2(\kappa_F n^{14/27}\xi^{-1})$	2×10^{24}
T-count	$(5\times10^{11})\kappa_F n^{3.5}\xi^{-2}\log_2(n)\log_2(\epsilon^{-1})\log_2(\kappa_F\xi^{-1})$	7×10^{29}

we believe the logical analysis already suffices to evaluate the overall outlook for the algorithm and identify its main bottlenecks.

At the core of any interior-point method (IPM) is the solving of a linear system of equations. The QIPM performs this step using a quantum linear-system solver (QLSS) together with pure-state quantum tomography. The cost of QLSS depends on a parameter κ_F, the Frobenius condition number $\|G\|_F\|G^{-1}\|$ of the matrix G that must be inverted (where $\|\cdot\|_F$ denotes the Frobenius norm and $\|\cdot\|$ denotes the spectral norm), while the cost of tomography depends on a parameter ξ, a precision parameter. We evaluate these parameters empirically by simulating the QIPM on small instances of the PO problem.

In Table I, we report a summary of our overall resource calculation, in which we show the asymptotically leading term (along with its constant prefactor) in terms of parameters κ_F and ξ, as well as n, the number of assets in the PO instance, and ϵ, the desired precision to which the portfolio should be optimized. We find (numerically) that κ_F grows with n and that ξ shrinks with n; we estimate that, at $n=100$ and $\epsilon=10^{-7}$, our implementation of the QIPM would require 8×10^6 qubits and 7×10^{29} total T gates spread out over 2×10^{24} layers. Needless to say, these resource counts are decidedly out of reach both in the near and far term for quantum hardware, even for a problem of modest size by classical standards. Even if quantum computers one day match the gigahertz-level clock speeds of modern classical computers, 10^{24} layers of T gates would take millions of years to execute. By contrast, the PO problem can be easily solved in a matter of seconds on a laptop for $n=100$ stocks.

We caution that the numbers we report should not be interpreted as the final word on the cost of the QIPM for PO. We are certain that further examination of the algorithm could uncover many improvements and optimizations that would reduce the costs compared to our calculations. On the other hand, we note that our results do already incorporate several innovations we have made to reduce the resource cost, including a basic attempt at preconditioning the linear system. Moreover, the pessimistic outlook our results convey is robust in the sense that the calculation would need to decrease by many orders of magnitude for the algorithm to be practical, suggesting that fundamental changes are necessary to multiple aspects of the algorithm, rather than merely superficial optimizations.

Besides the main resource calculation, we make several additional contributions and observations:

(1) We provide explicit quantum circuits for the important subroutines of the QIPM, namely, the state-of-the-art QLSS based on the discrete adiabatic theorem [18] and pure-state tomography, which complement the explicit circuits for block-encoding (using QRAM) that a subset of the authors have already reported separately in Ref. [33]. These circuits, and their precise resource calculations, could be useful elsewhere, as these subroutines are ubiquitous in quantum algorithms. For additional details, see Secs. IV F and V.

(2) We break down the resource calculation into its constituents to illustrate which parts of the algorithm are most costly. We find that many independent factors create significant challenges toward realizing quantum advantage with QIPMs and our work underscores those aspects of the algorithm that must be improved for it to be useful. We also note that the conditions under which QIPMs would be most successful (namely, when κ_F is small) also allow for classical IPMs based on iterative classical linear-system solvers to be competitive. For additional details, see Sec. VII.

(3) We numerically simulate several versions of the full QIPM solving the PO problem on portfolios as large as $n=120$ stocks and we report the empirical size and scaling of the relevant parameters, κ_F and ξ. There is considerable variability in the trends that we observe, depending on which version of the QIPM is chosen, and when the QIPM is terminated, which makes it difficult to draw robust conclusions. However, we find that both κ_F and ξ^{-1} appear to grow with n. Note that previous numerical experiments on a similar formulation of the PO problem [22] have suggested that κ_F does not grow with the

problem size but those authors scaled the number of "time epochs" while keeping n constant. Additionally, we observe that the "infeasible" version of the QIPM originally proposed in Ref. [13] empirically performs similarly to more sophisticated "feasible" versions [14], despite not enjoying the same theoretical guarantees of fast convergence. Finally, contrary to theoretical expectation, we observe that κ_F and ξ^{-1} do *not* diverge as $\epsilon \to 0$ in our examples. For additional details, see Sec. VI.

(4) We make various technical improvements to the underlying ingredients of QIPMs. A subset of the present authors have previously reported [33] a quadratic improvement in the minimum depth required for the problem of preparing an arbitrary L-dimensional quantum state or block-encoding an arbitrary $L \times L$ matrix, along with explicit quantum circuits and exact resource expressions. In this paper, we additionally contribute the following:

   (a) *Tomographic precision*. Performing tomography on the output of a QLSS necessarily causes the classical estimate of the solution to the linear system to be inexact. We illustrate how the allowable amount of tomography precision can be determined adaptively rather than relying on theoretical bounds. Nonetheless, we also improve the constant prefactor in the tomographic bounds. The total number of state-preparation queries needed to learn an unknown L-dimensional pure state to ξ error using the tomography method of Refs. [10,13] is, to leading order, at most $115L\ln(L)/\xi^2$ [34].

   (b) *Norm of the linear system*. Since QLSSs output a normalized quantum state, tomography does not directly yield the norm of the solution to the linear system. The norm can be learned through more complicated protocols but we observe that in the context of QIPMs, a sufficient estimate for the norm can be learned classically.

   (c) *Preconditioning*. We propose a simple preconditioning method that is compatible with the QIPM, while reducing the parameter κ_F. Our numerical simulations suggest that the reduction is more than an order of magnitude for the PO problem.

   (d) *Feasible QIPM*. We implement a "feasible" version of the QIPM proposed in Ref. [14], which relies on finding a basis for the null space of the SOCP matrix. We have identified an explicit basis for the PO problem, thereby avoiding the need for a costly QR decomposition. However, we observe that finding the basis via QR decomposition leads to more stable numerical results.

The outline for the remainder of the paper is as follows. In Sec. II, we describe and define the PO problem in terms of Markowitz portfolio theory. In Sec. III, we describe second-order cone programming (SOCP) problems, illustrate how PO can be represented as an instance of SOCP, and discuss how IPMs can be used for solving SOCPs. In Sec. IV, we review the *quantum* ingredients needed to turn an IPM into a QIPM. In particular, we review QLSSs, block-encoding for data loading, and quantum state tomography for data readout. We also present slightly better bounds on the required tomography procedure than were previously known. In Sec. V, we describe the full implementation of using QIPM and quantum algorithms for SOCP for the PO problem, including a detailed resource estimate for the end-to-end problem. In Sec. VI, we show numerical results from simulations of the full problem and in Sec. VII, we reflect on the calculation we have performed, identifying the main bottlenecks and drawing conclusions about the outlook for quantum advantage with QIPM.

The QIPM has many moving parts requiring several mathematical symbols. While all symbols are defined as they are introduced in the text, we also provide a full list of symbols for the reader's reference in Appendix A. Throughout the paper, we denote all vectors in bold lowercase letters to contrast with scalar quantities (unbolded lowercase) and matrices (unbolded uppercase). The only exception to this rule will be the symbols N, K, and L, which are positive integers (despite being uppercase) and which denote the number of rows or columns in certain matrices related to an SOCP instance.

II. PORTFOLIO OPTIMIZATION (PO)

A. Background

Portfolio optimization is the process widely used by financial analysts to assign allocations of capital across a set of assets within a portfolio, given optimization criteria such as maximizing the expected return and minimizing the financial risk. The creation of the mathematical framework for modern portfolio theory (MPT) is credited to Harry Markowitz [35,36], for which he received the 1990 Alfred Nobel Memorial Prize in Economic Sciences [37]. Markowitz describes the process of selecting a portfolio in two stages, where the first stage starts with "observation and experience" and ends with "beliefs about the future performances of available securities." The second stage starts with "the relevant beliefs about future performances" and ends with "the choice of portfolio." The theory is also known as *mean-variance analysis*. For further history, Markowitz's 1999 essay [38] gives the early history of portfolio theory, from 1600 to 1960.

Typically, PO strategies include diversification, which is the practice of investing in a wide array of asset types and classes as a risk-mitigation strategy. Some popular

asset classes are stocks, bonds, real estate, commodities, and cash. After building a portfolio, we expect a return (or profit) after a specific period of time. *Risk* is defined as the fluctuations of the asset value. MPT describes how high-variance assets can be combined with other uncorrelated assets through diversification to create portfolios with low variance on their return. Naturally, among equal-risk portfolios, investors prefer those with higher expected return, and among equal-return portfolios, they prefer those with lower risk.

B. Mathematical formulation

Within a portfolio, w_i represents the amount of an asset i we are holding over some period of time. Often, this amount is given as the price of the asset in dollars at the start of the period. When the price is positive ($w_i > 0$), we call this a *long* position; and when the price is negative ($w_i < 0$), we call this a *short* position with an obligation to buy this asset at the end of the period [39]. The optimization variable in our PO problem is the vector of n assets $\mathbf{w} \in \mathbb{R}^n$ in our portfolio.

The price of each asset i varies over time. We define u_i to be the relative change (positive or negative) during the period of interest. Then, we define the return of the portfolio for that period as $\bar{r} = \mathbf{u}^\intercal \mathbf{w}$ dollars. The relative changes $\mathbf{u} \in \mathbb{R}^n$ follow a stochastic process and we can model this with a random vector with mean $\hat{\mathbf{u}}$ and covariance Σ. The return $\bar{r}$ is then a random variable with mean $\hat{\mathbf{u}}^\intercal \mathbf{w}$ and covariance $\mathbf{w}^\intercal \Sigma \mathbf{w}$.

To capture realistic problem formulations, we add one or more mathematical constraints to the optimization problem corresponding to the problem-specific considerations. For example, two common constraints in PO problems are that we want no short positions ($w_i \geq 0$ for all i, denoted by $\mathbf{w} \geq 0$) and that the total investment budget is limited ($\mathbf{1}^\intercal \mathbf{w} = 1$, where $\mathbf{1}$ denotes the vector of ones). This forms the classical PO problem from Markowitz's mean-variance theory:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \mathbf{w}^\intercal \Sigma \mathbf{w} \\ \text{such that} \quad & \hat{\mathbf{u}}^\intercal \mathbf{w} \geq \bar{r}_{\min}, \\ & \mathbf{1}^\intercal \mathbf{w} = 1, \\ & \mathbf{w} \geq 0. \end{aligned} \tag{1}$$

This formulation is a quadratic optimization problem where we minimize the risk, while achieving a target return of at least $\bar{r}_{\min}$ with a fixed budget and no short positions. In practice, the PO problem is often reformulated in other ways, e.g., to maximize return subject to a fixed amount of risk or to optimize an objective function that weighs risk against return. In our application, we follow the latter approach, formulated as follows, where q is a tunable *risk-aversion coefficient*:

$$\begin{aligned} \min_{\mathbf{w}} \quad & -\hat{\mathbf{u}}^\intercal \mathbf{w} + q\sqrt{\mathbf{w}^\intercal \Sigma \mathbf{w}} \\ \text{such that} \quad & \mathbf{1}^\intercal \mathbf{w} = 1, \\ & \mathbf{w} \geq 0. \end{aligned} \tag{2}$$

This optimization problem is no longer a QO problem but it can be mapped to a conic problem, as described later, in Sec. III B. Depending on the problem, additional constraints can be added [40]. To illustrate the flexibility of this analysis, we include a maximum-transaction constraint and use the following problem formulation in our analysis in the rest of the paper:

$$\begin{aligned} \min_{\mathbf{w}} \quad & -\hat{\mathbf{u}}^\intercal \mathbf{w} + q\sqrt{\mathbf{w}^\intercal \Sigma \mathbf{w}} \\ \text{such that} \quad & \mathbf{1}^\intercal \mathbf{w} = 1, \\ & |\mathbf{w} - \bar{\mathbf{w}}| \leq \boldsymbol{\zeta}, \\ & \mathbf{w} \geq 0, \end{aligned} \tag{3}$$

where $\bar{\mathbf{w}}$ denotes the current portfolio, so that $|\mathbf{w} - \bar{\mathbf{w}}|$ is the vector of transaction quantities for each asset, which are constrained to be smaller than the values contained in the vector $\boldsymbol{\zeta}$. Note that the authors of Ref. [22] chose a formulation more akin to Eq. (1) for their numerical study of the QIPM for PO. For more information on the theory of convex optimization problems and algorithms for solving them, we direct the reader to Refs. [41,42]. For more information about optimization methods in finance, we refer to Refs. [43–45].

III. SECOND-ORDER CONE PROGRAMMING (SOCP) AND INTERIOR-POINT METHODS (IPMs)

A. Definitions

Second-order cone programming (SOCP) is a type of convex optimization that allows for a richer set of constraints than linear programming (LP), without many of the complications of semidefinite programming (SDP). Indeed, SOCP is a subset of SDP but SOCP admits IPMs that are essentially just as efficient as IPMs for LP [46]. Many real-world problems can be cast as SOCP, including the PO problem in which we are interested.

For any k-dimensional vector $\mathbf{v}$, we may write $\mathbf{v} = (v_0; \tilde{\mathbf{v}})$, where v_0 is the first entry of $\mathbf{v}$ and $\tilde{\mathbf{v}}$ contains the remaining $k-1$ entries.

Definition 1.—A *k-dimensional second-order cone* (for $k \geq 2$) is the convex set

$$\mathcal{Q}^k = \left\{ (x_0; \tilde{\mathbf{x}}) \in \mathbb{R}^k \mid x_0 \geq \|\tilde{\mathbf{x}}\| \right\}, \tag{4}$$

where $\|\cdot\|$ denotes the vector two-norm (standard Euclidean norm). For $k = 1$, $\mathcal{Q}^1 = \{x_0 \in \mathbb{R} \mid x_0 \geq 0\}$.

Definition 2.—In general, a *second-order cone program* is formulated as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^\intercal \mathbf{x} \\ \text{such that} \quad & A\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \in \mathcal{Q}, \end{aligned} \tag{5}$$

where $\mathcal{Q} = \mathcal{Q}^{N_1} \times \cdots \times \mathcal{Q}^{N_r}$ is a Cartesian product of r second-order cones of combined dimension $N = N_1 + \ldots + N_r$ and A is a full-rank $K \times N$ matrix encoding K linear equality constraints, with $K \leq N$.

Note that the special case of linear programming is immediately recovered if $N_i = 1$ for all i. We say that a point $\mathbf{x}$ is *primal feasible* whenever $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \in \mathcal{Q}$. It is *strictly primal feasible* if, additionally, it lies in the interior of $\mathcal{Q}$.

The dual to the program in Eq. (5) is a maximization problem over a variable $\mathbf{y} \in \mathbb{R}^K$, given as follows:

$$\begin{aligned} \max_{\mathbf{y}} \quad & \mathbf{b}^\intercal \mathbf{y} \\ \text{such that} \quad & A^\intercal \mathbf{y} + \mathbf{s} = \mathbf{c}, \\ & \mathbf{s} \in \mathcal{Q}. \end{aligned} \tag{6}$$

We say that a pair $(\mathbf{s}; \mathbf{y})$ is *dual feasible* whenever $A^\intercal \mathbf{y} + \mathbf{s} = \mathbf{c}$ and $\mathbf{s} \in \mathcal{Q}$. For any point $(\mathbf{x}; \mathbf{y}; \mathbf{s})$ with $\mathbf{x}, \mathbf{s} \in \mathcal{Q}$, we define the *duality gap* as

$$\mu(\mathbf{x}, \mathbf{s}) := \frac{1}{r}\mathbf{x}^\intercal \mathbf{s} = \frac{1}{r}(\mathbf{c}^\intercal \mathbf{x} - \mathbf{b}^\intercal \mathbf{y}), \tag{7}$$

where r is the number of cones, as in Definition 2, and the second equality holds under the additional assumption that the point is primal and dual feasible. The fact that $\mathbf{x}, \mathbf{s} \in \mathcal{Q}$ implies that $\mu(\mathbf{x}, \mathbf{s}) \geq 0$. Moreover, assuming that both the primal and dual problems have a strictly feasible point, the optimal primal solution $\mathbf{x}^*$ and the optimal dual solution $(\mathbf{y}^*; \mathbf{s}^*)$ are guaranteed to exist and satisfy $\mathbf{c}^\intercal \mathbf{x}^* = \mathbf{b}^\intercal \mathbf{y}^*$, and hence $\mu = \mathbf{x}^{*\intercal}\mathbf{s}^*/r = \mathbf{x}^{*\intercal}(\mathbf{c} - A^\intercal \mathbf{y}^*)/r = (\mathbf{c}^\intercal \mathbf{x}^* - \mathbf{b}^\intercal \mathbf{y}^*)/r = 0$ [46]. Thus, the primal-dual condition of optimality can be expressed by the system

$$\begin{aligned} A\mathbf{x} &= \mathbf{b}, \\ A^\intercal \mathbf{y} + \mathbf{s} &= \mathbf{c}, \\ \mathbf{x}^\intercal \mathbf{s} &= 0, \\ \mathbf{x} \in \mathcal{Q}, \quad & \mathbf{s} \in \mathcal{Q}. \end{aligned} \tag{8}$$

B. Portfolio optimization as SOCP

The PO problem can be solved by reduction to SOCP [45] and this reduction is often made in practice. Here, we describe one way of translating the PO problem, as given in Eq. (3), into a second-order cone program.

The objective function in Eq. (3) has a nonlinear term $q\sqrt{\mathbf{w}^\intercal \Sigma \mathbf{w}}$, which we linearize by introducing a new scalar variable t and a new constraint $t \geq \sqrt{\mathbf{w}^\intercal \Sigma \mathbf{w}}$. We obtain the equivalent optimization problem:

$$\begin{aligned} \min_{\mathbf{x}=(\mathbf{w};t)} \quad & [-\hat{\mathbf{u}}; q]^\intercal (\mathbf{w}; t) \\ \text{such that} \quad & \mathbf{1}^\intercal \mathbf{w} = 1, \\ & |w_i - \bar{w}_i| \leq \zeta_i, \\ & w_i \geq 0, \\ & t^2 \geq \mathbf{w}^\intercal \Sigma \mathbf{w}. \end{aligned} \tag{9}$$

Our goal now is to write the constraints in Eq. (9) as second-order cone constraints. Given an $m \times n$ matrix M for which $\Sigma = M^\intercal M$, the constraint on t can be expressed by introducing an m-dimensional variable $\boldsymbol{\eta}$ subject to the equality constraint $\boldsymbol{\eta} = M\mathbf{w}$ and the second-order cone constraint $(t; \boldsymbol{\eta}) \in \mathcal{Q}^{m+1}$.

The matrix M can be determined from Σ via a Cholesky decomposition, although for large matrices Σ, this computation may be costly. Alternatively, if Σ and $\hat{\boldsymbol{\mu}}$ are calculated from stock-return vectors $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(m)}$ during m independent *time epochs* (e.g., returns for each of m days or each of m months), then a valid matrix $M^\intercal$ is given by $(\mathbf{u}^{(1)} - \hat{\mathbf{u}}, \ldots, \mathbf{u}^{(m)} - \hat{\mathbf{u}})$, i.e., the columns of $M^\intercal$ are given by the deviation of the returns from the mean in each epoch. This was the approach taken in Ref. [22] and is also the approach we take in our numerical experiments, presented later. The downside to this approach is that the number of time epochs must grow with the number of assets. We note that, in practice, computing the matrix Σ can be a research topic unto itself, which is beyond the scope of this paper [48].

The absolute-value constraints are handled by introducing a pair of n-dimensional variables $\boldsymbol{\phi}$ and $\boldsymbol{\rho}$, subject to equality constraints $\boldsymbol{\phi} = \boldsymbol{\zeta} - (\mathbf{w} - \bar{\mathbf{w}})$ and $\boldsymbol{\rho} = \boldsymbol{\zeta} + (\mathbf{w} - \bar{\mathbf{w}})$. The absolute-value constraints are then imposed as positivity constraints $\phi_i \geq 0$, $\rho_i \geq 0$, which we include as second-order cone constraints of dimension 1 [49].

In summary, we may write the PO problem from Eq. (3) as the following SOCP that minimizes over the variable $\mathbf{x} = (\mathbf{w}; \boldsymbol{\phi}; \boldsymbol{\rho}; t; \boldsymbol{\eta}) \in \mathbb{R}^{3n+m+1}$:

$$\min_{\mathbf{x}} \quad [-\hat{\mathbf{u}}; \mathbf{0}; \mathbf{0}; q; \mathbf{0}]^\intercal (\mathbf{w}; \boldsymbol{\phi}; \boldsymbol{\rho}; t; \boldsymbol{\eta}) =: \mathbf{c}^\intercal \mathbf{x}$$

such that

$$\begin{pmatrix} \mathbf{1}^\intercal & \mathbf{0}^\intercal & \mathbf{0}^\intercal & 0 & \mathbf{0}^\intercal \\ I & I & 0 & \mathbf{0} & 0 \\ I & 0 & -I & \mathbf{0} & 0 \\ M & 0 & 0 & \mathbf{0} & -I \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \boldsymbol{\phi} \\ \boldsymbol{\rho} \\ t \\ \boldsymbol{\eta} \end{pmatrix} = \begin{pmatrix} 1 \\ \bar{\mathbf{w}} + \boldsymbol{\zeta} \\ \bar{\mathbf{w}} - \boldsymbol{\zeta} \\ \mathbf{0} \end{pmatrix}, \tag{10}$$

$$(\mathbf{w};\boldsymbol{\phi};\boldsymbol{\rho};t;\boldsymbol{\eta}) \in \underbrace{\mathcal{Q}^1 \times \cdots \times \mathcal{Q}^1}_{n \text{ positivity constraints}} \times \underbrace{\mathcal{Q}^1 \times \cdots \times \mathcal{Q}^1}_{2n \text{ budget constraints}} \times \underbrace{\mathcal{Q}^{m+1}}_{\text{risk}},$$

where I denotes an identity block, 0 denotes a submatrix of all 0s, $\mathbf{0}$ is a vector of all 0s, $\mathbf{1}$ is a vector of all 1s, and the size of each block of A can be inferred from its location in the matrix. Thus, the total number of cones is $r = 3n + 1$ and the combined dimension is $N = 3n + m + 1$ [50]. The SOCP constraint matrix A is a $K \times N$ matrix, with $K = 2n + m + 1$. This SOCP is very similar to that considered by Kerenidis, Prakash, and Szilágyi [22]; however, rather than optimize a weighted combination of risk and return, they optimized risk subject to a fixed value for return and they did not include the budget constraints.

Note that many of the rows of the $K \times N$ matrix A are sparse and contain only one or two nonzero entries. However, the final m rows of the matrix A will be dense and will contain $n + 1$ nonzero entries due to the appearance of the matrix M containing historical stock data; in total, a constant fraction of the matrix entries will be nonzero, so sparse-matrix techniques will provide only limited benefit.

Finally, we can observe that the primal SOCP in Eq. (10) has an interior feasible point as long as $\boldsymbol{\zeta}$ has strictly positive entries. To see this, choose $\mathbf{w}$ to be any strictly positive vector that satisfies $|\mathbf{w} - \bar{\mathbf{w}}| < \boldsymbol{\zeta}$ and let $\boldsymbol{\phi} = \boldsymbol{\zeta} + (\bar{\mathbf{w}} - \mathbf{w})$, $\boldsymbol{\rho} = \boldsymbol{\zeta} - (\bar{\mathbf{w}} - \mathbf{w})$, and $\boldsymbol{\eta} = M\mathbf{w}$ and let t be equal to any number strictly greater than $\|\boldsymbol{\eta}\|$. It can be verified that the dual program likewise has a strictly feasible point; this guarantees that the optimal primal-dual pair for the SOCP exists and satisfies Eq. (8).

C. Interior-point methods for SOCP

1. Introduction

IPMs are a class of efficient algorithms for solving convex optimization problems including LPs, SOCPs, and SDPs, where (in contrast to the simplex method) intermediate points generated by the method lie in the *interior* of the convex set and they are guaranteed to approach the optimal point after a polynomial number of iterations of the method. Each iteration involves forming a linear system of equations that depends on the current intermediate point. The solution to this linear system determines the *search direction* and the next intermediate point is formed by taking a small step in that direction. We will consider path-following primal-dual IPMs, where, if the step size is sufficiently small, the intermediate points are guaranteed to approximately follow the *central path*, which ends at the optimal point for the convex optimization problem.

2. Central path

To define the central path, we first establish some notation related to the algebraic properties of the second-order cone. Following formulations in the prior literature [13,46], we let the product $\mathbf{u} \circ \mathbf{v}$ of two vectors $\mathbf{u} = (u_0;\tilde{\mathbf{u}})$, $\mathbf{v} = (v_0;\tilde{\mathbf{v}}) \in \mathcal{Q}^k$ be defined as

$$\mathbf{u} \circ \mathbf{v} = (\mathbf{u}^\intercal \mathbf{v}; u_0\tilde{\mathbf{v}} + v_0\tilde{\mathbf{u}}) \tag{11}$$

and we denote the identity element for this product by the vector $\mathbf{e} = (1;\mathbf{0}) \in \mathcal{Q}^k$. For the Cartesian product $\mathcal{Q} = \mathcal{Q}^{N_1} \times \ldots \times \mathcal{Q}^{N_r}$ of multiple second-order cones, the vector $\mathbf{e}$ is defined as the concatenation of the identity element for each cone and the circle product of two vectors is given by the concatenation of the circle product of each constituent. A consequence of this definition is the that $\mathbf{e}^\intercal\mathbf{e}$ is equal to the number of cones r.

Now, for the SOCP problem of Eq. (5), the central path $(\mathbf{x}(\nu);\mathbf{y}(\nu);\mathbf{s}(\nu))$ is the one-dimensional set of *central points*, parametrized by $\nu \in [0,\infty)$, which satisfies the conditions

$$\begin{aligned} A\mathbf{x}(\nu) &= \mathbf{b}, \\ A^\intercal\mathbf{y}(\nu) + \mathbf{s}(\nu) &= \mathbf{c}, \\ \mathbf{x}(\nu) \circ \mathbf{s}(\nu) &= \nu\mathbf{e}, \\ \mathbf{x}(\nu) \in \mathcal{Q}, \quad \mathbf{s}(\nu) &\in \mathcal{Q}. \end{aligned} \tag{12}$$

We can immediately see that the central-path point $(\mathbf{x}(\nu);\mathbf{y}(\nu);\mathbf{s}(\nu))$ has a duality gap that satisfies $\mu(\mathbf{x}(\nu), \mathbf{s}(\nu)) = \nu$ and that when $\nu = 0$, Eq. (12) recovers Eq. (8).

3. Finding an initial point on the central path via self-dual embedding

Path-following primal-dual IPMs find the optimal point by beginning at a central point with $\nu > 0$ and following the central path to a very small value of ν, which is taken to be a good approximation of the optimal point. For a given SOCP, finding an initial point on the central path is nontrivial and, in general, can be just as hard as solving the SOCP itself. One solution to this problem is the homogeneous self-dual embedding [51,52], where one forms a slightly larger self-dual SOCP with the properties that (i) the optimal point for the original SOCP can be determined from the optimal point for the self-dual SOCP and (ii) the self-dual SOCP has a trivial central point that can be used to initialize the IPM.

To do this, we introduce new scalar variables τ, θ, and $\varkappa$, which are used to give more flexibility to the constraints. Previously, we required $A\mathbf{x} = \mathbf{b}$. In the larger program, we relax this constraint to read $A\mathbf{x} = \mathbf{b}\tau - (\mathbf{b} - A\mathbf{e})\theta$, such that the original constraint is recovered when $\tau = 1$ and $\theta = 0$ but $\mathbf{x} = \mathbf{e}$ is a trivial solution when $\tau = 1$ and $\theta = 1$. Similarly, we relax the constraint $A^\intercal\mathbf{y} + \mathbf{s} =$

$\mathbf{c}$ to read $A^\intercal\mathbf{y}+\mathbf{s}=\mathbf{c}\tau-(\mathbf{c}-\mathbf{e})\theta$, which has the trivial solution $\mathbf{y}=\mathbf{0}$, $\mathbf{s}=\mathbf{e}$ when $\tau=\theta=1$. We complement these with two additional linear constraints to form the program

$$\min_{(\mathbf{x};\mathbf{y};\tau;\theta;\mathbf{s};\varkappa)} \quad (r+1)\theta$$

such that (13)

$$\begin{pmatrix} 0 & A^\intercal & -\mathbf{c} & \bar{\mathbf{c}} \\ -A & 0 & \mathbf{b} & -\bar{\mathbf{b}} \\ \mathbf{c}^\intercal & -\mathbf{b}^\intercal & 0 & -\bar{z} \\ -\bar{\mathbf{c}}^\intercal & \bar{\mathbf{b}}^\intercal & \bar{z} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \tau \\ \theta \end{pmatrix} + \begin{pmatrix} \mathbf{s} \\ \mathbf{0} \\ \varkappa \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ 0 \\ r+1 \end{pmatrix}$$

$$\mathbf{x},\mathbf{s}\in\mathcal{Q}; \qquad \tau,\varkappa\geq 0; \qquad y,\theta \text{ free},$$

where $\bar{\mathbf{b}}=\mathbf{b}-A\mathbf{e}$, $\bar{\mathbf{c}}=\mathbf{c}-\mathbf{e}$, $\bar{z}=\mathbf{c}^\intercal\mathbf{e}+1$, and $r=\mathbf{e}^\intercal\mathbf{e}$ is the number of cones in the original SOCP. While Eq. (13) is not exactly of the form given in Eq. (5), we may still think of it as a primal SOCP. Since the block matrix in Eq. (13) is skew symmetric and the objective-function coefficients are equal to the right-hand side of the equality constraints, when we compute the dual program [cf. Eq. (6)], we arrive at an equivalent program; we conclude that Eq. (13) is *self-dual* [51]. Thus, when applying path-following primal-dual IPMs to Eq. (13), we need only keep track of the primal variables, i.e., $\mathbf{x},\mathbf{y},\tau,\theta,\mathbf{s},\varkappa$. Taking into account the addition of τ and $\varkappa$, which are effectively an extra pair of primal-dual variables, we redefine the duality gap [cf. Eq. (7)] as

$$\mu(\mathbf{x},\tau,\mathbf{s},\varkappa):=\frac{1}{r+1}(\mathbf{x}^\intercal\mathbf{s}+\varkappa\tau). \tag{14}$$

Note that if the point $(\mathbf{x};\mathbf{y};\tau;\theta;\mathbf{s};\varkappa)$ is feasible, i.e., if it satisfies the four linear constraints in Eq. (13), then we have the identity

$$\begin{aligned}\mu(\mathbf{x},\tau,\mathbf{s},\varkappa)&=\frac{-\mathbf{x}^\intercal A^\intercal\mathbf{y}+\mathbf{x}^\intercal\mathbf{c}\tau-\mathbf{x}^\intercal\bar{\mathbf{c}}\theta+\varkappa\tau}{r+1}\\&=\frac{-\mathbf{b}^\intercal\mathbf{y}\tau+\bar{\mathbf{b}}^\intercal\mathbf{y}\theta+\mathbf{x}^\intercal\mathbf{c}\tau-\mathbf{x}^\intercal\bar{\mathbf{c}}\theta+\varkappa\tau}{r+1}\\&=\frac{\bar{\mathbf{b}}^\intercal\mathbf{y}\theta-\mathbf{x}^\intercal\bar{\mathbf{c}}\theta+\bar{z}\tau\theta}{r+1}\\&=\theta,\end{aligned} \tag{15}$$

where the first, second, third, and fourth rows of Eq. (13) are invoked above in lines one, two, three, and four, respectively. This equality justifies the redefinition in Eq. (14): noting that the primal objective function in Eq. (13) is $(r+1)\theta$ and (since the program is self-dual) the associated dual objective function is $-(r+1)\theta$, we see that the *gap* between primal and dual objective functions, divided by the number of conic constraints $(2r+2)$, is exactly equal to θ.

The central path for the augmented SOCP in Eq. (13) is defined by the feasibility conditions for the SOCP combined with the relaxed complementarity conditions $\mathbf{x}\circ\mathbf{s}=\nu\mathbf{e}$ and $\varkappa\tau=\nu$. Thus, we see that the point $(\mathbf{x}=\mathbf{e};\mathbf{y}=0;\tau=1;\theta=1;\mathbf{s}=\mathbf{e};\varkappa=1)$ is not only a feasible point for the SOCP in Eq. (13) but also a central point with $\nu=1$.

Finally, a crucial property [51] of the self-dual SOCP in Eq. (13) is that the optimal point for the original SOCP in Eq. (5) can be derived from the optimal point for the SOCP in Eq. (13). Specifically, let $(\mathbf{x}^*_{sd};\mathbf{y}^*_{sd};\tau^*;\theta^*;\mathbf{s}^*_{sd};\varkappa^*)$ be the optimal point for Eq. (13) (it can be shown that $\theta^*=0$). Then, if $\tau^*>0$, $(\mathbf{x}^*;\mathbf{y}^*;\mathbf{s}^*)=(\mathbf{x}^*_{sd}/\tau^*;\mathbf{y}^*_{sd}/\tau^*;\mathbf{s}^*_{sd}/\tau^*)$ is an optimal primal-dual point for Eqs. (5) and (6). If $\tau^*=0$, then at least one of the original primal SOCPs in Eq. (5) and the original dual SOCP in Eq. (6) must be infeasible [51,52]. As previously demonstrated, the specific SOCP for PO in Eq. (10) is primal and dual feasible, so $\tau^*\neq 0$ for that example.

What if we only have a point that is *approximately* optimal for the self-dual SOCP? We can still deduce an approximately optimal point for the original SOCP. Suppose that we have a feasible point for which $\mu(\mathbf{x},\tau,\mathbf{s},\varkappa)=\epsilon$. The point $(\mathbf{x}/\tau;\mathbf{y}/\tau;\mathbf{s}/\tau)$ is $\mathcal{O}(\epsilon)$ close to feasible for the original SOCP in the sense that the equality constraints are satisfied up to $\mathcal{O}(\epsilon)$ error:

$$\left\|A\frac{\mathbf{x}}{\tau}-\mathbf{b}\right\|=\frac{\epsilon}{\tau}\|\mathbf{b}-A\mathbf{e}\|, \tag{16}$$

$$\left\|A^\intercal\frac{\mathbf{y}}{\tau}+\frac{\mathbf{s}}{\tau}-\mathbf{c}\right\|=\frac{\epsilon}{\tau}\|\mathbf{c}-\mathbf{e}\|. \tag{17}$$

Moreover, since $\varkappa>0$ and $\theta=\epsilon$, we can assert using the third row of Eq. (13) that the difference in objective function achieved by the primal and dual solutions is also $\mathcal{O}(\epsilon)$, i.e.,

$$\mathbf{c}^\intercal\frac{\mathbf{x}}{\tau}-\mathbf{b}^\intercal\frac{\mathbf{y}}{\tau}\leq\frac{|\mathbf{c}^\intercal\mathbf{e}+1|}{\tau}\epsilon. \tag{18}$$

In summary, by using the self-dual SOCP of Eq. (13), we obtain a trivial point from which to start the IPM and given an (approximately) optimal point we obtain either an (approximately) optimal point to the original SOCP or a certificate that the original SOCP was not feasible to begin with.

4. Iterating the IPM

Each iteration of the IPM takes as input an intermediate point $(\mathbf{x};\mathbf{y};\tau;\theta;\mathbf{s};\varkappa)$ that is feasible (or in some formulations, nearly feasible), has duality gap $(\mathbf{x}^\intercal\mathbf{s}+\varkappa\tau)/(r+1)$ equal to μ, and is close to the central path with parameter $\nu=\mu$. The output of the iteration is a new intermediate point $(\mathbf{x}+\Delta\mathbf{x};\mathbf{y}+\Delta\mathbf{y};\tau+\Delta\tau;\theta+\Delta\theta;\mathbf{s}+\Delta\mathbf{s},\varkappa+$

$\Delta\varkappa$) that is also feasible and close to the central path, with a reduced value of the duality gap. Thus, iterating many times leads to a solution with a duality gap arbitrarily close to zero.

One additional input is the step size, governed by a parameter $\sigma < 1$. The IPM iteration aims to bring the next intermediate point onto the central path with parameter $\nu = \sigma\mu$. This is accomplished by taking one step using Newton's method, where the vector $(\Delta\mathbf{x}; \Delta\mathbf{y}; \Delta\tau; \Delta\theta; \Delta\mathbf{s}; \Delta\varkappa)$ is uniquely determined by solving a linear system of equations called the Newton system. The first part of the Newton system is the conditions that must be met for the new point to be feasible, given in the following system of $N + K + 2$ linear equations:

$$\begin{pmatrix} 0 & A^\intercal & -\mathbf{c} & \bar{\mathbf{c}} \\ -A & 0 & \mathbf{b} & -\bar{\mathbf{b}} \\ \mathbf{c}^\intercal & -\mathbf{b}^\intercal & 0 & -\bar{z} \\ -\bar{\mathbf{c}}^\intercal & \bar{\mathbf{b}}^\intercal & \bar{z} & 0 \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\tau \\ \Delta\theta \end{pmatrix} + \begin{pmatrix} \Delta\mathbf{s} \\ \mathbf{0} \\ \Delta\varkappa \\ 0 \end{pmatrix} = \begin{pmatrix} -A^\intercal\mathbf{y} + \mathbf{c}\tau - \bar{\mathbf{c}}\theta - \mathbf{s} \\ A\mathbf{x} - \mathbf{b}\tau + \bar{\mathbf{b}}\theta \\ -\mathbf{c}^\intercal\mathbf{x} + \mathbf{b}^\intercal\mathbf{y} + \bar{z}\theta \\ \bar{\mathbf{c}}^\intercal\mathbf{x} - \bar{\mathbf{b}}^\intercal\mathbf{y} - \bar{z}\tau \end{pmatrix}. \tag{19}$$

Note that if the point is already feasible, the right-hand side is equal to zero.

The second part of the Newton system is the linearized conditions for arriving at the point on the central path with duality gap $\sigma\mu$. That is, we aim for $(\mathbf{x} + \Delta\mathbf{x}) \circ (\mathbf{s} + \Delta\mathbf{s}) = \sigma\mu\mathbf{e}$ and $(\varkappa + \Delta\varkappa)(\tau + \Delta\tau) = \sigma\mu$. By ignoring second-order terms (i.e., the $\mathcal{O}(\Delta\mathbf{x} \circ \Delta\mathbf{s})$ and $\mathcal{O}(\Delta\varkappa\Delta\tau)$ terms), these become

$$\begin{aligned} \mathbf{x} \circ \Delta\mathbf{s} + \mathbf{s} \circ \Delta\mathbf{x} &= \sigma\mu\mathbf{e} - \mathbf{x} \circ \mathbf{s}, \\ \varkappa\Delta\tau + \tau\Delta\varkappa &= \sigma\mu - \varkappa\tau. \end{aligned} \tag{20}$$

The above expression can be rewritten as a matrix equation by first defining the arrowhead matrix U for a vector $\mathbf{u} = (u_0; \tilde{\mathbf{u}}) \in \mathcal{Q}^k$ as

$$U = \begin{pmatrix} u_0 & \tilde{\mathbf{u}}^\intercal \\ \tilde{\mathbf{u}} & u_0 I \end{pmatrix} = \mathbf{u}\mathbf{e}^\intercal + \mathbf{e}\mathbf{u}^\intercal + u_0 I - 2u_0\mathbf{e}\mathbf{e}^\intercal. \tag{21}$$

When $\mathbf{u} \in \mathcal{Q}$ lies in the direct product of multiple second-order cones, the arrowhead matrix is formed by placing the appropriate matrices of the above form on the block diagonal. The arrowhead matrix has the property that, for any vector $\mathbf{v}$, $U\mathbf{v} = \mathbf{u} \circ \mathbf{v}$.

Using this notation, the Newton equations in Eq. (20) can be written as

$$\begin{pmatrix} S & 0 & 0 & 0 & X & 0 \\ 0 & 0 & \varkappa & 0 & 0 & \tau \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\tau \\ \Delta\theta \\ \Delta\mathbf{s} \\ \Delta\varkappa \end{pmatrix} = \begin{pmatrix} \sigma\mu\mathbf{e} - X\mathbf{s} \\ \sigma\mu - \varkappa\tau \end{pmatrix}, \tag{22}$$

where X and S are the arrowhead matrices for vectors $\mathbf{x}$ and $\mathbf{s}$.

Equations (19) and (22) together form the Newton system. We can see that there are $2N + K + 3$ constraints to match the $2N + K + 3$ variables in the vector $(\Delta\mathbf{x}; \Delta\mathbf{y}; \Delta\tau; \Delta\theta; \Delta\mathbf{s}; \Delta\varkappa)$. In Ref. [53], it is shown that, as long as the duality gap is positive and $(\mathbf{x}; \mathbf{y}; \tau; \theta; \mathbf{s}; \varkappa)$ is not too far from the central path (which will be the case as long as σ is chosen sufficiently close to 1 in every iteration), the Newton system has a single unique solution. Note that one can choose different *search directions* than the one that arises from solving the Newton system presented here; this consists of first applying a scaling transformation to the product of second-order cones, then forming and solving the Newton system that results, and finally applying the inverse scaling transformation. Alternative search directions are explained in Appendix D but in the main text we stick to the basic search direction illustrated above, since in our numerical simulations the simple search direction gave equal or better results than more complex alternatives and it enjoys the same theoretical guarantee of convergence [53].

5. Solving the Newton system

The Newton system formed by combining Eqs. (19) and (22) is an $L \times L$ linear system of the form $G\mathbf{u} = \mathbf{h}$, where $L = 2N + K + 3$. Classically, this can be solved *exactly* in a number of ways, the most straightforward being Gaussian elimination, which scales as $\mathcal{O}(L^3)$. Using Strassen-like tricks [54], this can be asymptotically accelerated to $\mathcal{O}(L^\omega)$, where $\omega < 2.38$ [55], although in practice the runtime is closer to $\mathcal{O}(L^3)$. Meanwhile, the linear system can be *approximately* solved using a variety of iterative solvers, such as conjugate gradient descent or the randomized Kaczmarz method [56]. The complexity of these approaches depends on the condition number of the Newton matrix. Section IV discusses *quantum* approaches to solving the Newton system.

It is important to distinguish between methods that exactly solve the Newton system and methods that solve it inexactly, because inexact solutions typically lead to infeasible intermediate points. As presented above, the Newton system in Eqs. (19) and (22) can tolerate infeasible intermediate points; the main consequence is that the right-hand

side of Eq. (19) becomes nonzero. This inexact formulation was the one pursued by Kerenidis, Prakash, and Szilágyi [13], who first examined QIPMs for SOCP (although they did not implement the self-dual embedding as we have done). However, it was pointed out in Refs. [11,14] that the theoretical convergence analysis that Ref. [13] relies upon requires intermediate points to be exactly feasible (i.e., the right-hand side of Eq. (19) is always zero) and that analyses allowing for infeasibility generally have poorer guaranteed convergence time [although in practice they can be just as fast [42]]. As discussed in Sec. IV, exact feasibility is difficult to maintain in quantum IPMs, since the Newton system cannot be solved exactly.

Reference [14] proposed a workaround by which exact feasibility can be maintained despite an inexact linear-system solver, which the authors call an *inexact-feasible IPM* (IF-IPM). For the IF-IPM, we assume that we have access to a basis for the null space of the feasibility constraint equations, i.e., a linearly independent set of solutions to Eq. (19) when the right-hand side is zero. We arrange these basis vectors as the columns of a matrix B; since there are $N+K+2$ linear feasibility constraints and $2N+K+3$ variables, the matrix B should have $N+1$ columns. In the case of PO, a matrix B satisfying this criterion can be deduced by inspection, as discussed in Appendix C; however, this choice does not yield a B with orthogonal columns. Generation of a B with orthonormal columns can be done by performing a QR decomposition of the matrix in Eq. (19), which would incur a large one-time classical cost of $\mathcal{O}((N+K)^3)$ operations [57]. In either case, since B is a basis for the null space of the constraint equations, there is a one-to-one correspondence between vectors $\Delta\mathbf{z} \in \mathbb{R}^{N+1}$ and vectors that satisfy Eq. (19) via the relation $(\Delta\mathbf{x}; \Delta\mathbf{y}; \Delta\tau; \Delta\theta; \Delta\mathbf{s}; \Delta\varkappa) = B\Delta\mathbf{z}$. Thus, our Newton system can be reduced to

$$\left[\begin{pmatrix} S & 0 & 0 & 0 & X & 0 \\ 0 & 0 & \varkappa & 0 & 0 & \tau \end{pmatrix} B\right] \Delta\mathbf{z} = \begin{pmatrix} \sigma\mu\mathbf{e} - Xs \\ \sigma\mu - \varkappa\tau \end{pmatrix}, \quad (23)$$

$$(\Delta\mathbf{x}; \Delta\mathbf{y}; \Delta\tau; \Delta\theta; \Delta\mathbf{s}; \Delta\varkappa) = B\Delta\mathbf{z}. \quad (24)$$

The above Newton system can be solved by first computing $\Delta\mathbf{z}$ by inverting the quantity in brackets in the first line and applying it to the right-hand side and then computing $(\Delta\mathbf{x}; \Delta\mathbf{y}; \Delta\tau; \Delta\theta; \Delta\mathbf{s}; \Delta\varkappa)$ by performing the multiplication $B\Delta\mathbf{z}$. This matrix-vector product can be accomplished classically in $\mathcal{O}(N^2)$ operations. Note that matrix-matrix products where one of the matrices is an arrowhead matrix (S or X) can also be carried out in $\mathcal{O}(N^2)$ classical time, as the form of arrowhead matrices given in Eq. (21) implies that the product can be computed by summing several matrix-vector products. Finally, note that since the second and fourth block columns of the first matrix in Eq. (22) are zero, the second and fourth block rows of B [e.g., in Eq. (C1)] can be completely omitted from the calculation.

Thus, we see three main choices for how to run the IPM when the solution to linear systems is inexact: first, by solving Eqs. (19) and (22) directly and allowing intermediate solutions to be infeasible; second, by finding a matrix B by inspection as described in Appendix C and then solving Eqs. (23) and (24); and, third, by finding a matrix B via QR decomposition and then solving Eqs. (23) and (24). When the linear system is solved using a quantum algorithm, as discussed in Sec. IV, we refer to the algorithms that result from each of these three options by II-QIPM, IF-QIPM, and IF-QIPM-QR, respectively. The pros and cons of each method are summarized in Table II.

6. Neighborhood of the central path and polynomial convergence

The prior literature establishes that if sufficiently small steps are taken (i.e., if σ is sufficiently close to 1), then each intermediate point stays within a small neighborhood of the central path. We now review these conclusions. Following Ref. [53], for a vector $\mathbf{u} = (u_0; \tilde{\mathbf{u}}) \in \mathcal{Q}^k$, we define the matrix

$$T_{\mathbf{u}} = \begin{pmatrix} u_0 & \tilde{\mathbf{u}}^\intercal \\ \tilde{\mathbf{u}} & \sqrt{u_0^2 - \|\tilde{\mathbf{u}}\|^2}\, I + \frac{\tilde{\mathbf{u}}\tilde{\mathbf{u}}^\intercal}{u_0 + \sqrt{u_0^2 - \|\tilde{\mathbf{u}}\|^2}} \end{pmatrix}, \quad (25)$$

which, as for the arrowhead matrix, generalizes to the product of multiple cones by forming a block diagonal of matrices of the above form. We use the distance metric

TABLE II. The choices on which version of the Newton system to solve lead to different versions of the QIPM, even with the same underlying quantum subroutines.

	II-QIPM	IF-QIPM	IF-QIPM-QR
Newton system	Eqs. (19) and (22)	Eqs. (23) and (24)	Eqs. (23) and (24)
Size of Newton system (L)	$2N+K+3$	$N+1$	$N+1$
Feasible intermediate points	No	Yes	Yes
Caveats	Theoretical convergence guarantee requires $\mathcal{O}(r^2)$ [rather than $\mathcal{O}(\sqrt{r})$)] iterations	Ill-conditioned null-space basis leads to large condition number of Newton system	Requires classical QR decomposition, which could dominate overall runtime

defined in Ref. [53]:

$$d_F(\mathbf{x},\tau,\mathbf{s},\varkappa) = \sqrt{2}\sqrt{\|T_{\mathbf{x}}\mathbf{s}-\mu(\mathbf{x},\tau,\mathbf{s},\varkappa)\mathbf{e}\|^2+(\tau\varkappa-\mu(\mathbf{x},\tau,\mathbf{s},\varkappa))^2}. \tag{26}$$

The distance metric induces a neighborhood $\mathcal{N}$, which includes both feasible and infeasible points, as well as the neighborhood $\mathcal{N}_F$, which includes only feasible points:

$$\mathcal{N}(\gamma) = \{(\mathbf{x};\mathbf{y};\tau;\theta;\mathbf{s};\varkappa) : d_F(\mathbf{x},\tau,\mathbf{s},\varkappa) \leq \gamma\mu(\mathbf{x},\tau,\mathbf{s},\varkappa)\}, \tag{27}$$

$$\mathcal{N}_F(\gamma) = \mathcal{N}(\gamma)\cap\mathcal{P}_F, \tag{28}$$

where $\mathcal{P}_F$ denotes the set of feasible points for the self-dual SOCP. Note that the vector $T_{\mathbf{x}}s$ can be computed classically in $\mathcal{O}(N)$ time given access to the entries of $\mathbf{x}$ and $\mathbf{s}$. Thus, whether or not a point lies in $\mathcal{N}(\gamma)$ can be determined in $\mathcal{O}(N)$ time.

Reference [53, Corollary 1] then implies that, so long as $0 \leq \gamma \leq 1/3$ and $(\mathbf{x};\mathbf{y};\tau;\theta;\mathbf{s};\varkappa) \in \mathcal{N}_F(\gamma)$, then we have

$$(\mathbf{x}+\Delta\mathbf{x};\mathbf{y}+\Delta\mathbf{y};\tau+\Delta\tau;\theta+\Delta\theta;\mathbf{s}+\Delta\mathbf{s};\varkappa+\Delta\varkappa) \in \mathcal{N}_F(\Gamma), \tag{29}$$

where

$$\Gamma = \frac{4(\gamma^2+2(r+1)(1-\sigma)^2)}{(1-3\gamma)^2\sigma}. \tag{30}$$

Thus, if $\Gamma \leq \gamma$, and assuming that the Newton system is solved exactly, every intermediate point will lie in $\mathcal{N}_F(\gamma)$. This condition is met if, e.g., $\gamma = 1/10$ and $\sigma = 1-(20\sqrt{2}\sqrt{(r+1)})^{-1}$. Since each iteration reduces the duality gap by a factor σ, the duality gap can be reduced to ϵ after roughly only $20\sqrt{2(r+1)}\ln(1/\epsilon)$ iterations. If the Newton system is solved inexactly but such that feasibility is preserved (e.g., by solving inexactly for $\Delta\mathbf{z}$ and then multiplying by B, as described above), then an error δ on the vector $(\mathbf{x};\tau;\mathbf{s};\varkappa)$ can be tolerated and the resulting vector can still be within the neighborhood at each iteration.

On the other hand, if the Newton system is not solved exactly, then the resulting vector may not be feasible. Since $\mathcal{N}_F(\gamma)$ is defined as a subset of the feasible space, the analysis of Ref. [53] breaks down (as pointed out in Refs. [11,14]). Thus, the II-QIPM version of the QIPM does not enjoy the theoretical guarantee of convergence in $\mathcal{O}(\sqrt{r})$ iterations that the IF-QIPM and IF-QIPM-QR versions do (see Table II). The best guarantees for the II-QIPM would imply convergence only after $\mathcal{O}(r^2)$ iterations [11,14]. Nevertheless, it is unclear whether a small amount of infeasibility makes a substantial difference in practice: we have simulated multiple versions of the QIPM and observed similar overall performance when intermediate solutions were allowed to be infeasible, despite an inferior theoretical guarantee of success. Thus, in Secs. V and VI, where we present the full QIPM implementation, resource count, and numerical analysis, we focus on the II-QIPM. In Appendix E, we present some of the results of our numerical simulations of the IF-QIPM and IF-QIPM-QR algorithms.

IV. QUANTUM INTERIOR-POINT METHODS (QIPMs)

A. Basic idea of QIPM

As discussed in Sec. III, each iteration of an IPM SOCP solver involves forming and solving a linear system of equations that depends on the intermediate point at the current iteration. For classical IPM implementations for SOCP, the linear systems of equations are typically solved exactly; e.g., the numerical SOCP-solving package ECOS solves linear systems with a sparse LDL (Cholesky) factorization [25]. For arbitrary dense systems, the runtime of solving an $L\times L$ system in this way is $\mathcal{O}(L^3)$ [58] but by exploiting sparsity the actual runtime in practice could be much faster, by an amount that is hard to assess. Alternatively, it would, in principle, be possible to employ classical iterative approximate linear-system solvers such as conjugate gradient descent or the randomized Kaczmarz method. The choice of the linear-system solver thereby determines the overall complexity of the IPM SOCP solver. The idea of QIPM, as pioneered in Refs. [10,11], is to use a quantum subroutine to solve the linear system of equations [15]. Notably, all other steps of IPMs stay classical and remain the same as described in Sec. III. As a QLSS does not solve the exact same mathematical problem as classical linear-system solvers and, moreover, a QLSS needs coherent (quantum) access to the classical data as given by the entries of the relevant matrices, there are various additional tools that we will discuss that allow us to embed QLSS subroutines as a step of IPM SOCP solvers.

First, we discuss in Sec. IV B the input and output model of QLSSs and present the complexity of state-of-the-art QLSSs. Then, in Sec. IV C, we give constructions based on QRAM to load classical data as input into a QLSS and discuss the complexity overhead arising from that step. Subsequently, in Sec. IV D, we present so-called pure-state quantum tomography, which allows us to convert the output of the QLSS into an estimate of the classical solution vector of the linear system of equations. Finally, in Sec. IV E, we put all the steps together and state the overall classical and quantum complexities of using QLSSs as a subroutine in IPM SOCP solvers. As described in previous work [22], the ultimate idea is to compare these costs to the complexities of classical IPM SOCP solvers and point

out regimes where quantum methods can potentially scale better than any purely classical methods (e.g., in terms of the SOCP size N, the matrix condition number κ, etc.)

We note that the content of this section largely corresponds to collecting various state-of-the-art results from the prior literature. These ingredients are used together with the conceptual framework of Refs. [10,11,14,22] to lift the QIPMs presented there to superior efficiency. In Sec. V, we present a few novel enhancements to the implementation of the QIPM and fully explicit end-to-end quantum circuits with corresponding novel finite-size complexities.

B. Quantum linear-system solvers

For our purposes, a linear system of equations is given by a real invertible $L \times L$ matrix G together with a real vector $\mathbf{h} = (h_1, \ldots, h_L)$ and one is looking to give an estimate of the unknown solution vector $\mathbf{u} = (u_1, \ldots, u_L)$ defined by $G\mathbf{u} = \mathbf{h}$. We define the (Frobenius) condition number

$$\kappa_F(G) := \|G\|_F \left\|G^{-1}\right\|, \tag{31}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\|\cdot\|$ for a matrix argument denotes the spectral norm.

For this setting, the input to a QLSS is then comprised of: (i) a preparation unitary $U_{\mathbf{h}}$ that creates the $\ell := \lceil \log L \rceil$ qubit quantum state

$$|\mathbf{h}\rangle := \|\mathbf{h}\|^{-1} \cdot \sum_{i=1}^{L} h_i |i\rangle \quad \text{via } |\mathbf{h}\rangle = U_{\mathbf{h}} |0\rangle^{\otimes \ell}, \tag{32}$$

where $\|\cdot\|$ for a vector argument denotes the vector two-norm (standard Euclidean norm); (ii) a block-encoding unitary U_G in the form

$$U_G := \begin{pmatrix} \frac{G}{\|G\|_F} & \cdot \\ \cdot & \cdot \end{pmatrix} \tag{33}$$

on $\ell + \ell_G$ qubits for some $\ell_G \in \mathbb{N}$; and (iii) an approximation parameter $\varepsilon_{\mathrm{QLSP}} \in (0, 1]$. The quantum linear-system problem (QLSP) is stated as follows. For a triple $(G, \mathbf{h}, \varepsilon_{\mathrm{QLSP}})$ as above, the goal is to create an ℓ-qubit quantum state $|\tilde{\mathbf{v}}\rangle$ such that [59]

$$\left\| |\tilde{\mathbf{v}}\rangle - |\mathbf{v}\rangle \right\| \leq \varepsilon_{\mathrm{QLSP}} \quad \text{for } |\mathbf{v}\rangle := \frac{\sum_{i=1}^{L} u_i |i\rangle}{\left\| \sum_{i=1}^{L} u_i |i\rangle \right\|}, \tag{34}$$

defined by $G\mathbf{u} = \mathbf{h}$ with $\mathbf{u} = (u_1, \ldots, u_L)$, by employing, as few times as possible, the unitary operators $U_G, U_{\mathbf{h}}$, their inverses $U_G^\dagger, U_{\mathbf{h}}^\dagger$, controlled versions of $U_G, U_{\mathbf{h}}$, and additional quantum gates on potentially additional ancilla qubits. The QLSP together with the first QLSS was introduced in Ref. [15] and then gradually improved in Refs. [16,17,19,60,61]. The state-of-the-art QLSS [18], using the fewest calls to $U_G, U_{\mathbf{h}}$ and their variants, is based on ideas from discrete adiabatic evolution [62]. We note the following explicit complexities from Ref. [18, Theorem 9], adapted to our setting.

Proposition 1.—The QLSP for $(G, \mathbf{h}, \varepsilon_1)$ can be solved with a quantum algorithm on $\lceil \log_2(L) \rceil + 4$ qubits for

$$\varepsilon_1 \leq C \cdot \frac{\kappa_F(G)}{Q} + \mathcal{O}\left(\frac{\sqrt{\kappa_F(G)}}{Q}\right) \tag{35}$$

for some constant $C \leq 44864$ using $Q \geq \kappa_F(G)$ controlled queries to each of U_G and $U_G^\dagger$ and $2Q$ queries to each of $U_{\mathbf{h}}$ and $U_{\mathbf{h}}^\dagger$ and constant quantum gate overhead.

We note that a stronger version of the above proposition works with the (regular) condition number $\kappa(G) := \|G\|\|G^{-1}\|$ but it requires a block-encoding of the form Eq. (33), in which the normalization factor is $\|G\|$ rather than $\|G\|_F$. For general matrices of classical data, we do not know of a method to produce such a block-encoding. In our case, we work with the Frobenius version $\kappa_F(G)$, since we do have a straightforward method to perform U_G with normalization factor $\|G\|_F$, described in Sec. IV C. It is then sufficient to give upper bounds for the remaining $\kappa_F(G)$ to run the algorithm from Proposition 1. In practice, we will give such upper bounds by using appropriate heuristics (cf. Sec. V on implementations).

Note that Proposition 1 implies a solution to the QLSP in Eq. (34) with an asymptotic query complexity of $\mathcal{O}(\kappa_F/\varepsilon_{\mathrm{QLSP}})$ to U_G, $U_{\mathbf{h}}$, and their variants and under standard complexity-theoretic assumptions this is optimal in terms of the scaling $\mathcal{O}(\kappa)$ [15] but not in terms of the scaling $\mathcal{O}(\varepsilon_{\mathrm{QLSP}})$. To get to an improved $\mathcal{O}(\log(1/\varepsilon_{\mathrm{QLSP}}))$ scaling, the authors of Ref. [18] further rely on the eigenstate-filtering method of Ref. [61, Sec. 3] that additionally invokes a quantum singular-value transform based on a minimax polynomial. We note the following overall complexities from Ref. [18, Theorem 11], adapted to our setting.

Proposition 2.—The QLSP problem for $(G, \mathbf{h}, \varepsilon_2)$ can be solved with a quantum algorithm on $\lceil \log_2(L) \rceil + 5$ qubits that produces a quantum state

$$\sqrt{p}\, |0^5\rangle |\tilde{\mathbf{v}}\rangle + \sqrt{1-p}\, |\perp\rangle |\mathrm{fail}\rangle \tag{36}$$

with $\langle 0^5 | \perp \rangle = 0$ and success probability $p \geq 1/2$. With that, the sought-after ε_2-approximate solution quantum state $|\tilde{\mathbf{v}}\rangle$ can be prepared using $Q + d$ controlled queries to each of U_G and $U_G^\dagger$, and $2Q + 2d$ queries to each of $U_{\mathbf{h}}$ and $U_{\mathbf{h}}^\dagger$, where

$$Q = \frac{1}{\sqrt{2 - \sqrt{2}}} C \kappa_F(G) + \mathcal{O}\left(\sqrt{\kappa_F(G)}\right), \tag{37}$$

$$d = 2\kappa_F(G)\ln(2/\varepsilon_2). \tag{38}$$

Here, $C \leq 44864$ is the same constant as in Proposition 1.

This version of the algorithm essentially uses Proposition 1 with a constant choice of $\varepsilon_1 \leq \sqrt{2-\sqrt{2}}$, which ensures that the state prepared has overlap at least $1/\sqrt{2}$ with the ideal state $|\mathbf{v}\rangle$. Then, it uses eigenstate filtering to measure whether the final state is the correct solution state. On average, we need to repeat the algorithm no more than twice to produce the desired state, $|\tilde{\mathbf{v}}\rangle$. The resulting scaling that Proposition 2 implies for the QLSP problem in Eq. (34) is $\mathcal{O}(\kappa\log(1/\varepsilon_{\mathrm{QLSP}}))$. Following the findings from Ref. [18, Sec. V], we note that in practice the $Q = 1.31C\kappa_F(G)$ dominates over d and all other terms can be safely neglected for typical settings—even for finite-scale analyses. Moreover, the constant C is typically an order of magnitude smaller than the estimates given in Ref. [18, Sec. IV.E]; numerical estimates produced a smaller value of 2305. No direct estimates for general matrices G are available from Ref. [18] but we will henceforth assume $C = 2305$ for our numerical estimates. Additionally, note that for the eigenstate-filtering step via quantum singular-value transform (QSVT), the minimax polynomial from Ref. [61, Sec. 3] and its corresponding quantum signal-processing angles have to be computed. This is done as part of classical preprocessing [63, Sec. III] (see also Ref. [64]).

Note that the implementation of the QLSS in each of Propositions 1 and 2 assumes perfect implementation of the underlying circuits, without additional gate-synthesis errors. In practice, however, these circuits will not be implemented perfectly and hence we will later include additional sources of error (e.g., block-encoding error, imperfect rotation gates, etc.) that also contribute to $\varepsilon_{\mathrm{QLSP}}$. We include these additional contributions in, e.g., Sec. IV D.

In the following, we continue by laying out the additional classical and quantum resources needed to employ QLSS for estimating, in an end-to-end fashion, the classical solution vector $\mathbf{v} = (v_1, \ldots, v_L)$ instead of the quantum state $|\mathbf{v}\rangle$.

C. Block-encoding via quantum random access memory (QRAM)

In many quantum algorithms (and, in particular, for our use case), one needs coherent access to classical data for use in the algorithm. *Block-encodings* of matrices provide a commonly used access model for the classical data by encoding matrices into unitary operators, thereby providing oracular access to the data. As mentioned above, for a matrix $G \in \mathbb{R}^{L\times L}$, a unitary matrix U_G block-encodes G when the top-left block of U_G is proportional to G, i.e.,

$$U_G = \begin{pmatrix} G/\alpha & \cdot \\ \cdot & \cdot \end{pmatrix}, \tag{39}$$

where $\alpha \geq \|G\|$ is a normalization constant, chosen as $\alpha = \|G\|_F$ for our use case. The other blocks in U_G are irrelevant but they must be encoded such that U_G is unitary. For our purposes, we focus on real matrices G but the extension to complex matrices is straightforward. A block-encoding makes use of unitaries that implement (controlled) state preparation, as well as QRAM data structures for loading the classical data. Specifically, we refer to QRAM as the quantum circuit that allows query access to classical data in superposition,

$$\sum_j \psi_j\,|j\rangle\,|0\rangle \xrightarrow{\mathrm{QRAM}} \sum_j \psi_j\,|j\rangle\,|a_j\rangle\,, \tag{40}$$

where j is the address in superposition with amplitude ψ_j and $|a_j\rangle$ is the classical data loaded into a quantum state. There are several models of QRAM that one can use that differ in the way in which the data is loaded. The two most notable QRAM models are the select-swap (SS) model, which is particularly efficient in terms of T-gate utilization [65], and the bucket-brigade (BB) model [66], which has reduced susceptibility to errors when operated on potentially faulty hardware [67].

The block-encoding unitary U_G acts on $\ell + \ell_G$ qubits, where $\ell = \lceil \log_2(L)\rceil$ and, in our construction, $\ell_G = \ell$. To build it, we follow the prescription of Refs. [47,68,69], in which one forms U_G as the product of a pair of controlled-state-preparation unitaries U_L and U_R. Specifically,

$$U_G = U_R^\dagger U_L, \tag{41}$$

$$U_R : |0\rangle^{\otimes l}\,|j\rangle \mapsto |\psi_j\rangle\,|j\rangle\,, \tag{42}$$

$$U_L : |0\rangle^{\otimes l}\,|k\rangle \mapsto |k\rangle\,|\phi_k\rangle\,, \tag{43}$$

where the ℓ-qubit states $|\psi_j\rangle$ and $|\phi_k\rangle$ are determined from the matrix elements G_{jk} of G as follows:

$$|\psi_j\rangle = \sum_k \frac{G_{jk}}{\|G_{j,\cdot}\|}\,|k\rangle\,, \tag{44}$$

$$|\psi_k\rangle = \sum_j \frac{\|G_{j,\cdot}\|}{\|G\|_F}, \tag{45}$$

where $G_{j,\cdot}$ denotes the jth row of G. That is, controlled on the second ℓ-qubit register in the state $|j\rangle$, U_R prepares the ℓ-qubit state $|\psi_j\rangle$ into the first ℓ-qubit register and U_L performs the same operation for the states $|\phi_k\rangle$ modulo a swap of the two registers. Both U_L and U_R utilize an additional ℓ' QRAM ancilla qubits that begin and end in the state $|0\rangle$. These controlled-state-preparation unitaries U_R and U_L are implemented by combining a QRAM-like data-loading step with a protocol for state preparation of ℓ-qubit states. There are several combinations of state-preparation procedure and QRAM model that one can choose, with varying

TABLE III. The logical quantum resources required to block-encode (left column) and control block-encode (right column) an $L \times L$ matrix G to precision $\varepsilon_G \in [0, 1]$, where we assume that $L = 2^\ell$. Here, we have suppressed terms doubly and triply logarithmic in L and $1/\varepsilon_G$ (see Ref. [33]).

Resource	Block-encoding	Controlled block-encoding
Number of qubits	$N_{\mathrm{Qbe}} := 4L^2 - 3L + 2\ell - 1$	$N_{\mathrm{Qcbe}} := N_{\mathrm{Qbe}} + L$
T-depth	$T_{\mathrm{Dbe}} := 10\ell + 24\log_2(1/\varepsilon_G) + 44$	$T_{\mathrm{Dcbe}} := T_{\mathrm{Dbe}} + 4$
T-count	$T_{\mathrm{Cbe}} := (12\log_2(1/\varepsilon_G) + 56)L^2 - 24L - 12\log_2(1/\varepsilon_G) - 32\ell - 32$	$T_{\mathrm{Ccbe}} := T_{\mathrm{Cbe}} + 16(L-1)$

benefits and resource requirements. In Ref. [33], a subset of the authors of the present work have studied the resources required to implement these block-encodings and provided explicit circuits for their implementation. For our immediate purposes, we will simply import the relevant resource estimates from that work in Table III, and we refer the interested reader to Ref. [33] for further details [70]. For our purposes, we will work with the minimum depth circuits that achieve a T-gate depth of $\mathcal{O}(\log L)$, at the price of using a total number of $\mathcal{O}(L^2)$ qubits for the data structure implementing the block-encoding unitary U_G. Finally, the ℓ-qubit unitary $U_{\mathbf{h}}$ defined by $|\mathbf{h}\rangle = U_{\mathbf{h}}\,|0\rangle^{\otimes \ell}$ corresponds to the special case of quantum state preparation and is directly treated by the methods outlined in Ref. [33, Sec. III.C]. The resources required to synthesize $U_{\mathbf{h}}$ up to error $\varepsilon_{\mathbf{h}}$ are also reported in Table III.

The minimum-depth block-encodings of Ref. [33] also incur some classical costs. Specifically, the quoted depth values are only achievable assuming that a number of angles have been classically precomputed and, for each angle, a gate sequence of single-qubit Clifford and T gates that synthesizes a single-qubit rotation by that angle up to small error. Calculation of one of the angles can be done by summing a subset of the entries of G and computing an arcsin. Meanwhile, circuit synthesis requires applying a version of the Solovay-Kitaev algorithm [71,72]. For the block-encoding procedure, $L(L-1)$ angles and their corresponding gate sequences must be computed, which requires a total runtime of $L^2\mathrm{polylog}(1/\varepsilon_G)$ [72], although this computation is amenable to parallelization. For the state-preparation procedure, $L-1$ angles and their sequences are needed.

D. Quantum state tomography

We have described how we can produce a quantum state $|\tilde{\mathbf{v}}\rangle$ approximating the (real-valued) solution $|\mathbf{v}\rangle$ of a linear system up to precision $\varepsilon_{\mathrm{QLSP}}$. As mentioned in Sec. IV B, in the actual circuit implementation, the approximation error $\varepsilon_{\mathrm{QLSP}}$ accounts for both the inherent error from eigenstate filtering captured in Proposition 2 as well as additional gate-synthesis error arising from imperfect implementation of block-encoding unitaries and single-qubit rotations. The next step is to approximately read out the amplitudes of $|\tilde{\mathbf{v}}\rangle$ into classical form. To start out, we will prove the following proposition, which tells us how many copies of a quantum state are needed to provide a good enough classical description of it, up to a phase on each amplitude. This proposition and its proof are adapted from Ref. [73, Proposition 13], with somewhat sharpened constant factors.

Proposition 3.—Let $0 < \varepsilon, \delta < 1$ and let $|\psi\rangle = \sum_{j\in[L]} \alpha_j\,|j\rangle$ be a quantum state. Then, $\left(5 + \sqrt{21}\right)\varepsilon^{-2}\ln(2L/\delta)/3 < 3.1942\varepsilon^{-2}\ln(2L/\delta)$ measurements of $|\psi\rangle$ in the computational basis suffice to learn an ε-ℓ_∞-norm estimate $|\tilde{\boldsymbol{\alpha}}|$ of $|\boldsymbol{\alpha}|$, with success probability at least $1-\delta$.

We give the proof in Appendix B 1. Recall that Proposition 2 gives a unitary U such that

$$U\,|0^5\rangle\,|0^\ell\rangle = \sqrt{p}\,|0^5\rangle\,|\tilde{\mathbf{v}}\rangle + \sqrt{1-p}\,|\perp\rangle\,|\mathrm{fail}\rangle\,, \tag{46}$$

with $|\tilde{\mathbf{v}}\rangle = \sum_{i=1}^{N} \tilde{v}_i\,|i\rangle$, $\langle 0^5\,|\perp\rangle = 0$, and $p \geq 1/2$. The vector $\tilde{\mathbf{v}}$ may have complex coefficients but it approximates a real vector $\mathbf{v}$ up to some error $\varepsilon_{\mathrm{QLSP}}$ in ℓ_2 norm. Our goal

TABLE IV. The logical quantum resources required to prepare an arbitrary ℓ-qubit quantum state $|\mathbf{h}\rangle$ from classical data (left column) and a single-qubit controlled version (right column) to precision $\varepsilon_{\mathbf{h}} \in (0, 1]$. Here, we have suppressed terms doubly and triply logarithmic in L and $1/\varepsilon_{\mathbf{h}}$ (see Ref. [33]). For a single-qubit control, there are no additional Clifford gates required, which can be observed by examining the state-preparation procedure in Ref. [33, Sec. IIID] and noting that we can prepare the state $|0\rangle\,|0\rangle^{\otimes\ell} + |1\rangle\,|\psi\rangle$ with minor modifications to the procedure that prepares $|\psi\rangle$. First, we use the "flag" qubits to control both the angle loading and unloading steps (rather than just the unloading steps) and, second, we control every flip of the flag qubits in that procedure with the first single-qubit control, thus turning NOT gates into controlled-NOT (CNOT) gates, which are also Clifford. When the control is *on*, the procedure works as before and when the control is *off*, none of the qubits leave the $|0\rangle$ state.

Resource	State preparation	Controlled state preparation
Number of qubits	$N_{\mathrm{Qsp}} := 4L + \ell - 6$	$N_{\mathrm{Qcsp}} := N_{\mathrm{Qsp}} + 1$
T-depth	$T_{\mathrm{Dsp}} := 3\ell + 12\log_2(1/\varepsilon_{\mathbf{h}}) + 24$	$T_{\mathrm{Dcsp}} := T_{\mathrm{Dsp}}$
T-count	$T_{\mathrm{Csp}} := (12\log_2(1/\varepsilon_{\mathbf{h}}) + 40)L - 12\log_2(1/\varepsilon_{\mathbf{h}}) - 16\ell - 40$	$T_{\mathrm{Ccsp}} := T_{\mathrm{Csp}}$

is to obtain an estimate $\tilde{\mathbf{v}}' = (v_1', \ldots, v_N')$ such that

$$\|\mathbf{v} - \tilde{\mathbf{v}}'\| \leq \xi \quad \text{for an error parameter } \xi \in [0, 1]. \tag{47}$$

where ξ captures all sources of error. Proposition 3 is not quite sufficient because it only gives us an estimate of the absolute value of $\tilde{\mathbf{v}}$. However, the following procedure, adapted from Ref. [10, Sec. 4], will be sufficient:

(1) Create $k = 57.5L\ln(6L/\delta)/(\varepsilon^2(1-\varepsilon^2/4))$ many copies of the quantum state $U|0^{5+\ell}\rangle = \sqrt{p}\,|0^5\rangle\,|\tilde{\mathbf{v}}\rangle + \sqrt{1-p}\,|\perp\rangle\,|\text{fail}\rangle$ and measure them all in the computational basis to give empirical estimates $\{p_i\}_{i=1}^L$ of the probabilities $p|\tilde{v}_i|^2$.

(2) Using controlled applications of U, create $k = 57.5L\ln(6L/\delta)/(\varepsilon^2(1-\varepsilon^2/4))$ copies of

$$\begin{aligned} &2^{-1/2}\,|0^5\rangle\,|0\rangle\,\sqrt{p}\,|\tilde{\mathbf{v}}\rangle \\ &\quad + 2^{-1/2}\,|0^5\rangle\,|1\rangle\sum_{i=1}^{L}\sqrt{p_i'}\,|i\rangle \\ &\quad + |\perp'\rangle\,|\text{fail}'\rangle\,, \end{aligned} \tag{48}$$

which, by applying a Hadamard, can be mapped to

$$\begin{aligned} &|0^5\rangle\,|0\rangle\,\frac{\sqrt{p}\,|\tilde{\mathbf{v}}\rangle + \sum_{i=1}^{L}\sqrt{p_i'}\,|i\rangle}{2} \\ &\quad + |0^5\rangle\,|1\rangle\,\frac{\sqrt{p}\,|\tilde{\mathbf{v}}\rangle - \sum_{i=1}^{L}\sqrt{p_i'}\,|i\rangle}{2} \\ &\quad + |\perp'\rangle\,|\text{fail}''\rangle\,. \end{aligned} \tag{49}$$

Here, $|\perp'\rangle$ is an arbitrary state orthogonal to $|0^5\rangle$ and $|\text{fail}'\rangle$ and $|\text{fail}''\rangle$ are arbitrary unnormalized states. The quantities $\sqrt{p_i'}$ are (possibly complex) amplitudes that satisfy $|\sqrt{p_i'} - \sqrt{p_i}| \leq \varepsilon_{\text{tsp}}$ for all i; they arise because the state $\sum_{i=1}^{L}\sqrt{p_i'}\,|i\rangle$ can only be prepared up to some error. Next, measure this state in the computational basis, denoting the measurement count of the result $0^6 i$ as k_i^+ and the result $0^5 1 i$ as k_i^-.

(3) Define

$$a_i^+ = \min\left(\sqrt{p_i}, \frac{k_i^+ - k_i^-}{\sqrt{p_i}}\right), \tag{50}$$

$$a_i^- = \max\left(-\sqrt{p_i}, \frac{k_i^+ - k_i^-}{\sqrt{p_i}}\right), \tag{51}$$

and let

$$\tilde{a}_i = \begin{cases} 0, & \text{if } \sqrt{p_i} \leq \frac{2}{3\sqrt{2L}}\varepsilon\sqrt{1-\frac{\varepsilon^2}{4}} + \varepsilon_{\text{tsp}}, \\ a_i^+, & \text{if } \tilde{a}_i \neq 0 \text{ and } k_i^+ \geq k_i^-, \\ a_i^-, & \text{if } \tilde{a}_i \neq 0 \text{ and } k_i^+ < k_i^-. \end{cases} \tag{52}$$

Output the estimate $|\tilde{\mathbf{v}}'\rangle = \sum_{i=1}^{L}\tilde{a}_i\,|i\rangle\,/\sqrt{\sum_{i=1}^{L}\tilde{a}_i^2}$.

Proposition 4.—Suppose that $\|\tilde{\mathbf{v}} - \mathbf{v}\| \leq \varepsilon_{\text{QLSP}}$ and that $\mathbf{v}$ is a real-valued vector. Let ε and ε_{tsp} be constants that satisfy $\varepsilon + \sqrt{2L}\varepsilon_{\text{tsp}} + \sqrt{2}\varepsilon_{\text{QLSP}} \leq 1/2$. Then, the above algorithm outputs an estimate $\tilde{\mathbf{v}}'$ such that $\|\tilde{\mathbf{v}}' - \mathbf{v}\| < \varepsilon + 1.58\sqrt{L}\varepsilon_{\text{tsp}} + 1.58\varepsilon_{\text{QLSP}}$ with probability $1-\delta$.

We give the proof in Appendix B 1. The statement is used to bound the total error parameter ξ by the quantity $\varepsilon + 1.58\sqrt{L}\varepsilon_{\text{tsp}} + 1.58\varepsilon_{\text{QLSP}}$. We note that a similar procedure in Ref. [10, Sec. 4] has already been proven to work, with somewhat worse success probability guarantees and worse constants. Reference [73, Proposition 16] shows a similar result for complex-valued states but we use a sharper proof for input states close to real valued. Proposition 4, together with Proposition 2, produces with high probability an $\mathcal{O}(\varepsilon)$ good estimate $\tilde{\mathbf{v}}'$ of $\mathbf{v}$ by using $\mathcal{O}(L\ln(L)/\varepsilon^2)$ many samples. If our goal is to resolve the initial linear system $G\mathbf{u} = \mathbf{h}$, then the vector $\tilde{\mathbf{v}}'$, produced as in Sec. IV D as an estimate for the normalized vector $\mathbf{v} = \mathbf{u}/\|\mathbf{u}\|$, gives an estimate for $\mathbf{u}$ via

$$\tilde{\mathbf{u}} := \tilde{\mathbf{v}}' \cdot \frac{\|\mathbf{h}\|}{\|G\tilde{\mathbf{v}}'\|},$$

for which we find

$$\|\mathbf{u} - \tilde{\mathbf{u}}\| \leq \|\mathbf{v} - \tilde{\mathbf{v}}'\| \cdot \big(1 + \kappa(G)\big) \cdot \frac{\|\mathbf{h}\|}{\|G\tilde{\mathbf{v}}'\|}.$$

Note that as a worst-case guarantee, this picks up an additional factor $\kappa(G)$ in error scaling. However, for our purposes it will be sufficient to work directly with the normalized estimate $\tilde{\mathbf{v}}'$ for $\mathbf{v}$, the reason being that only the direction of the solution vector is important to us and not its exact normalization. There are other methods in the literature that allow us to perform pure-state quantum tomography with comparable query complexities (see, e.g., Ref. [74]) but we favor the above method because of its computational simplicity and the fact that it does not require us to solve any potentially costly additional optimization problems. Very recently, the sample complexity has been improved to $\mathcal{O}(L\ln(L)/\varepsilon)$, which comes at the cost of more complicated quantum circuits and higher constant overheads [73, Theorem 23]. It would be interesting to work out the more involved finite complexity of this result and we comment further on the potential impact of this in Sec. VII.

E. Asymptotic quantum complexity

Putting everything together, the steps of our QLSS for given real $L \times L$ matrix G and real vector $\mathbf{h}$ of size L are as follows:

(1) Construct the circuits that implement the block-encoding unitaries U_G and $U_{\mathbf{h}}$ up to error ε_G and $\varepsilon_{\mathbf{h}}$ via quantum state preparation and QRAM, which involves a classical preprocessing cost scaling as $L^2\text{polylog}(1/\varepsilon_{G,\mathbf{h}})$. The quantum resources required are described in Table III. The T-gate depth (what we call time complexity) is $\mathcal{O}(\log L)$ and the total T-gate count is $\mathcal{O}(L^2)$.
(2) Employ the QLSS unitary from Proposition 2 to approximately solve the corresponding QLSP, leading to the quantum state $|\tilde{\mathbf{v}}\rangle$. The query complexity to U_G, $U_{\mathbf{h}}$, their controlled versions, and their inverses, is $\mathcal{O}(\kappa_F(G)\log(1/\varepsilon))$. The number of qubits needed is $\lceil \log L \rceil + 5$.
(3) Repeat the previous step $\mathcal{O}(L\ln(L/\delta)\varepsilon^{-2})$ many times to implement the pure-state quantum tomography scheme from Sec. IV D, which also requires the use of an $\mathcal{O}(L)$ qubit QRAM structure and one ancilla qubit. Tomography leads to the sought-after classical vector estimate $\tilde{\mathbf{v}}'$ with $\|\tilde{\mathbf{v}}' - \mathbf{v}\| \leq \varepsilon$.

The QLSS can then be used for each iteration of an IPM SOCP solver, which involves forming and solving a linear system of equations, resulting in the QIPM SOCP solver. We provide the quantum circuits needed to implement the solver in Sec. IV F. However, we emphasize that we have not yet considered the various practical aspects and difficulties of setting up an *end-to-end* QIPM SOCP solver, which is discussed further in Sec. V.

F. Quantum circuits

The following are the quantum circuits needed for the QLSS of Proposition 1. The QLSS requires applying a unitary $U[s]$ for many different values of s, where $U[s]$ is a block-encoding of a certain Hamiltonian related to G and $\mathbf{h}$, as specified below. The unitary acts on $4+\ell+\ell_G$ total qubits, where the final ℓ_G qubits are ancillas associated with U_G. The four single-qubit registers are referred to with labels a_1, a_2, a_3, and a_4, the ℓ-qubit register with label L, and the ℓ_G-qubit register with label ℓ_G. These labels are used as subscripts on bras, kets, and operators to clarify the register to which they apply. The circuit for $U[s]$ is depicted in Fig. 1 and is described in Ref. [18, Appendix E]. Specifically, the unitary $U[s]$ is a block-encoding of the $(2+\ell)$-qubit Hamiltonian $c(s)\cdot\mathcal{H}[s] := (1-f(s))\mathcal{H}_0 + f(s)\mathcal{H}_1$ on registers a_4a_1L, where $c(s)$ is a normalization factor [defined later in Eq. (60)],

$$\mathcal{H}_0 := \begin{pmatrix} 0 & 0 & I_L - |\mathbf{h}\rangle\langle\mathbf{h}|_L & 0 \\ 0 & 0 & 0 & -I_L \\ I_L - |\mathbf{h}\rangle\langle\mathbf{h}|_L & 0 & 0 & 0 \\ 0 & -I_L & 0 & 0 \end{pmatrix} \tag{53}$$

and

$$\mathcal{H}_1 := \begin{pmatrix} 0 & 0 & 0 & G \\ 0 & 0 & G^\dagger(I_L - |\mathbf{h}\rangle\langle\mathbf{h}|_L) & 0 \\ 0 & (I_L - |\mathbf{h}\rangle\langle\mathbf{h}|_L)G & 0 & 0 \\ G^\dagger & 0 & 0 & 0 \end{pmatrix}, \tag{54}$$

and where I_L denotes the identity operation on subsystem L, and the four rows and columns correspond to the sectors with qubits a_4 and a_1 set to $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. Figure 1 features the expressions

$$CR^0(s) := |0\rangle\langle 0|_{a_4} \otimes R(s)_{a_2} + |1\rangle\langle 1|_{a_4} \otimes H_{a_2}, \tag{55}$$

$$CR^1(s) := |1\rangle\langle 1|_{a_4} \otimes R(s)_{a_2} + |0\rangle\langle 0|_{a_4} \otimes H_{a_2}, \tag{56}$$

$$V_G := |0\rangle\langle 0|_{a_2} \otimes Z_{a_1} \otimes I_{L\ell_G} + |1\rangle\langle 1|_{a_2} \otimes \begin{pmatrix} 0 & U_G \\ U_G^\dagger & 0 \end{pmatrix}_{a_1 L\ell_G}, \tag{57}$$

where H denotes the single-qubit Hadamard gate, and $R(s)$ is given by

$$R(s) := \frac{1}{\sqrt{(1-f(s))^2 + f(s)^2}} \begin{pmatrix} 1-f(s) & f(s) \\ f(s) & -(1-f(s)) \end{pmatrix}, \tag{58}$$

$$f(s) := \frac{\kappa_F(G)}{\kappa_F(G)-1} \cdot \left(1 - \left(1 + s\left(\sqrt{\kappa_F(G)} - 1\right)\right)^{-2}\right). \tag{59}$$

The normalization factor of $R(s)$ above combines with a factor of $1/\sqrt{2}$ introduced by the Hadamard gate to give an overall normalization factor for $\mathcal{H}(s)$ of

$$c(s) = \left(2((1-f(s))^2 + f(s)^2)\right)^{-1/2} \in [2^{-1/2}, 1] \tag{60}$$

and a scheduling function $f(s)$ with $f(0) = 0$ and $f(1) = 1$. Note that we have the self-inverse property $U[s]^2 = 1\ \forall s \in [0,1]$, as demonstrated in Ref. [18, Appendix E]. The overall quantum circuit U for the quantum algorithm of Proposition 1 is then given as (cf. Ref. [75])

$$U := \prod_{j=1}^{Q} P[1 - j/Q] \tag{61}$$

with the walk operator

$$P[s] := WU[s],$$

where W is the operator that acts as identity on registers a_4a_1L (which host the Hamiltonian $\mathcal{H}[s]$) while performing the reflection $(2|0\rangle\langle 0|_{a_2a_3\ell_G} - I_{a_2a_3\ell_G})$ on the remaining qubits. The unitary U makes Q controlled queries to

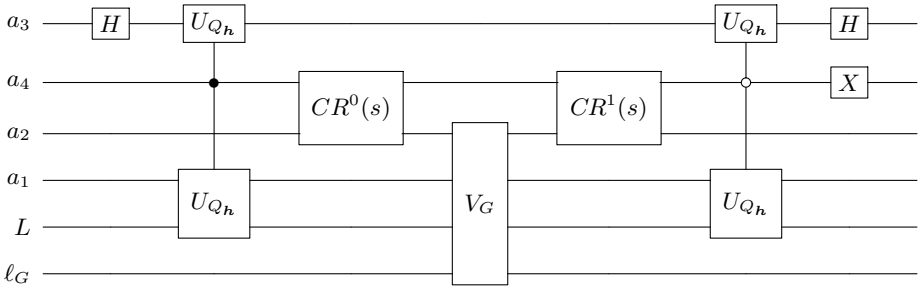

FIG. 1. The main component of the quantum circuit for Proposition 1, described in Ref. [18, Appendix E], enacting the unitary $U[s]$ on registers $a_3a_4a_2a_1L\ell_G$ of the scaled Hamiltonian $c(s)\cdot\mathcal{H}[s]$, where $\mathcal{H}[s]=(1-f(s))\mathcal{H}_0+f(s)\mathcal{H}_1$, on registers a_4a_1L. The necessary quantum gates and functions are defined in Eqs. (53)–(59), except for subcircuit $U_{Q_\mathbf{h}}$, which is depicted in Fig. 4. The unitary $U[s]$ is then used in Eq. (61) to define the overall quantum circuit U for Proposition 1.

each of U_G and $U_G^\dagger$ and $2Q$ queries to each of $U_\mathbf{h}$ and $U_\mathbf{h}^\dagger$, and it has constant quantum gate overhead.

Next, we give the remaining QSVT eigenstate-filtering quantum circuit for the refined QLSS of Proposition 2. We are interested in the null space of $c(1)\cdot\mathcal{H}[1]$, which has ground-state energy equal to zero and spectral gap at least $c(1)\kappa_F^{-1}(G)=(\sqrt{2}\kappa_F)^{-1}$. As such, we employ the Chebyshev minimax polynomial

$$R_l(x,\kappa_F^{-1}(G)):=\frac{T_l\left(-1+2\frac{x^2-\kappa_F^{-2}(G)/2}{1-\kappa_F^{-2}(G)/2}\right)}{T_l\left(-1+2\frac{-\kappa_F^{-2}(G)/2}{1-\kappa_F^{-2}(G)/2}\right)}, \tag{62}$$

where $T_l(\cdot)$ is the lth Chebyshev polynomial of the first kind, as part of the corresponding QSVT quantum circuit. From Ref. [61, Lemma 2], R_l has even degree d equal to

$$d:=2l=2\left\lceil\kappa_F(G)\ln(2/\varepsilon_{\text{qsp}})\right\rceil \quad \text{for some } \varepsilon_{\text{qsp}}\in(0,1) \tag{63}$$

where ε_{qsp} is the precision to which R_l approximates the optimal filter operator. The QSP subscript stands for "quantum signal processing."

The circuit for the eigenstate-filtering step is depicted in Fig. 2. To implement it, one has to classically precompute the corresponding QSP angles $\{\phi_1,\ldots,\phi_d\}$, which is best done by the methods of Ref. [63] (see also Refs. [68] and [76]). The query complexity to the block-encoding $U[1]$ is given by d, the additional gate overhead is as in Fig. 2, and the total number of qubits is $1+4+\ell$. Finally, use of the overall quantum circuit U from Proposition 1 with constant approximation parameter $\varepsilon_1=\sqrt{2-\sqrt{2}}$ therein (to produce an input state to the quantum circuit of Fig. 2) gives the overall quantum circuit of the QLSS from Proposition 2, which then solves the QLSP to error $\varepsilon_2=\varepsilon_{\text{qsp}}$.

The tomography routine also requires the ability to perform controlled versions of the above circuits as described in Eq. (48) and illustrated in Fig. 3 (which replaces Fig. 1). The controlled circuits can be accomplished by rather simple modifications to the circuits in Figs. 1 and 2 as follows.

Any QSVT circuit can be made controlled by simply controlling the application of the z rotation gates, since the rest of the circuit contains only symmetric applications of unitary gates and their inverses. Thus, we can create a controlled version of Fig. 2 by simply performing controlled-σ_z rotations, which requires two CNOT gates and an extra single-qubit σ_z-rotation gate.

Control of the linear-system portion is not enough to implement Eq. (48). One must also follow this with a controlled-state-preparation routine, controlled on the value of the qubit c being in the $|1\rangle$ state. The full resource

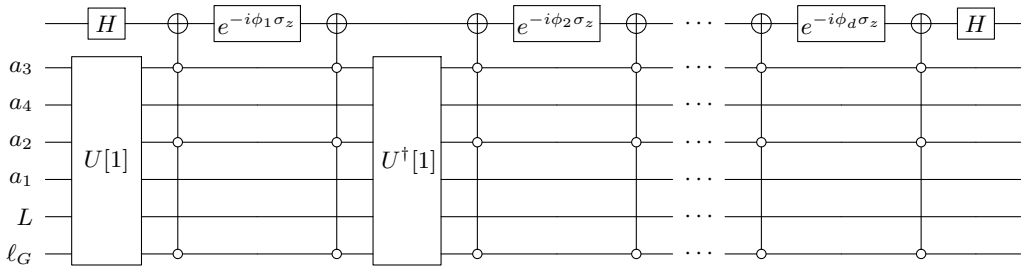

FIG. 2. The quantum singular-value transform (QSVT) circuit, described in Ref. [68], acting on the block-encoding $U[1]$ of $\mathcal{H}(1)=\mathcal{H}_1/\sqrt{2}$, as defined in Eq. (54). The circuit features one additional ancilla qubit and depends on the classically precomputed rotation angles $\{\phi_1,\ldots,\phi_d\}$.

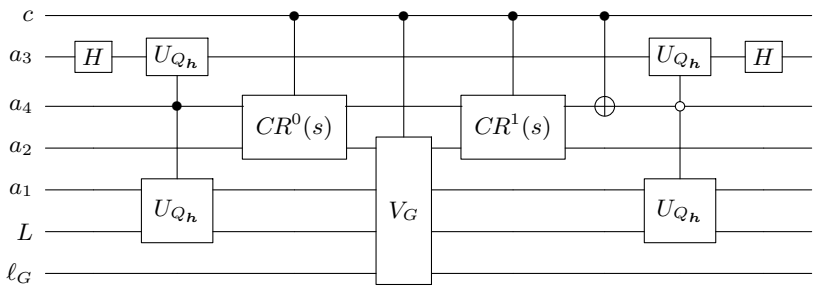

FIG. 3. The controlled version of the quantum circuit in Fig. 1, controlled on qubit c. Note that not all gates need to be controlled on c, as their inverses follow in the circuit.

analysis for controlled state preparation has been reported in Ref. [33] and we refer the reader there for further details. We report the resource counts here, in Table IV.

V. IPM IMPLEMENTATION AND RESOURCE ESTIMATES FOR PO

In Sec. IV, we reviewed the ingredients needed to implement the QIPM, namely, QLSS, block-encoding, and tomography. Here, we combine those ingredients to describe how the QIPM is actually implemented, making several observations that go beyond the prior literature. We also perform a full resource analysis of the entire protocol and report resources needed to run the algorithm.

A. Main IPM loop and full pseudocode

A QIPM is formed from an IPM by performing the step of solving a linear system with a quantum algorithm; the rest of the steps are classical. In Algorithm 1, we present pseudocode for the IPM where the single quantum subroutine—approximately solving a linear system—appears in blue text. The input to Algorithm 1 is an SOCP instance with N variables, K linear constraints, and r second-order cone constraints, along with a tolerance parameter ϵ. Here, we note that $K = \mathcal{O}(N)$ in the case of the formulation of the PO problem that we simulate in Sec. VI. The output of the QIPM is a vector $\mathbf{x}$ that is $\mathcal{O}(\epsilon)$ close to feasible and $\mathcal{O}(\epsilon)$ close to optimal.

The structure of the QIPM is, in essence, the same as that proposed by Ref. [13] but we give a more complete specification of the algorithm and make several new observations:

(a) *Classical costs*. The IPM requires $\mathcal{O}(\sqrt{r}\log(1/\epsilon))$ iterations. In the classical case, when solving the PO problem via SOCP with an IPM, the cost of an iteration is dominated by the time needed to solve a linear system of size $L \times L$, which is $\mathcal{O}(N^3)$ if done via Gaussian elimination, since $L \sim \mathcal{O}(N)$ in the PO problem. In the quantum case, this step is performed quantumly. However, even in the quantum case, some classical costs are incurred: one must *classically* compute the left-hand and right-hand sides of the Newton system in Eqs. (19) and (22) to be able to load these classical data into quantum circuits that perform the QLSS and tomography required to gain a classical estimate of the solution to the linear system. In particular, constructing the linear system requires classical matrix-vector multiplication to compute the residuals on the right-hand side of the Newton system in Eq. (19). If the SOCP constraint matrix A is $\mathcal{O}(N) \times N$ and the number of cones $r = \mathcal{O}(N)$, then this classical matrix-vector multiplication takes $\mathcal{O}(N^2)$ time in each of the $\mathcal{O}(\sqrt{N})$ iterations. Thus, the QIPM requires at least $\mathcal{O}(N^{2.5})$ classical time. Additionally, in our resource counts we use the minimal depth block-encoding circuits from Ref. [33], which require $N^2\text{polylog}(1/\varepsilon)$ classical time per iteration (although this can be parallelized) to compute angles and corresponding gate sequences to precision ε. These classical costs limit the maximum possible speed-up of the QIPM over the classical IPM but if the quantum subroutine is sufficiently fast that classical matrix-vector multiplication and angle computation is the bottleneck step, then this is a good signal for the utility of the QIPM.

(b) *Preconditioning*. Since the runtime of the QLSS depends on the condition number of the matrix G that appears in the linear system $G\mathbf{u} = \mathbf{h}$, it is worth examining preconditioning techniques [77] for reducing the condition number. In the implementation that we propose, we perform a very simple form of preconditioning. Let D be a diagonal matrix, where entry D_{ii} is equal to the norm of row i of the matrix G. Instead of solving the linear system $G\mathbf{u} = \mathbf{h}$, we solve the equivalent system $(D^{-1}G)\mathbf{u} = D^{-1}\mathbf{h}$. Note that $D^{-1}G$ and $D^{-1}\mathbf{h}$ can each be classically computed in $\mathcal{O}(N^2)$ time, roughly equal to the time required to compute $\mathbf{h}$ in the first place [see (a)], so this step is unlikely to be a bottleneck in the algorithm. In our numerical experiments, we observe that the condition number

of $D^{-1}G$ is typically more than an order of magnitude smaller than G and sometimes several orders of magnitude (see Fig. 9 in Sec. VI).

(c) *Norm of linear system and step length*. As discussed in Sec. IV B, QLSSs produce a normalized state $|\mathbf{u}\rangle$, where $\mathbf{u}$ is the solution to $G\mathbf{u} = \mathbf{h}$ and quantum state tomography on $|\mathbf{u}\rangle$ can only reveal the direction of the solution $\mathbf{u}$ and not its norm. The norm can be estimated separately with a comparable amount of resources but we observe that in the context of QIPMs, *it is not necessary to learn the norm of the solution*. If the direction of the solution is known, the amount by which to update the vector in that direction can be determined classically in $\mathcal{O}(N)$ time as follows. If $(\Delta\mathbf{x}; \Delta\mathbf{y}; \Delta\tau; \Delta\theta; \Delta\mathbf{s}; \Delta\varkappa)$ is the normalized solution to the Newton linear system in Eqs. (19) and (22), then the amount to step in that direction is equal to

$$\frac{\mu(\mathbf{x}, \tau, \mathbf{s}, \varkappa)(1-\sigma)(r+1)}{-(\Delta\mathbf{x})^\intercal\mathbf{s} - (\Delta\mathbf{s})^\intercal\mathbf{x} - (\Delta\varkappa)\tau - (\Delta\tau)\varkappa}. \tag{64}$$

This expression is chosen such that the duality gap of the new point is exactly a factor of σ smaller than the old point, up to deviations that are second order in the step length. Note that if the old point is feasible and the solution to the linear system is exact, the second- and higher-order contributions vanish anyway.

(d) *Adaptive tomographic precision and neighborhood detection*. In Ref. [13], the choice of tomography precision parameter ξ was determined by a formula that aimed to guarantee staying within the neighborhood of the central path under a worst-case outcome. We observe that since determining whether a point is within the neighborhood of the central path can be done in classical $\mathcal{O}(N)$ time (see Sec. III C 6), the precision parameter can instead be determined adaptively for optimal results: start with $\xi = 1/2$, solve the linear system to precision ξ, and check if the resulting point is within the neighborhood of the central path. If yes, continue to the next iteration; if no, repeat the tomography with $\xi \leftarrow \xi/2$. Since the complexity of tomography is $\mathcal{O}(1/\xi^2)$, the cost of this adaptive scheme is proportional to a geometric series $4 + 16 + 64 + \ldots + \mathcal{O}(1/\xi^2)$, of which the final term will make up most of the cost (accordingly, for simplicity, in our resource calculation we only account for the final term). This cost could be much lower than the theoretical value if the typical errors are not as adverse for the IPM as a worst-case error of the same size.

The pseudocode in Algorithm 1 illustrates the infeasible version of the algorithm (II-QIPM from Table II). To implement the feasible versions (IF-QIPM and IF-QIPM-QR), minor modifications are made to reflect the process described in Sec. III.

B. End-to-end quantum resource estimates

The QIPM described in the pseudocode takes $20\sqrt{2}\sqrt{r}\ln(\epsilon^{-1})$ iterations to reduce the duality gap to ϵ, where r is the number of second-order cone constraints. In the case of the PO problem that we study, $r = 3n + 1$, where n is the number of stocks in the portfolio. Choosing the constant prefactor to be $20\sqrt{2}$ allows us to utilize theoretical guarantees of convergence (modulo the issue of infeasibility discussed in Sec. III C 5); however, it would not be surprising if additional optimization of the parameters or heuristic changes to the implementation of the algorithm (e.g., adaptive step size during each iteration) were to lead to constant-factor speed-ups in the number of iterations. Since the number of iterations would be the same for both the quantum and classical IPM, these sorts of improvements would not impact the performance of the QIPM relative to its classical counterpart.

1. Quantum circuit compilation and resource estimate for quantum circuits appearing within QIPM

The QIPM consists of repeatedly performing a quantum circuit associated with the QLSS and measuring in the computational basis. Here, we account for all the costs of each of these individual quantum circuits. There are two kinds of circuits that are needed: first, the circuit that creates the output of the QLSS subroutine, given by the state in Eq. (36) and, second, the circuit that creates the state needed to determine the signs of the amplitudes during the tomography subroutine corresponding to a controlled-QLSS subroutine, given in Eq. (48).

To simplify the analysis, we first compile the circuits from the previous section into a primitive gate set that consists of Toffoli gates (and multicontrolled versions of them), rotation gates, block-encoding unitaries, and state-preparation and controlled-state-preparation unitaries. This compilation allows us to combine our previous in-depth resource analysis for these primitive routines [33] with the additional circuits shown here.

From left to right in the $U[s]$ circuit shown in Fig. 1, we show the circuits for $U_{Q_\mathbf{h}}$, $CR^0(s)$ (and, equivalently, $CR^1(s)$), and V_G in Figs. 4–6, respectively. In addition to these circuits, we must also perform controlled versions of them within the tomography routine to estimate the sign of the amplitudes. The controlled-$U[s]$ gate is given in Fig. 3. The implementation of the controlled versions of $CR^0(s)$ (and, equivalently, $CR^1(s)$) and V_G are also depicted in Figs. 5 and 6, respectively.

With these decompositions in place, we now report in Table V the resources required to perform each of the two kinds of quantum circuits involved in the QIPM (each of

Input: SOCP instance $(A, \boldsymbol{b}, \boldsymbol{c})$, list of cone sizes $(N_1, \ldots, N_r)$ and tolerance ϵ
Output: Vector $\boldsymbol{x}$ that optimizes objective function (eq. (5)) to precision ϵ
/* For portfolio optimization, A, $\boldsymbol{b}$, $\boldsymbol{c}$ are given in eq. (10). First n entries of $\boldsymbol{x}$ give optimal stock weights. */

$(\boldsymbol{x}; \boldsymbol{y}; \tau; \theta; \boldsymbol{s}; \varkappa) \leftarrow (\boldsymbol{e}; \boldsymbol{0}; 1; 1; \boldsymbol{e}; 1)$ /* initialize on central path */
$\mu \leftarrow 1, \quad \sigma \leftarrow 1 - \frac{1}{20\sqrt{2}} \frac{1}{\sqrt{r}}, \quad \gamma \leftarrow 1/10$ /* set parameters */
while $\mu \geq \epsilon$: /* Follow central path until duality gap less than ϵ */

$$G \leftarrow \begin{pmatrix} 0 & A^\intercal & -\boldsymbol{c} & \bar{\boldsymbol{c}} & I & 0 \\ -A & 0 & \boldsymbol{b} & -\bar{\boldsymbol{b}} & 0 & 0 \\ \boldsymbol{c}^\intercal & -\boldsymbol{b}^\intercal & 0 & -\bar{z} & \boldsymbol{0} & 1 \\ -\bar{\boldsymbol{c}}^\intercal & \bar{\boldsymbol{b}}^\intercal & \bar{z} & 0 & \boldsymbol{0} & 0 \\ S & 0 & 0 & 0 & X & 0 \\ 0 & 0 & \varkappa & 0 & 0 & \tau \end{pmatrix}$$ /* from eqs. (19) and (22) */

$$\boldsymbol{h} \leftarrow \begin{pmatrix} -A^\intercal \boldsymbol{y} + \boldsymbol{c}\tau - \bar{\boldsymbol{c}}\theta - \boldsymbol{s} \\ A\boldsymbol{x} - \boldsymbol{b}\tau + \bar{\boldsymbol{b}}\theta \\ -\boldsymbol{c}^\intercal \boldsymbol{x} + \boldsymbol{b}^\intercal \boldsymbol{y} + \bar{z}\theta \\ \bar{\boldsymbol{c}}^\intercal \boldsymbol{x} - \bar{\boldsymbol{b}}^\intercal \boldsymbol{y} - \bar{z}\tau \\ \sigma\mu \boldsymbol{e} - \tilde{X}\tilde{S}\boldsymbol{e} \\ \sigma\mu - \varkappa\tau \end{pmatrix}$$ /* mat.-vec. mult. performed classically */

for $j = 1, \ldots, L$: /* preconditioning via row normalization */
  $g \leftarrow \sqrt{\sum_k |G_{jk}|^2}$ /* norm of jth row of G */
  $h_j \leftarrow h_j / g$
  for $k = 1, \ldots, L$:
    $G_{jk} \leftarrow G_{jk} / g$
Classically compute L^2 angles and gate decompositions necessary to perform block-encoding of G and state-preparation of $|\boldsymbol{h}\rangle$ (see Ref. [33])
$\xi \leftarrow 1$
repeat /* try smaller and smaller ξ until central path is found */
  $\xi \leftarrow \xi/2$
  $(\boldsymbol{\Delta x}; \boldsymbol{\Delta y}; \Delta\tau; \Delta\theta; \boldsymbol{\Delta s}; \Delta\varkappa) \leftarrow \texttt{ApprSolve}(G, \boldsymbol{h}, \xi)$
  $\text{(step length)} \leftarrow \frac{\mu(\sigma-1)(r+1)}{(\boldsymbol{\Delta x})^\intercal \boldsymbol{s} + (\boldsymbol{\Delta s})^\intercal \boldsymbol{x} + (\Delta\varkappa)\tau + (\Delta\tau)\varkappa}$
  $(\boldsymbol{x}'; \boldsymbol{y}'; \tau'; \theta'; \boldsymbol{s}'; \varkappa') \leftarrow (\boldsymbol{x}; \boldsymbol{y}; \tau; \theta; \boldsymbol{s}; \varkappa) + \text{(step length)} \cdot (\boldsymbol{\Delta x}; \boldsymbol{\Delta y}; \Delta\tau; \Delta\theta; \boldsymbol{\Delta s}; \Delta\varkappa)$
until $(\boldsymbol{x}'; \boldsymbol{y}'; \tau'; \theta'; \boldsymbol{s}'; \varkappa') \in \mathcal{N}(\gamma)$
$(\boldsymbol{x}; \boldsymbol{y}; \tau; \theta; \boldsymbol{s}; \varkappa) \leftarrow (\boldsymbol{x}'; \boldsymbol{y}'; \tau'; \theta'; \boldsymbol{s}'; \varkappa')$
$\mu \leftarrow \sigma\mu$
return $\boldsymbol{x}/\tau$

def ApprSolve$(G, \boldsymbol{h}, \xi)$:
  $L \leftarrow 2N + K + 3$
  $\delta \leftarrow 0.1$
  $\varepsilon \leftarrow 0.9\xi$
  $k \leftarrow 57.5 L \ln(6L/\delta)/(\varepsilon^2(1 - \varepsilon^2/4))$
  Run tomography as described in section IV D using k applications and k controlled-applications of the QLSS algorithm on the system $(G, \boldsymbol{h})$
  return *Vector* $\tilde{\boldsymbol{v}}'$ *for which* $\|\tilde{\boldsymbol{v}}'\| = 1$ *and* $\|\tilde{\boldsymbol{v}}' - \boldsymbol{v}\| \leq \xi$ *with probability at least* $1 - \delta$, *where* $\boldsymbol{v} \propto G^{-1}\boldsymbol{h}$

Algorithm 1. Quantum interior-point method.

which is performed many times over the course of the whole algorithm). The resource quantities are reported in terms of the number of calls Q to the block-encoding (which scales linearly with the condition number), as well as the controlled-block-encoding and state-preparation resources given previously in Tables III and IV. The expressions also depend on various error parameters that must be specified to obtain a concrete numerical value. In Sec. VI, after observing empirical scaling of certain algorithmic parameters, we make choices for all error parameters and arrive at a concrete number for a specific problem size.

2. Resource estimate for producing classical approximation to linear-system solution

The resource estimates described above capture the quantum resources required for a single coherent quantum circuit that appears during the algorithm. The output of this quantum circuit is a quantum state but the QIPM requires a classical estimate of the amplitudes of this quantum state. This classical estimate is produced through tomography, as described in Sec. IV D, by performing $k = 57.5 L \ln(6L/\delta)/(\varepsilon^2(1 - \varepsilon^2/4))$ repetitions each of the QLSS and controlled-QLSS circuits, where ε is the desired tomography precision and δ is the probability that the

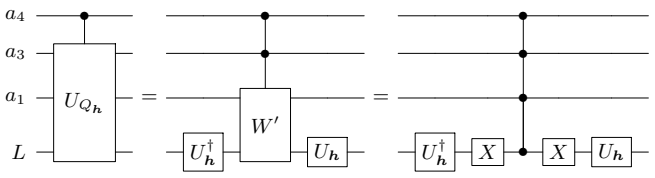

FIG. 4. The decomposition of the $U_{Q_\mathbf{h}}$ gate (shown, e.g., in Fig. 1) into a state-preparation unitary $U_\mathbf{h}$ and multicontrolled-Toffoli gates. The reflection operator W is given by $W' := I_{a_1 L} - 2\,|1\rangle\langle 1|_{a_1} \otimes |0\rangle\langle 0|_L$. Not pictured are additional ancillas that begin and end in $|0\rangle$ and are utilized to implement the unitary $U_\mathbf{h}$ in shallower depth.

tomography succeeds. In the implementation given in Algorithm 1, we fix $\delta = 0.1$. Thus, to estimate the quantum resources of a single *iteration* of the QIPM, the previous resource estimates reported in Table V should each be multiplied by k. We note that with P processors large enough to prepare the output of the QLSS, these k copies could be prepared in k/P parallel steps, saving a factor of P in the runtime at the expense of a factor-of-P additional space. Our resources and scaling estimates do not account for any parallelization and we assume completely serial execution and runtime.

After multiplication by k, these expressions give the quantum resources required to perform the single quantum line of the QIPM, `ApprSolve`. This subroutine has both classical input and output and can thus be compared to classical approaches for approximately solving linear systems.

3. Estimate for end-to-end portfolio-optimization problem

Recall that the full QIPM algorithm is an iterative algorithm, where each iteration involves approximately solving a linear system by preparing many copies of the same quantum states. The duality gap μ, which measures the proximity of the current interior point to the optimal point, begins at 1 and decreases by a constant factor σ with each iteration. Thus, the required number of iterations to reach a final duality gap ϵ is given by

$$N_{it} = \lceil \ln(\epsilon)/\ln(\sigma) \rceil = \left\lceil \frac{\ln(\epsilon)}{\ln(1 - \frac{1}{20\sqrt{2r}})} \right\rceil \approx \left\lceil 20\sqrt{2}\ln(\epsilon^{-1})\sqrt{r} \right\rceil. \tag{65}$$

Recall from the discussion in Sec. III C 3 that the output of the QIPM will achieve an $\mathcal{O}(\epsilon)$ approximation to the optimal value of the objective function.

Pulling this all together, we now estimate the resources to perform the full QIPM algorithm, including the multiplicative factors needed to perform tomography as well as the number of iterations to converge to the optimal solution. Note that the relevant condition number $\kappa_F(G)$ and required linear-system precision ξ will vary from iteration to iteration as the Newton matrix G changes. The overall runtime can be upper bounded using the maximum observed value of $\kappa_F(G)$, which we denote by κ_F, and the minimum observed value of ξ across all iterations. At each iteration, to achieve overall precision ξ, the tomography precision ε is chosen to be just smaller than ξ (we choose $\varepsilon = 0.9\xi$), while all other error parameters (ε_{ar}, ε_{tsp}, ε_z, etc.) are chosen to be small constant fractions of ξ, such that a total error budget of ξ is not exceeded. As the nontomographic error parameters all appear underneath logarithms, these small constant factors will drop out of a leading-order analysis and it suffices to replace all of these error parameters with ξ.

We may then express the overall runtime in terms of κ_F, ξ, L (the size of the Newton system), and r (the number of second-order cone constraints) up to leading order and including all constant factors, which we report in Table VI. Recall that for the infeasible version of the QIPM acting on the self-dual embedding, we have $L = 2N + K + 3$, where N is the number of SOCP variables and K is the number of linear constraints. Note that in our leading-order expression, we have assumed that the contributions proportional to $Q = \mathcal{O}(\kappa_F)$ dominate over terms proportional to $d = \mathcal{O}(\kappa_F \log(1/\xi))$ at practical choices of

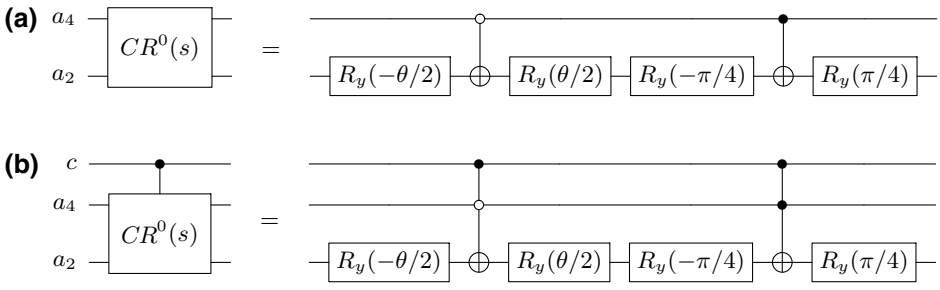

FIG. 5. The decomposition of the (a) $CR^0(s)$ gate and (b) controlled-$CR^0(s)$ gate, as defined in Eq. (55), into single-qubit rotation gates and CNOTs (a) or Toffolis (b). The gate $R_y(\phi)$ is defined to map $|0\rangle \mapsto \cos(\phi/2)\,|0\rangle + \sin(\phi/2)\,|1\rangle$ and $|1\rangle \mapsto -\sin(\phi/2)\,|0\rangle + \cos(\phi/2)\,|1\rangle$. The rotation angle $\theta = 2\arctan(1 - f(s)/f(s))$, where $f(s)$ given in Eq. (59). The $CR^1(s)$ gate is identical but with the control-bit sign flipped. Note that the $R_y(\pm\pi/4)$ gates are Clifford conjugate to a single T or $T^\dagger$ gate.

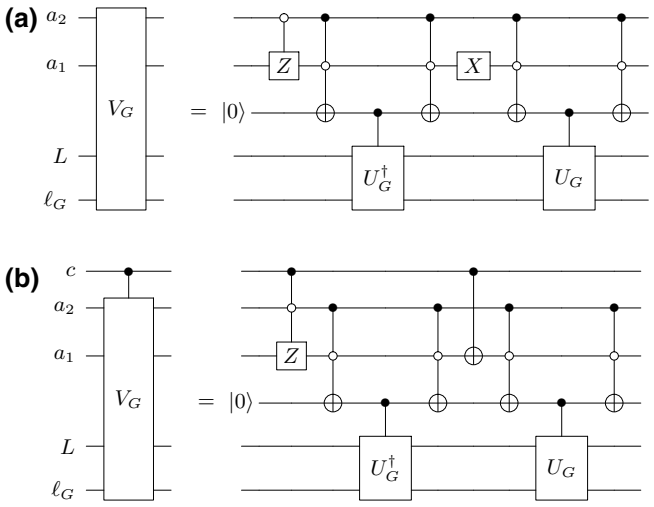

FIG. 6. The decomposition of the (a) V_G unitary and (b) controlled-V_G unitary (bottom), as defined in Eq. (57), into calls to a standard block-encoding unitary U_G [33] and other elementary gates, using a single ancilla qubit initialized to the $|0\rangle$ state. Not pictured are additional ancillas that begin and end in $|0\rangle$ and are utilized to implement the unitary U_G in shallower depth.

ξ due to the large constant prefactor in the definition of Q (see Proposition 2 and the surrounding discussion). The left column of Table I from the introduction is formed using the expressions in Table VI and substituting the corresponding relations between L and n, where n is the number of stocks in the PO problem given in Eq. (10). That is, we substitute $r = 3n + 1$ and $L = 2N + K + 3 = 8n + 3m + 6 = 14n + 6$ when we take $m = 2n$, where N is the number of SOCP variables, K is the number of SOCP constraints, n is the number of stocks, and m is the number of time epochs used to create the matrix M as described in Sec. II.

VI. NUMERICAL EXPERIMENTS WITH HISTORICAL STOCK DATA

The resource expressions in Table VI include constant factors but leave parameters κ_F and ξ unspecified. These parameters depend on the specific SOCP being solved. As a final step, we use numerical simulations of small PO problems to study the size of these parameters for different PO problem sizes. This information enables us to give concrete estimates for the resources needed to solve realistic PO problems with our implementation of the QIPM and sheds light on whether there could be an asymptotic quantum advantage.

Our numerical experiments simulate the entirety of Algorithm 1. The only quantum part of the algorithm is to carry out the subroutine `ApprSolve` $(G, \mathbf{h}, \xi)$. We simulate the quantum algorithm for this subroutine by solving the linear system exactly using a classical solver and then adding noise to the resulting estimated values to simulate the output of tomography. Since the tomography scheme illustrated in Sec. IV D repeatedly prepares the same state and draws k samples from measurements in the computational basis, the result is a sample from the multinomial distribution. In our numerical simulation, we draw samples from this same multinomial distribution, thus capturing tomographic noise in a more precise way than by simply adding uniform Gaussian noise, as was done in Ref. [22]. For simplicity, we assume that the part of the tomography protocol that calculates the signs of each amplitude correctly computes each sign. To numerically estimate resource counts, we must understand ultimately what level of precision ξ is required to stay close enough to the central

TABLE V. The quantum resources required to create the state output by the QLSS, given in Eq. (36) (QLSS, left) or the state needed to compute the signs during the tomography subroutine, given in Eq. (48) (controlled QLSS, right) for a square linear system of size $L = 2^\ell$. Note that these resource quantities do not yet account for the k classical repetitions needed in order to perform tomography on the output state. The parameters Q and d each scale linearly with the condition number of the linear system, as defined in Proposition 2. The symbols N_{Qcbe}, T_{Dcbe}, and T_{Ccbe} denote the number of logical qubits, the T-depth, and the T-count, respectively, for performing a *controlled* block-encoding, as reported in Table III. The symbols T_{Dsp} and T_{Csp} are analogous quantities for state preparation, as reported in Table IV. The parameters $\varepsilon_{\mathrm{ar}}$, $\varepsilon_{\mathrm{tsp}}$, and $\varepsilon_z \in (0, 1]$ are error parameters corresponding to the gate-synthesis precision required for the $CR^0(s)$ and $CR^1(s)$ rotations, the controlled-state-preparation step required by tomography, and the QSVT phases, respectively.

Resource	QLSS	Controlled QLSS
Number of qubits	$N_{\mathrm{Qcbe}} + 5$	$N_{\mathrm{Qcbe}} + 6$
T-depth	$12Q\log_2(1/\varepsilon_{\mathrm{ar}}) + 2(Q+d)T_{\mathrm{Dcbe}} + 4(Q+d)T_{\mathrm{Dsp}}$ $+Q(24\ell+31) + 3d\log_2(1/\varepsilon_z) + d(32\ell-2)$	$12Q\log_2(1/\varepsilon_{\mathrm{ar}}) + 2(Q+d)T_{\mathrm{Dcbe}} + 4(Q+d)T_{\mathrm{Dsp}}$ $+Q(24\ell+36) + 6d\log_2(1/\varepsilon_z) + d(32\ell-2)$ $+12\log_2(1/\varepsilon_{\mathrm{tsp}}) + 3(\ell-1)$
T-count	$12Q\log_2(1/\varepsilon_{\mathrm{ar}}) + 2(Q+d)T_{\mathrm{Ccbe}} + 4(Q+d)T_{\mathrm{Csp}}$ $+Q(24\ell+31) + 3d\log_2(1/\varepsilon_z) + d(32\ell-2)$	$12Q\log_2(1/\varepsilon_{\mathrm{ar}}) + 2(Q+d)T_{\mathrm{Ccbe}} + 4(Q+d)T_{\mathrm{Csp}}$ $+Q(24\ell+51) + 6d\log_2(1/\varepsilon_z) + d(32\ell-2)$ $+12(L-1)\log_2(1/\varepsilon_{\mathrm{tsp}}) + 16(L-\ell-1)$

TABLE VI. The leading-order contribution to the logical qubit count, T-depth, and T-count for the entire QIPM, including constant factors. The parameter L denotes the size of the Newton linear system and r denotes the number of second-order cone constraints, while ϵ denotes the final duality gap that determines when the algorithm is terminated. For the infeasible QIPM running on an n-asset instance of PO, as given in Eq. (10), we have $L = 14n + 6$ and $r = 3n + 1$; these substitutions yield the results in Table I. The parameter κ_F denotes the maximum observed Frobenius condition number and ξ denotes the minimum observed tomographic precision parameter across all iterations.

Resource	QIPM complexity
Number of qubits	$4L^2$
T-depth	$(5 \times 10^8)\frac{\kappa_F L\sqrt{r}}{\xi^2}\log_2\left(\frac{1}{\epsilon}\right)\log_2(L)\log_2\left(\frac{\kappa_F L^{14/27}}{\xi}\right)$
T-count	$(1 \times 10^8)\frac{\kappa_F L^3\sqrt{r}}{\xi^2}\log_2\left(\frac{1}{\epsilon}\right)\log_2(L)\log_2\left(\frac{\kappa_F}{\xi}\right)$

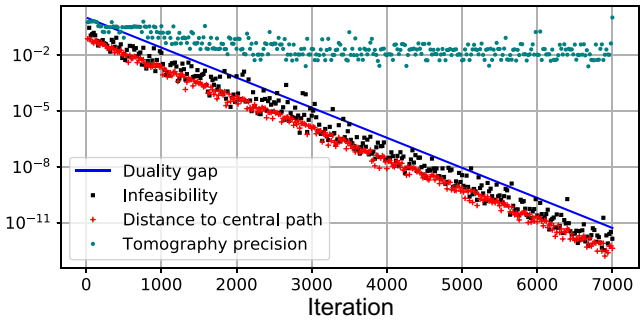

FIG. 7. The simulation of the QIPM on an SOCP instance corresponding to PO on $n = 30$ randomly chosen stocks using $m = 60$ time epochs. The duality gap μ [defined in Eq. (14)], the distance to the central path d_F [defined in Eq. (26)], and the infeasibility [defined as the norm of the residual on the right-hand side in Eq. (19)] each decrease exponentially with the number of iterations. The tomography precision ξ required to stay near the central path (defined adaptively as outlined in Algorithm 1) initially decreases and then plateaus at about 10^{-2}.

path throughout the algorithm, as well as how large the Frobenius condition number κ_F of the Newton system is. Importantly, we would like to know how these quantities scale with the system size and the duality gap μ, which decreases by a constant factor with each iteration of the QIPM.

In Sec. III C 5, we have discussed three formulations of the QIPM (see Table II). The first (II-QIPM) is closely related to the original formulation from Ref. [13], which does not guarantee that the intermediate points generated by the IPM are feasible. The other two are instantiations of the inexact-feasible formulation proposed in Ref. [14], which requires precomputing a basis for the null space of the SOCP constraint matrix. The first of these computes a valid basis by hand (IF-QIPM), while the second uses a QR decomposition to find the basis (IF-QIPM-QR). We have simulated all three versions and have found that the II-QIPM was always able to stay close to the central path, despite the lack of a theoretical guarantee that this would be the case. Here, we present the results of the II-QIPM. For comparison, in Appendix E, we present some numerical results for the feasible QIPMs, which do benefit from a theoretical convergence guarantee but have other drawbacks.

As discussed in Sec. V A, we have also implemented a very simple preconditioner that we find reduces the condition number by at least an order of magnitude with negligible additional classical cost. In all cases, we report resources estimates assuming a preconditioned matrix.

A. Example instance

In Fig. 7, we present as an example the results of one of our simulations. We construct a PO instance of Eq. (3) by randomly choosing $n = 30$ stocks from the Dow Jones U.S. Total Stock Market Index (DWCF). We (arbitrarily) set parameters $q = 1$ and $\boldsymbol{\zeta} = 0.05 \cdot \mathbf{1}$ and we assume that our previous portfolio $\bar{\mathbf{w}}$ allocates weight to each stock in proportion to its market capitalization. The returns of the 30 stocks on the first $m = 2n = 60$ days in our data set have been used to construct an average return vector $\hat{\mathbf{u}}$ and an $m \times n$ matrix M for which $M^\intercal M = \Sigma$, the covariance matrix for the stock returns, as described in Sec. III B.

We simulate the infeasible QIPM acting on the corresponding SOCP in Eq. (10). The figure illustrates how the simulation successfully follows the central path to the optimal solution after many iterations. The duality gap decreases with each step and, crucially, the infeasibility and distance to the central path also decrease (exponentially) with iteration number. Also plotted is the tomography precision ξ that was required to ensure that each iteration stayed sufficiently close to the central path (determined adaptively as described in the pseudocode in Algorithm 1). The plot exemplifies how, despite the lack of theoretical convergence guarantees, our simulations suggest that in practice the II-QIPM acting on the PO SOCP will yield valid solutions.

Remarkably, for this instance, we also observe that both the Frobenius condition number κ_F and the inverse-tomography precision ξ^{-1} initially increase but ultimately plateau with the iteration number, even as the duality gap gets arbitrarily small (see Fig. 8 for data on κ_F). This scaling behavior was a generic feature of our simulations across all the instances that we simulated. This contrasts with the worst-case expectation that the condition number can increase as $\kappa_F = \mathcal{O}(1/\mu)$ or $\kappa_F = \mathcal{O}(1/\mu^2)$ (depending on the formulation of the Newton system) [13,14]. The prior literature does not say much about whether the

quantity ξ^{-1} should be expected to diverge. One might expect that since the neighborhood of the central path gets smaller as μ gets smaller [e.g., the radius is proportional to μ in Eq. (27)], the precision requirement to stay close to the central path would get more stringent in proportion to μ. However, it is important to recall that the step size from one iteration to the next also shrinks with μ and that ξ represents the size of the error on the *normalized* Newton system solution; thus the neighborhood does not shrink *relative* to the distance to the optimum and the length of the next step and there is no immediate reason that ξ^{-1}, as we have defined it, must diverge as $\mu \to 0$. However, one does expect that in the worst case, if the condition number κ diverges, then ξ^{-1} should also diverge, as errors of constant size ξ on the estimate of $\mathbf{u}/\|\mathbf{u}\|$ can lead to residual errors of divergent size $\kappa\xi$ on the normalized product $G\mathbf{u}/\|G\mathbf{u}\|$. We hope that future work can better elucidate why κ_F and ξ^{-1} do not diverge on these instances [78].

B. Scaling of condition number

To understand the problem scaling with the portfolio size, we generate example problem instances by randomly sampling n stocks from the DWCF, using returns over $m = 2n$ time epochs (days) to construct our SOCP as in Eq. (10). Parameters q, $\boldsymbol{\zeta}$, $\bar{\mathbf{w}}$, $\hat{\mathbf{u}}$, and M are all chosen in the same way as described above. We plot the Frobenius condition number of the Newton matrix as well as the preconditioned Newton matrix as a function of the duality gap in Fig. 8 for portfolios of size $n \in \{60, 80, 100, 120\}$. Here, we confirm our previous remark that the condition number appears to plateau at a certain value of the duality gap, especially for the preconditioned matrix.

Key to understanding the asymptotic scaling of the quantum algorithm is to determine how the condition number scales as a function of the number of assets, as the runtime of the QLSS algorithm grows linearly with the condition number. In Fig. 9, we plot the Frobenius condition number κ_F as a function of n, the number of stocks, observed at duality gaps $\mu \in \{10^{-1}, 10^{-3}, 10^{-5}, 10^{-7}\}$. At duality gaps of 10^{-5} and 10^{-7}, the condition number κ_F has plateaued as observed in Fig. 8. We perform a nonlinear fit to the data using a power-law $\kappa_F = an^b$ model, where a and b are fit parameters, and we report the exponents b in Table VII. All exponents appear to be near or less than unity.

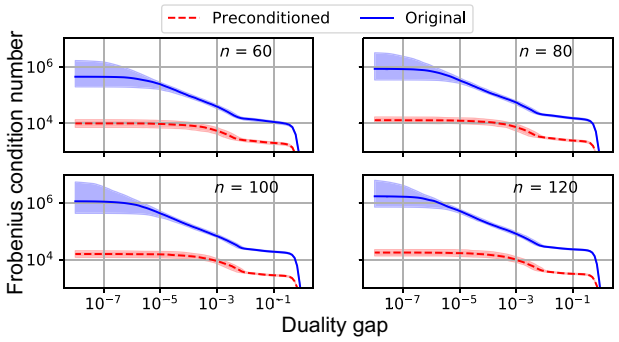

FIG. 8. The median Frobenius condition κ_F number for 128 randomly sampled stock portfolios from the DWCF index as a function of the duality gap for portfolios of size 60, 80, 100, and 120 stocks. The shaded regions indicate the 16th to 84th percentiles. We observe that the condition number appears to plateau at small values of the duality gap.

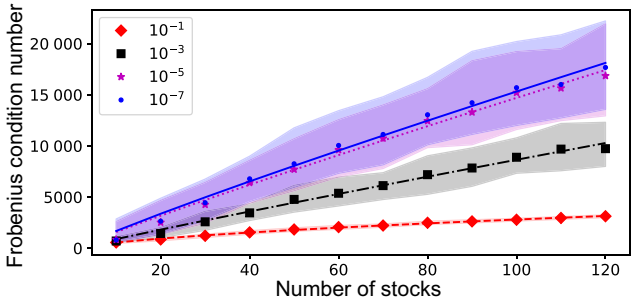

FIG. 9. The median Frobenius condition number κ_F for 128 randomly sampled stock portfolios from the DWCF index as a function of the portfolio size for duality gaps of 10^{-1}, 10^{-3}, 10^{-5}, and 10^{-7}. The shaded regions correspond to the 16th to 84th percentiles. The lines represent power-law fits of the form an^b, where the values for b are reported in Table VII. In all four cases, the exponent is less than 1 and in the latter three cases it is greater than 0.9, suggesting a trend that is nearly linear in n.

C. Scaling of tomography precision

While the depth of the individual quantum circuits that compose the QIPM scales only with the Frobenius condition number, the QIPM also requires a number of repetitions of this circuit for tomography that scales as $1/\xi^2$, the inverse of the tomography precision squared. To see how this scales with problem size, we have performed an analysis for ξ^{-2} similar to the one we have previously performed for κ_F. These results are presented in Fig. 10 for the same four duality gaps of $\{10^{-1}, 10^{-3}, 10^{-5}, 10^{-7}\}$. To reduce the iteration-to-iteration variation in the tomography precision (which results from our adaptive approach to tomography in Algorithm 1), in calculating ξ^{-2} at duality gap μ, we have taken the average over the value of ξ^{-2} at the five iterations

TABLE VII. The estimated exponent parameters for the Frobenius condition number κ_F obtained from the fits that are plotted in Fig. 9.

Duality gap	Condition-number scaling
10^{-1}	$\mathcal{O}(n^{0.60\pm0.02})$
10^{-3}	$\mathcal{O}(n^{0.94\pm0.04})$
10^{-5}	$\mathcal{O}(n^{0.92\pm0.04})$
10^{-7}	$\mathcal{O}(n^{0.91\pm0.05})$

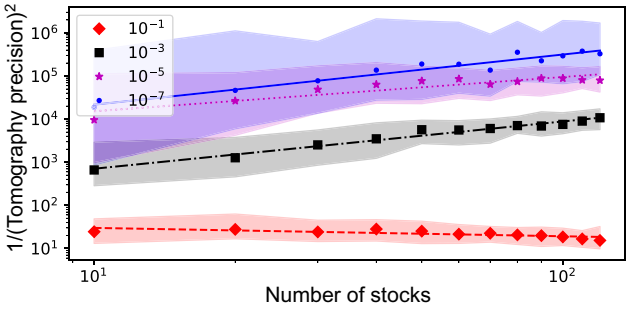

FIG. 10. The median value of the square of the inverse-tomography precision ξ^{-2} required to remain in the neighborhood of the central path for 128 randomly sampled stock portfolios from the DWCF index as a function of the portfolio size for duality gaps of 10^{-1}, 10^{-3}, 10^{-5}, and 10^{-7}. To reduce iteration-to-iteration variation, an artifact of the adaptive approach to tomography, we average over the observed value of ξ^{-2} at the five iterations for which the duality gap is nearest to the indicated value. The shaded regions correspond to the 16th to 84th percentiles. Here, logarithmic axes are used, since (unlike for κ_F) instance-to-instance variation covers multiple orders of magnitude even for a fixed value of n. The dashed lines correspond to a linear fit to the log-log data, where the slope is reported in Table VIII.

with a duality gap nearest to μ. We have fitted the median of ξ^{-2} at each value of n to a linear model on a log-log plot, corresponding to a relationship $\xi^{-2} = an^b$, and we report the implied exponent b in Table VIII. In this case, it is hard to draw robust conclusions from the fits. The fit suggests that the median of ξ^{-2} is increasing with n on the interval $n \in [10, 120]$. However, the most striking feature of the data is that the instance-to-instance variation of ξ^{-2} is significantly larger than that of κ_F. In fact, at $\mu = 10^{-7}$, the 84th percentile of instances at $n = 10$, the smallest size we have simulated, has a larger value of ξ^{-2} than the 50th percentile of instances at $n = 120$, the largest size we have simulated.

D. Asymptotic scaling of overall runtime

Above, we have provided fits for κ_F and ξ^{-2} as a function of n on the range $n \in [10, 120]$. Here, we study the quantity $n^{1.5}\kappa_F/\xi^2$, which determines the asymptotic scaling of the runtime of the QIPM. In Fig. 11, we plot this quantity at the same four duality-gap values $\mu \in \{10^{-1}, 10^{-3}, 10^{-5}, 10^{-7}\}$. The implied exponents arising from linear fits on a log-log axis are reported in Table IX. They are generally consistent with summing the exponents from the previously reported fits. The data inherit from ξ^{-2} the feature that the instance-to-instance variation is orders of magnitude larger than the median. Taken at face value, the fits suggest that the scaling of the median algorithmic runtime on the interval $n \in [10, 120]$ is similar to the $n^{3.5}$ scaling of classical IPMs using Gaussian elimination and worse than the asymptotic $n^{2.87}$ arising from classical IPMs using fast matrix-multiplication techniques to solve linear systems [54,55] (note that this scaling does not apply until n becomes very large, so it is not a good practical comparator). However, the large variance and imperfect fits do not give us confidence that these trends can be reliably extrapolated to larger n. Accordingly, when we compute actual resource counts in Sec. VI E, we stick to $n = 100$ and do not speculate on precise estimates for larger (more industrially relevant) n. Our numerical experiments fail to provide significant evidence for an asymptotic polynomial quantum speed-up but nor do they definitively rule it out. Toward that end, note that if the version of tomography we have studied were to be replaced with the more advanced recently proposed tomography scheme of Ref. [73], the runtime of the QIPM would instead grow as $n^{1.5}\kappa_F/\xi$, while introducing some additional gate overhead. Our fits from Table VIII suggest that this could reduce the asymptotic exponent but by no more than about $\mathcal{O}(n^{0.6})$ or so.

Ultimately, we do not believe it is essential to pin down the asymptotic scaling of the algorithm, because the main finding of our work is that even if a slight asymptotic

TABLE VIII. The estimated exponent parameters for $1/\xi^2$ obtained from the fits that are plotted in Fig. 10.

Duality gap	Tomography scaling
10^{-1}	$\mathcal{O}(n^{-0.19\pm0.05})$
10^{-3}	$\mathcal{O}(n^{1.10\pm0.06})$
10^{-5}	$\mathcal{O}(n^{0.79\pm0.11})$
10^{-7}	$\mathcal{O}(n^{1.16\pm0.10})$

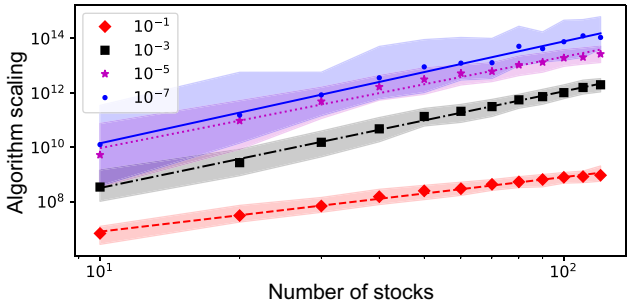

FIG. 11. The median value of the estimated algorithm scaling factor computed as the median of $n^{1.5}\kappa_F/\xi^2$ for 128 randomly sampled stock portfolios from the DWCF index as a function of the portfolio size for duality gaps of 10^{-1}, 10^{-3}, 10^{-5}, and 10^{-7}. As in Fig. 10, we average over five consecutive points to reduce iteration-to-iteration variance deriving from adaptive tomography. Here, we also use the actual number of observed samples that were required to achieve sufficient tomographic precision in place of the tomographic factor n/ξ^2. The shaded regions correspond to the 16th to 84th percentiles. The lines correspond to a linear fit to the log-log data, where the slope is reported in Table IX.

TABLE IX. The exponent parameter estimates from the fits to the line generated by plotting $n^{1.5}\kappa_F/\xi^2$ in Fig. 11, which determines the overall scaling of the runtime of the QIPM. For comparison, CIPMs using Gaussian elimination have runtime $\mathcal{O}(n^{3.5})$ and CIPMs using faster methods for solving linear systems have runtime $\mathcal{O}(n^{2.87})$.

Duality gap	Algorithm scaling
10^{-1}	$\mathcal{O}(n^{2.01\pm0.05})$
10^{-3}	$\mathcal{O}(n^{3.56\pm0.07})$
10^{-5}	$\mathcal{O}(n^{3.36\pm0.14})$
10^{-7}	$\mathcal{O}(n^{3.75\pm0.12})$

polynomial speed-up exists, the size of the constant prefactors involved in the algorithm precludes an actual practical speed-up, barring significant improvements to multiple aspects of the algorithm. In Sec. VI E, we elaborate on this point in a more quantitative fashion.

E. Numerical-resource estimates

Rather than examine algorithmic scaling, we now compute actual resource counts for the QIPM applied to PO. Ultimately, it is these resource counts that matter most from a practical perspective. We estimate the total circuit size in terms of the number of qubits, T-depth, and T-count for a portfolio of 100 assets. We have chosen this size because it is small enough that we can simulate the entire quantum algorithm classically. However, at this size, solving the PO problem is not classically hard; generally speaking, the PO problem becomes challenging to solve with classical methods only once n is on the order of 10^3–10^4. A similar concrete calculation could be performed at larger n by extrapolating trends observed in our numerical simulations but we are not confident that the fits on $n \in [10, 120]$ reported above are reliable predictors for larger n.

Recall that the only step in the QIPM performed by a quantum computer is the task of producing a classical estimate to the solution of a linear system to error ξ. The complexity of this task as it is performed within the QIPM depends on ξ as well as the Frobenius condition number κ_F. The first step of our calculation is to fix values for ξ and κ_F at $n = 100$. We choose them by taking the median over the 128 samples in our numerical simulation at duality gap $\mu = 10^{-7}$.

Once κ_F and ξ are fixed, we must now determine concrete values for the various other error parameters that appear in the algorithm such that overall error ξ can be achieved. Tomography dominates the complexity and overall error but there are a number of other factors that contribute to the error in the final solution. We enumerate and label the sources of error here, for completeness:

(a) ε_G: error in block-encoding the matrix G
(b) $\varepsilon_{\mathbf{h}}$: error in the unitary that prepares the state $|\mathbf{h}\rangle$
(c) $\varepsilon_{\mathrm{ar}}$: gate-synthesis error for single-qubit rotations needed by $CR^0(s)$ and $CR^1(s)$ (see Fig. 5)
(d) ε: tomography error
(e) ε_z: gate-synthesis error for each single-qubit rotation needed for QSVT eigenstate filtering (see Fig. 2)
(f) $\varepsilon_{\mathrm{qsp}}$: error due to polynomial approximation in eigenstate filtering
(g) $\varepsilon_{\mathrm{tsp}}$: error in preparing the state $\sum_{i=1}^{L}\sqrt{p_i}|i\rangle$ needed for computing the signs in the tomography routine

In Sec. IV, we have described a quantum circuit that prepares a state $|\tilde{\mathbf{v}}\rangle$ (after postselection) for which $\||\tilde{\mathbf{v}}\rangle - |\mathbf{v}\rangle\| \leq \varepsilon_{\mathrm{QLSP}}$. If the block-encoding unitaries, state-preparation unitaries, and single-qubit rotations were perfect, then the only contribution to $\varepsilon_{\mathrm{QLSP}}$ would be from eigenstate filtering and we would have $\varepsilon_{\mathrm{QLSP}} \leq \varepsilon_{\mathrm{qsp}}$. Note the relationship $d = 2\kappa_F \ln(2/\varepsilon_{\mathrm{qsp}})$ from Proposition 2. Since the block-encoding unitary U_G, the state-preparation unitary $U_{\mathbf{h}}$, and the single-qubit rotations are implemented imperfectly, there is additional error. In preparing the state, the unitary U_G is called $2Q + 2d$ times and the unitary $U_{\mathbf{h}}$ is called $4Q + 4d$ times, where Q is given in Proposition 2. Additionally, there are $2Q$ combined appearances of $CR^0(s)$ and $CR^1(s)$ gates, where each appearance requires two single-qubit rotations. Note that the appearances of $CR^0(s)$ and $CR^1(s)$ within the eigenstate-filtering portion of the circuit do not contribute to the error, because at $s = 1$ these gates can be implemented exactly. Finally, there are another d single-qubit rotations required to implement the eigenstate-filtering step. Since operator norm errors add sublinearly, we can thus say that

$$\varepsilon_{\mathrm{QLSP}} \leq \varepsilon_{\mathrm{qsp}} + (2Q + 2d)\varepsilon_G + (4Q + 4d)\varepsilon_{\mathbf{h}} + 4Q\varepsilon_{\mathrm{ar}} + 2d\varepsilon_z. \tag{66}$$

Now, the result of Proposition 4 implies that in order to assert that the classical estimate $\tilde{\mathbf{v}}'$ output by tomography satisfies $\|\tilde{\mathbf{v}}' - \mathbf{v}\| \leq \xi$, it suffices to have

$$\xi \geq \varepsilon + 1.58\sqrt{L}\varepsilon_{\mathrm{tsp}} + 1.58\Big[\varepsilon_{\mathrm{qsp}} + (2Q + 2d)\varepsilon_G + (4Q + 4d)\varepsilon_{\mathbf{h}} + 4Q\varepsilon_{\mathrm{ar}} + d\varepsilon_z\Big], \tag{67}$$

where, for convenience, we recall the definitions (ignoring the $\mathcal{O}(\sqrt{\kappa_F})$ term) of Q and d as

$$Q = 1.31C\kappa_F, \tag{68}$$

$$d = 2\kappa_F \ln(2/\varepsilon_{\mathrm{qsp}}). \tag{69}$$

Recalling that the dominant term in the complexity of the algorithm scales as ε^{-2} but logarithmically in the other

TABLE X. The estimated number of logical qubits N_Q, the T-depth T_D, and the T-count T_C required to perform the QLSS subroutine within the QIPM running on a PO instance with $n = 100$ stocks. This calculation uses the empirically observed median value for the condition number at duality gap $\mu = 10^{-7}$, which was $\kappa_F = 1.6 \times 10^4$. The full QIPM repeats this circuit $k = \mathcal{O}(n \ln(n)\xi^{-2})$ times in each iteration to generate a classical estimate of the output of the QLSS and also performs $N_{it} = \mathcal{O}(n^{0.5})$ iterations, where the linear system being solved changes from iteration to iteration. In the left column, we write the resources as numerical prefactors times the resources required to perform the controlled block-encoding of the matrix G (denoted by a subscript "cbe") and the state preparation of the vector $|\mathbf{h}\rangle$ (denoted by a subscript "sp"), defined in Tables III and IV. Written in this way, one can see the large prefactors occurring from the linear-system-solver portion of the algorithm. In the right column, we compute the exact resources, including those coming from the block-encoding.

QLSS prefactors	Total
$N_Q = N_{\text{Qcbe}} + 5$	$N_Q = 8 \times 10^6$
$T_D = (1 \times 10^8)T_{\text{Dcbe}} + (2 \times 10^8)T_{\text{Dsp}} + (4 \times 10^{10})$	$T_D = 3 \times 10^{11}$
$T_C = (1 \times 10^8)T_{\text{Ccbe}} + (2 \times 10^8)T_{\text{Csp}} + (4 \times 10^{10})$	$T_C = 1 \times 10^{17}$

error parameters, to minimize the complexity we assign the majority of the error budget to ε: we let $\varepsilon = 0.9\xi$ and we split the remaining 0.1ξ across the remaining six terms of Eq. (67). There is room for optimizing this error-budget allocation but the savings would be at most a small constant factor in the overall complexity.

Note that elsewhere in the paper, we have referred to ξ as "tomography precision," since ε will dominate the contribution to ξ. Here, the resource calculation requires that we differentiate ε from ξ but when speaking conceptually about the algorithm, we focus on ξ, as it is the more fundamental parameter: it represents the precision at which the classical-input–classical-output linear-system problem is solved, allowing apples-to-apples comparisons between classical and quantum approaches.

With values for κ_F, ε_G, $\varepsilon_{\mathbf{h}}$, ε_{qsp}, ε_z, and ε_{tsp} now fixed, we can proceed to complete the resource count using the expressions in Table V. Note that for gate-synthesis error, we use the formula $R_y = 3\log_2(1/\varepsilon_r)$, where R_y is the number of T gates needed to achieve an ε_r-precise Clifford-plus-T-gate decomposition of the rotation gate [72]. Putting this all together yields the resource estimates for a single run of the (uncontrolled) QLSS in Table X, at $n = 100$. We report these estimates both in terms of primitive block-encoding and state-preparation resources, as well as the raw numerical estimates. For the total runtime, we must also estimate the resources required for the controlled-state-preparation routine. We have estimated these quantities but to the precision of the estimates that we report, the numbers are the same as for the controlled version, so we exclude them for brevity.

To estimate the total runtime, our estimates must be multiplied by the tomography factor k (for controlled and for uncontrolled) as well as the number of iterations $N_{it} = \lceil \ln(\epsilon)/\ln(\sigma) \rceil$, where ϵ is the target duality gap (which we take to be $\epsilon = 10^{-7}$) and $\sigma = 1.0 - 1/(20\sqrt{2r})$. While k will vary from iteration to iteration, in our calculation we assume that the total number of repetitions is given by the simple product $(2k)N_{it}$, which, noting that the value of ξ plateaus after a certain number of iterations, will give a roughly accurate estimate. Note that these $2kN_{it}$ repetitions need not be done coherently, in the sense that the entire system is measured and reprepared in between each repetition. One can bound the tomography factor k to be $k \leq 57.5L\ln(L)/\xi^2$, where ξ is determined empirically. However, our numerical simulations of the algorithm yield an associated value of k needed to generate the estimate to precision ξ, so we can use this numerically determined value directly. We find that the observed median value of $k = 3.3 \times 10^8$ from simulation is multiple orders of magnitude smaller than the theoretical bound. Using this substitution for k and N_{it}, we find the results shown in the right column of Table I in Sec. I A.

To aid in understanding which portions of the algorithm dominate the complexity, we show a breakdown of the resources in Fig. 12. The width of the boxes is representative of the T-depth, while the height of the boxes represents the T-count. The number of classical repetitions, composed of tomography samples as well as IPM iterations needed to reach a target duality gap, contributes the largest factor to the algorithmic runtime. Of these two, quantum state tomography contributes more than the iterations needed to reach the target duality gap. Our exact calculation confirms that for the individual quantum circuits involved in the QLSS, the discrete adiabatic portion of the algorithm dominates over the eigenstate-filtering step in its contribution to the overall quantum circuit T-depth. Within the adiabatic subroutine, the primary driver of the T-depth and T-count is the need to apply the block-encoding operator Q times [see, e.g., Eq. (61)], where Q is proportional to the Frobenius condition number. An additional source of a large T-count arises from the need to block-encode the linear system, which causes the T-count to scale as $\mathcal{O}(L^2)$.

VII. CONCLUSIONS

A. Bottlenecks

The resource quantities that we report are prohibitively large, even for the classically easy problem size of $n = 100$ assets in the PO instance. Our detailed analysis allows us to see exactly how this large number arises, which is essential for understanding how best to improve it. We outline the

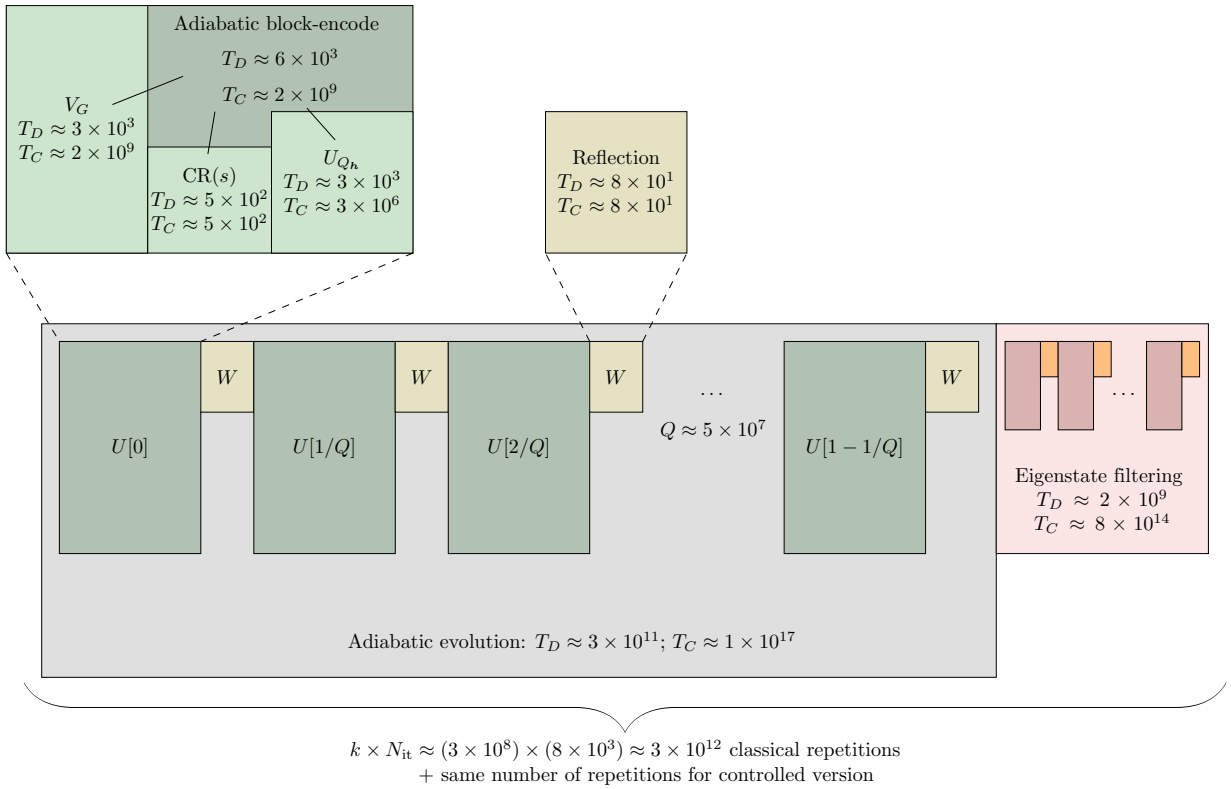

FIG. 12. The breakdown of the quantum resources required for a single coherent run of the uncontrolled version of the quantum algorithm needed to produce the state given in Eq. (36). As we did in Table X, here we take the final duality gap to be $\mu = 10^{-7}$ and the number of assets to be $n = 100$. Our choices for the Frobenius condition number, $\kappa_F = 1.6 \times 10^4$, and the number of tomographic repetitions, $k = 3.3 \times 10^8$, are informed by our numerical experiments, as discussed in Sec. VI. A similar breakdown for the controlled version needed to produce the state given in Eq. (48) would be essentially the same. The eigenstate-filtering subcircuit follows a very similar alternating structure to the adiabatic evolution, with the $U[j]$ block-encodings replaced with either $U[1]$ or $U[1]^\dagger$, the reflection operator W replaced with phase rotations, and only a $d \ll Q$ total number of iterations (for details, see Fig. 2).

several independent factors leading to the large resource estimates:

(a) The block-encoding of the classical data is called many times by the QLSS. These data are arranged in an $L \times L$ matrix (note that for a PO instance of size n with $m = 2n$, the Newton linear system has size roughly $L \approx 14n$). These block-encodings can be implemented up to error ε_G in $\mathcal{O}(\log(L/\varepsilon_G))$ T-depth using circuits for QRAM as a subroutine [33]. While the asymptotic scaling is favorable, after close examination of the circuits for block-encoding, we find that in practice the T-depth can be quite large: at $n = 100$ and $\varepsilon_G = 10^{-10}$ (it is necessary to take ε_G very small since the condition number of G is quite large), block-encoding to precision ε_G has a T-depth of nearly 1000. Importantly, this T-depth arises *even after implementing several new ideas to minimize the circuit depth*, presented by a subset of the authors separately in Ref. [33].

(b) The condition number κ_F determines how many calls to the block-encoding must be made and we observe that κ_F is quite large for the application of PO. *Even after an attempt at preconditioning, κ_F is already on the order of 10^4 for small SOCP instances corresponding to $n = 100$ stocks* and empirical trends suggest that it grows nearly linearly with n. However, we believe that additional preconditioning could significantly reduce the effective value of κ_F in this algorithm.

(c) The constant factor in front of the $\mathcal{O}(\kappa_F)$ in state-of-the-art QLSSs is also quite large: the theoretical analysis proves an upper bound on the prefactor of 1.2×10^5. Numerical simulations performed in Ref. [18] have suggested that, in practice, it can be one order of magnitude smaller than the theoretical value. Following these numerics, we take the constant prefactor to be 1.31×2305 in our numerical estimates, which still contributes significantly to the estimate. Future work should aim

to reduce this constant or, alternatively, investigate whether other approaches, such as those based on variable-time amplitude amplification (VTAA) [60,79], could achieve better performance despite being asymptotically suboptimal [80].

(d) Pure-state tomography requires preparing many copies of the output $|\mathbf{v}\rangle$ of the QLSS. We have improved the constant prefactors in the theoretical analysis beyond what was known but even with this improvement, the number of queries needed to produce an estimate $\mathbf{v}'$ of the amplitudes of $|\mathbf{v}\rangle$ up to error ε in ℓ_2 norm is $115L\ln(L)/\varepsilon^2$, which for $n = 100$ and $\varepsilon = 10^{-3}$ is on the order of 10^{11} (although our simulations suggest that $2k = 7 \times 10^8$ suffice in practice). We note that this is another avenue for substantial improvement. For instance, the results of Ref. [73] could be used (for more details, see Ref. [34]).

(e) QIPMs, like CIPMs, are iterative algorithms; the number of iterations in our implementation is roughly $20\sqrt{2r}\ln(\epsilon^{-1})$, a number chosen to utilize theoretical guarantees of convergence (note that $r \approx 3n$). Taking $n = 100$ and $\epsilon = 10^{-7}$, our implementation would require 8×10^3 iterations. We suspect that the number of iterations could be significantly decreased if more aggressive choices were made for the step size. For example, similar to our adaptive approach to tomographic precision, one could try longer step sizes first and shorten the step size when the iteration does not succeed. This sort of optimization would apply equally to CIPMs and QIPMs.

Remarkably, the five factors described above all contribute roughly equally to the overall T-depth calculation; the exception being the number of copies needed to do tomography, which is a much larger number than the others. Tomography would be the obvious place to begin to try to reduce the resource depth, perhaps by implementing the scheme recently proposed in Ref. [73] and by making modifications to the QIPM that might allow the parameter ξ to be larger in practice, or by using an iterative refinement method [14]. Another comment regarding tomography is that, in principle, the k tomographic samples can be taken in parallel rather than in series. Running in parallel leads to a huge overhead in memory: one can reduce the tomographic depth by a multiplicative factor P at the cost of a multiplicative factor P additional qubits. Note that even preparing a single copy requires a daunting number of nearly 1×10^7 logical qubits at $n = 100$. Moreover, it is unlikely that improvements to tomography alone could make the algorithm practical, as the other four factors still contribute roughly 10^{16} to the T-depth.

Besides the rather large constant factors pointed out above for tomography, and especially for the QLSS, we also note that the multiplicative "log factors" that are typically hidden underneath $\tilde{\mathcal{O}}$ notation in asymptotic analyses contribute meaningfully here. For instance, the entire block-encoding depth is $\mathcal{O}(\log(n/\varepsilon_G))$, which, in practice, is as large as 1000. Moreover, there is an additional $\ln(\epsilon^{-1}) \approx 16$ coming from the iteration count and a $\ln(L) \approx 7$ from tomography.

This quantitative analysis of bottlenecks for QIPMs can inform likely bottlenecks in other applications where QLSS, tomography, and QRAM subroutines are required. While some parameters such as κ_F and ξ are specific to the application we have considered here, other observations such as the numerical size of various constant and logarithmic factors (e.g., the block-encoding depth) would apply more generally in other situations.

B. Resource estimate given dedicated QRAM hardware

The above bottlenecks have focused mainly on the T-depth and have not taken into account the total T-count or the number of logical qubits, which are also large. Indeed, our estimate of 8×10^6 logical qubits, as reported in Table I, is drastically larger than estimates for other quantum algorithms, such as Shor's algorithm [81] and algorithms for quantum chemistry (see, e.g., Ref. [82]), both of which can be on the order of 10^3 logical qubits. By contrast, the current generation of quantum processors have tens to hundreds of *physical* qubits and no logical qubits; a long way from the resources required for this QIPM.

However, it is important to note that, as for other algorithms requiring repeated access to classical data, the vast majority of the gates and qubits in the QIPM arise in the block-encoding circuits, which are themselves dominated by QRAM-like data-loading subcircuits [33]. These QRAM-like subcircuits have several special features. First, they are largely composed of controlled-SWAP gates, each of which can be decomposed into four T gates that can even be performed in a single layer, given one additional ancilla and classical feed-forward capability [83]. Furthermore, in some cases, the ancilla qubits can be "dirty" [65,67], i.e., initialized to any quantum state, and, if designed correctly, the QRAM circuits can possess a natural noise resilience that may reduce the resources required for error correction [67]. Implementation of these circuits with full-blown universal and fault-tolerant hardware could be unnecessary given their special structure. Just as classical computers have dedicated hardware for RAM, quantum computers may have dedicated hardware optimized for performing the QRAM operation. Preliminary work on hardware-based QRAM data structures (as opposed to QRAM implemented via quantum circuits acting on logical qubits) shows promise in this direction [84,85].

Our estimates suggest that the size of the QRAM needed to solve an $n = 100$ instance of PO is 1 MB and that the QRAM size for $n = 10^4$ (i.e., sufficiently large to potentially be challenging by classical standards) is roughly 10 GB, which is comparable to the size of the classical RAM that one might find on a modern laptop. These numbers could perhaps be reduced by exploiting the structure of the Newton matrix, as certain blocks are repeated multiple times in the matrix and many of the entries are zero [86] [see Eqs. (19) and (10)].

With this in mind, we can ask the following hypothetical question. Suppose that we had access to a sufficiently large dedicated QRAM element in our quantum computer and, furthermore, that the QRAM ran at a 4-GHz clock speed (which is comparable to modern classical RAM). Would the algorithm become more practical in this case? Under the crude conservative simplifying assumption that each block-encoding and state-preparation unitary requires just a *single* call to QRAM and the rest of the gates are free, we can give a rough answer by referring to the expression in Table X, which states that 3×10^8 total block-encoding and state-preparation queries are needed [87]. Thus, even if the rest of our estimates stay the same, the number of QRAM calls involved in just a single QLSS circuit for $n = 100$ would be 3×10^8. Accounting for the fact that the QIPM involves an estimated 6×10^{12} repetitions of similarly sized circuits, the overall number of QRAM calls needed to solve the PO problem would be larger than 10^{21} and the total evaluation time would be on the order of 1×10^4 years. Thus, even at 4-GHz speed for the QRAM, the problem remains decidedly intractable. Nonetheless, we believe that if the QIPM were to be made practical, it would need to involve specialized QRAM hardware in combination with fundamental improvements to the algorithm itself.

C. Comparison between QIPMs and CIPMs and comments on asymptotic speed-up

The above discussion suggests that the current outlook for practicality with a QIPM is pessimistic but simultaneously highlights several avenues by which to improve the results. Even with such improvements, if QIPMs are to one day be practical, they need to at least have an asymptotic speed-up over CIPMs. Here, we comment on this possibility. The core step of both QIPMs and CIPMs is the problem of computing a classical estimate of the solution to a linear system, a task that is also of broad use beyond IPMs. Thus, we need only compare different approaches to solving linear systems and our conclusions are relevant in any application where linear systems must be solved. Accordingly, in Table XI we give the asymptotic runtime of several approaches to solving an $L \times L$ linear system to precision ξ, including the QLSS-plus-tomography approach utilized by QIPMs, as well as two classical approaches. Whereas the prior literature (see, e.g., Ref. [13]) has primarily compared against Gaussian elimination (which scales as $\mathcal{O}(L^3)$), we also note a comparison against the randomized Kaczmarz method [56], which scales as $\mathcal{O}(L\kappa_F^2 \ln(\xi^{-1}))$. This scaling comes from the fact that $2\kappa_F^2 \ln(\xi^{-1})$ iterations are needed and each iteration involves computing several inner products at cost $\mathcal{O}(L)$. We observe that the worst-case cost of an iteration is $4L$ floating-point multiplications, meaning that all the constant prefactors involved are more or less mild. Thus, the asymptotic quantum advantage of the QIPM is limited to an amount equal to $\mathcal{O}(\min(\xi^2\kappa_F, \xi^2 L^2/\kappa_F))$, which is at most $\mathcal{O}(L)$ when $\kappa_F \propto L$ and $\xi = O(1)$. Encouragingly, our numerical results are consistent with $\kappa_F \propto L$. However, our results are not consistent with $\xi = \mathcal{O}(1)$, suggesting instead that ξ is decreasing with L.

If $\kappa_F \propto L$ and $\xi = \mathcal{O}(1)$, we would find a total QIPM runtime of $\mathcal{O}(n^{2.5})$, improving over classical $\mathcal{O}(n^{3.5})$ for a portfolio with n stocks. This speed-up would be a material asymptotic improvement over the classical complexity but leveraging this speed-up for a *practical* advantage might still be difficult. First, the difference in the constant prefactor between the quantum and classical algorithms would likely negate the speed-up unless n was taken to be very large. Second, the speed-up would necessarily be subquadratic. In the context of combinatorial optimization, where quadratic speed-ups can be obtained easily via Grover's algorithm, even a quadratic speed-up is unlikely to exhibit actual quantum advantage after factoring in slower quantum clock speeds and error-correction overheads [88].

Our results suggest that finding a practical quantum advantage for PO might require structural improvements to the QIPM itself. In particular, it may be necessary to explore whether additional components of the IPM can

TABLE XI. A comparison of the time complexities of different approaches for exactly or approximately solving an $L \times L$ linear system with Frobenius condition number κ_F to precision ξ. The comparison highlights how a quantum advantage only persists when κ_F is neither too large nor too small. The constant prefactor roughly captures the T-depth that we have found for the quantum case (the same prefactor from Table VI after discounting the $20\sqrt{2}$ IPM iteration factor) and the number of multiplications in the classical case.

Solver	Type	Complexity	Prefactor estimate
QLSS plus tomography	Quantum, approximate	$L\kappa_F\xi^{-2} \ln(L) \ln(\kappa_F\xi^{-1}L^{14/27})$	4×10^7
Gaussian elimination	Classical, exact	L^3	1/3
Randomized Kaczmarz [56]	Classical, approximate	$L\kappa_F^2 \ln(\xi^{-1})$	8

be quantized and whether the costly contribution of quantum state tomography could be completely circumvented. Naively, circumventing tomography entirely is challenging, as it is vitally important to retrieve a *classical* estimate of the solution to the linear system at each iteration in order to update the interior point and construct the linear system at the next iteration. Nevertheless, tomography represents a formidable bottleneck that must be addressed.

While our results are pessimistic on the question of whether QIPMs will deliver quantum advantage for PO (and other applications), it is our hope that by highlighting the precise issues leading to daunting resource counts, our work can inspire innovations that render quantum algorithms for optimization more practical. Finally, we conclude by noting that detailed end-to-end resource estimations of the kind we performed here are vitally important for commercial viability of quantum algorithms and quantum applications. While it is essential to discover and prove asymptotic speed-ups of quantum algorithms over classical, an asymptotic speed-up alone does not imply practicality. For this, a detailed end-to-end resource estimate is required, as the quantum algorithm may nevertheless be far from practical to implement. As we have seen, the devil is in the details, and there are many details behind which the devil can hide.

ACKNOWLEDGMENTS

We thank Brandon Augustino, Kyle Booth, Paul Burchard, Connor Hann, Iordanis Kerenidis, Anupam Prakash, Dániel Szilágyi, and Tamás Terlaky for helpful discussions. We are especially grateful to Earl Campbell for early collaboration during an initial phase of the project. G.S., H.K., and M.S. are thankful to Shantu Roy for his leadership, trust, and vision for the Intelligent and Advanced Compute Technologies team at AWS. We also thank James Tarantino for his support throughout the project.

APPENDIX A: NOTATION

Here, we list the important symbols that appear in our paper, for reference.

(1) Symbols related to PO:

- (a) n: number of stocks in the portfolio
- (b) $\mathbf{w}$: length-n vector indicating fraction of portfolio allocated to each stock (the object to be optimized)
- (c) $\bar{\mathbf{w}}$: length-n vector indicating current portfolio allocation
- (d) $\boldsymbol{\zeta}$: length-n vector indicating maximum allowable change to portfolio
- (e) $\hat{\mathbf{u}}$: length-n vector of average returns
- (f) Σ: $n \times n$ covariance matrix capturing deviations from average returns
- (g) q: parameter in objective function that determines relative weight of risk versus return [Eq. (3)]
- (h) M: $m \times n$ matrix corresponding to the square root of Σ, i.e., $\Sigma = M^\intercal M$
- (i) m: number of rows in M, often equal to the number of time epochs (Sec. III B)

(2) Symbols related to second-order cone programs:

- (a) $\mathcal{Q}^k$: second-order cone of dimension k [Eq. (4)]
- (b) $\mathcal{Q}$: product set of several second-order cones
- (c) $\mathbf{e}$: identity element for $\mathcal{Q}$ or $\mathcal{Q}^k$ (depending on context)
- (d) N: total number of variables in the SOCP
- (e) K: total number of linear constraints in the SOCP
- (f) r: number of second-order cone constraints in the program
- (g) $\mathbf{x}$: length-N vector; primal variable to be optimized, constrained to $\mathcal{Q}$
- (h) $\mathbf{y}$: length-K vector; dual variable to be optimized
- (i) $\mathbf{s}$: length-N vector, appears in dual program, constrained to $\mathcal{Q}$
- (j) A: $K \times N$ matrix encoding linear constraints [Eq. (5)]
- (k) $\mathbf{b}$: length-K vector encoding right-hand side of linear constraints [Eq. (5)]
- (l) $\mathbf{c}$: length-N vector encoding objective function [Eq. (5)]
- (m) $\mu(\mathbf{x}, \mathbf{s})$: duality gap of the primal-dual point $(\mathbf{x}, \mathbf{s})$ [Eq. (7)]
- (n) $\tau, \varkappa, \theta$: additional scalar variables introduced to implement self-dual embedding (Sec. III C 3)
- (o) $\mu(\mathbf{x}, \tau, \mathbf{s}, \varkappa)$: duality gap of the point $(\mathbf{x}, \tau, \mathbf{s}, \varkappa)$ of the self-dual SOCP [Eq. (14)]
- (p) X, S: arrowhead matrices for vectors $\mathbf{x}$ and $\mathbf{s}$ [Eq. (21)]
- (q) B: basis for null space of self-dual constraint matrix

(3) Symbols related to second-order cone programs for PO:

- (a) $\boldsymbol{\phi}$: length-n variable introduced during reduction from PO to SOCP; part of $\mathbf{x}$ [Eq. (10)]
- (b) $\boldsymbol{\rho}$: length-n variable introduced during reduction from PO to SOCP; part of $\mathbf{x}$ [Eq. (10)]
- (c) t: scalar variable introduced during reduction from PO to SOCP; part of $\mathbf{x}$ [Eq. (10)]
- (d) $\boldsymbol{\eta}$: length-m variable introduced during reduction from PO to SOCP; part of $\mathbf{x}$ [Eq. (10)]

(4) Symbols related to IPMs:

- (a) ν: parametrizes central path [Eq. (12)]

(b) $d_F(\mathbf{x}, \tau, \mathbf{s}, \varkappa)$: distance of the point $(\mathbf{x}, \tau, \mathbf{s}, \varkappa)$ to the central path of the self-dual SOCP [Eq. (13)]
(c) $\mathcal{N}$, $\mathcal{N}_F$: neighborhoods of the "central path" [Eqs. (27) and (28)]
(d) γ: radius of neighborhood of central path
(e) σ: step-length parameter
(f) L: size of (square) Newton matrix
(g) ϵ: input to IPM specifying error tolerance; algorithm terminates once duality gap falls beneath ϵ

(5) Important relations between parameters:

(a) Self-dual embedding has $2N + K + 3$ parameters and $N + K + 2$ linear constraints
(b) Newton matrix has size $L = 2N + K + 3$ for infeasible approach and $L = N + 1$ for feasible approach
(c) For PO formulation in Eq. (10), $N = 3n + m + 1$, $r = 3n + 1$, $K = 2n + m + 1$
(d) In our numerical experiments, we choose $m = 2n$

(6) Symbols related to QLSSs:

(a) G: $L \times L$ matrix encoding linear constraints
(b) $\mathbf{h}$: length-L vector encoding right-hand side of linear constraints
(c) $\mathbf{u}$: solution to linear system $G\mathbf{u} = \mathbf{h}$
(d) $\mathbf{v}$: normalized solution to linear system $\mathbf{u}/\|\mathbf{u}\|$
(e) $\varepsilon_{\mathrm{QLSP}}$: error in solution to linear system
(f) $\tilde{\mathbf{v}}$: normalized output of the QLSS, which should satisfy $\|\mathbf{v} - \tilde{\mathbf{v}}\| \leq \varepsilon_{\mathrm{QLSP}}$
(g) ℓ: $\lceil \log_2 L \rceil$
(h) U_G: block-encoding unitary for G
(i) ℓ_G: number of ancilla qubits used by U_G
(j) $U_{\mathbf{h}}$: state-preparation unitary for $|\mathbf{h}\rangle$
(k) $\kappa_F(G)$: Frobenius condition number $\|G\|_F \|G^{-1}\|$ of G
(l) Q: number of queries to U_G and $U_{\mathbf{h}}$ (Proposition 1)
(m) C: constant prefactor of κ_F (Proposition 1)
(n) d: the degree of the polynomial used in eigenstate filtering (Proposition 2)

(7) Symbols related to block-encoding and state preparation:

(a) ε_G: block-encoding error for matrix G
(b) $\varepsilon_{\mathbf{h}}$: state-preparation error for vector $\mathbf{h}$
(c) $\varepsilon_{\mathrm{ar}}$: gate-synthesis error for rotations needed by $CR^0(s)$ and $CR^1(s)$
(d) ε_z: gate-synthesis error for rotations needed by the QSP phases
(e) $\varepsilon_{\mathrm{qsp}}$: error due to polynomial approximation in eigenstate filtering
(f) $\varepsilon_{\mathrm{tsp}}$: error in preparing the state $\sum_{i=1}^{L} \sqrt{p_i}|i\rangle$ needed for the tomography routine
(g) N_{Qbe}, T_{Dbe}, and T_{Cbe}: number of logical qubits, T-depth, and T-count required for block-encoding
(h) N_{Qcbe}, T_{Dcbe}, and T_{Ccbe}: number of logical qubits, T-depth, and T-count required for *controlled* block-encoding
(i) N_{Qsp}, T_{Dsp}, and T_{Csp}: number of logical qubits, T-depth, and T-count required for state preparation
(j) N_{Qcsp}, T_{Dcsp}, and T_{Ccsp}: number of logical qubits, T-depth, and T-count required for *controlled*-state preparation

(8) Symbols related to tomography:

(a) k: number of measurements on independent copies of the state
(b) δ: probability of failure
(c) ε: guaranteed error of tomographic estimate
(d) ξ: overall precision of solution to linear system, dominated by tomographic error

APPENDIX B: DEFERRED PROOFS

1. Quantum state tomography

Proof of Proposition 3.—Consider a single coordinate α_j with associated probability $p_j = |\alpha_j|^2$ and suppose that we take k samples to find an estimate $\tilde{p}_j$ of p_j. By Bernstein's inequality,

$$\Pr[|\tilde{p}_j - p_j| > \varepsilon_j] \leq 2\exp\left(-\frac{\varepsilon^2}{2(p_j + \varepsilon/3)}k\right) \tag{B1}$$

and so for a given component-wise target deviation in the probability ε_j, choosing

$$k \geq \frac{2(p_j + \varepsilon/3)}{\varepsilon^2}\ln(2/\delta') = \frac{2(|\alpha_j|^2 + \varepsilon/3)}{\varepsilon^2}\ln(2/\delta') \tag{B2}$$

guarantees that $\Pr[|\tilde{p}_j - p_j| > \varepsilon_j] \leq \delta'$.

We now pick $\varepsilon_j = \sqrt{3\gamma}|\alpha_j|\varepsilon + \gamma\varepsilon^2$ for some yet undetermined $\gamma > 0$. With this choice,

$$\begin{aligned}
&\frac{2(|\alpha_j|^2+\frac{\varepsilon}{3})}{\varepsilon^2}\ln(2/\delta')\\
&=\frac{2(|\alpha_j|^2+\sqrt{\frac{\gamma}{3}}\varepsilon+\frac{\gamma}{3}\varepsilon^2)}{(\sqrt{3\gamma}|\alpha_j|\varepsilon+\gamma\varepsilon^2)^2}\ln(2/\delta')\\
&\le\frac{2(|\alpha_j|^2+2\sqrt{\frac{\gamma}{3}}\varepsilon+\frac{\gamma}{3}\varepsilon^2)}{3\gamma\varepsilon^2(|\alpha_j|+\sqrt{\frac{\gamma}{3}}\varepsilon)^2}\ln(2/\delta')\\
&=\frac{2}{3\gamma\varepsilon^2}\ln(2/\delta') \qquad \text{(B3)}
\end{aligned}$$

and hence it suffices to choose $k=(2/(3\gamma\varepsilon^2))\ln(2/\delta')$. Letting $\delta'=\delta/L$, the union bound implies that for $k=(2/(3\gamma\varepsilon^2))\ln(2L/\delta)$, all estimates $\tilde{p}_j$ satisfy $|\tilde{p}_j-p_j|\le\varepsilon_j$. We now bound the distance between $|\tilde{\alpha}_j|$ and $|\alpha_j|$. First,

$$\begin{aligned}
|\tilde{\alpha}_j|-|\alpha_j|&\le\sqrt{p_j+\varepsilon}-|\alpha_j|\\
&=\sqrt{|\alpha_j|^2+\sqrt{3\gamma}|\alpha_j|\varepsilon+\gamma\varepsilon^2}-|\alpha_j|\\
&\le(|\alpha_j|+\sqrt{\gamma}\varepsilon)-|\alpha_j|\\
&=\sqrt{\gamma}\varepsilon. \qquad \text{(B4)}
\end{aligned}$$

Next, we bound $|\alpha_j|-|\tilde{\alpha}_j|$. If $p_j\le\varepsilon_j$, then

$$|\alpha_j|^2\le\sqrt{3\gamma}|\alpha_j|\varepsilon+\gamma\varepsilon^2\Leftrightarrow|\alpha_j|\le\frac{(\sqrt{3}+\sqrt{7})\sqrt{\gamma}}{2}\varepsilon, \qquad \text{(B5)}$$

while if $p_j>\varepsilon_j$,

$$\begin{aligned}
|\alpha_j|-|\tilde{\alpha}_j|&\le|\alpha_j|-\sqrt{p_j-\varepsilon_j}\\
&=|\alpha_j|-\sqrt{|\alpha_j|^2-\sqrt{3\gamma}|\alpha_j|\varepsilon-\gamma\varepsilon^2}\\
&<\frac{(\sqrt{3}+\sqrt{7})\sqrt{\gamma}}{2}\varepsilon, \qquad \text{(B6)}
\end{aligned}$$

which follows because the function $f(x)=x-\sqrt{x^2-\sqrt{3}x-1}$ has its maximum at $f\left(\left(\sqrt{3}+\sqrt{7}\right)/2\right)=\left(\sqrt{3}+\sqrt{7}\right)/2$. Therefore, with the choice $\gamma=\left(\left(\sqrt{3}+\sqrt{7}\right)/2\right)^{-2}$, we can guarantee that $||\tilde{\alpha}_j|-|\alpha_j||\le\varepsilon$, which corresponds to

$$k=\frac{2}{3\gamma\varepsilon^2}\ln(2L/\delta)=\frac{5+\sqrt{21}}{3\varepsilon^2}\ln(2L/\delta) \qquad \text{(B7)}$$

measurements. ■

Proof of Proposition 4.—Define $\varepsilon'=\varepsilon\sqrt{1-\varepsilon^2/4}/\sqrt{2L}$. Then, $k=28.75\varepsilon'^{-2}\ln(6L/\delta)$. Consider the following three assertions:

(1) The estimates p_i satisfy $|\sqrt{p_i}-|\tilde{v}_i|\sqrt{p}|\le\varepsilon'/3$ for all i.
(2) The estimates $p_i^+=k_i^+/k$ satisfy

$$\left|\sqrt{p_i^+}-\frac{|\sqrt{p}\tilde{v}_i+\sqrt{p_i'}|}{2}\right|\le\varepsilon'/3$$

and the estimates $p_i^-=k_i^-/k$ satisfy

$$\left|\sqrt{p_i^-}-\frac{|\sqrt{p}\tilde{v}_i-\sqrt{p_i'}|}{2}\right|\le\varepsilon'/3,$$

for all i.
(3) The actual amplitudes $\sqrt{p_i'}$ of the state created in the second step satisfy $|\sqrt{p_i'}-\sqrt{p_i}|\le\varepsilon_{\text{tsp}}$.

From Proposition 3, we know that assertion (1) holds with probability at least $1-\delta/3$, and that assertion (2) holds with probability at least $1-2\delta/3$. Therefore, both assertions hold with probability at least $1-\delta$. Moreover, assertion (3) holds by assumption. From here on, we will assume that all three assertions hold.

Let a_i be the real part and let b_i be the imaginary part of the quantity $\sqrt{p}\tilde{v}_i$. Let $r_i^+=|\sqrt{p}\tilde{v}_i+\sqrt{p_i}|$ and let $r_i^-=|\sqrt{p}\tilde{v}_i-\sqrt{p_i}|$. Note that r_i^+ and r_i^- are proportional to the absolute value of the ideal amplitudes of the state created in Eq. (49). One can show that

$$a_i=\frac{(r_i^+)^2-(r_i^-)^2}{4\sqrt{p_i}}. \qquad \text{(B8)}$$

Define $f_i(x,y)=(x^2-y^2)/\sqrt{p_i}$; then, $a_i=f_i(r_i^+/2,r_i^-/2)$. We first note that the estimates $\sqrt{p_i^\pm}$ give us good approximations of $r_i^+/2$ and $r_i^-/2$:

$$\left|\sqrt{p_i^\pm}-\frac{r^\pm}{2}\right|\le\frac{\varepsilon'}{3}\varepsilon+\frac{\varepsilon_{\text{tsp}}}{2}, \qquad \text{(B9)}$$

which follows from assertions (2) and (3). The amplitudes $\tilde{a}_i$ that define the estimate output by the tomography algorithm are given in Eq. (52), which can now be rewritten as

$$\tilde{a}_i = \begin{cases} 0, & \sqrt{p_i} \leq \frac{2}{3}\varepsilon' + \varepsilon_{\text{tsp}}; \text{ else,} \\ \min\left(\sqrt{p_i}, f_i\left(\sqrt{p_i^+}, \sqrt{p_i^-}\right)\right), & f_i\left(\sqrt{p_i^+}, \sqrt{p_i^-}\right) \geq 0, \\ \max\left(-\sqrt{p_i}, f_i\left(\sqrt{p_i^+}, \sqrt{p_i^-}\right)\right), & f_i\left(\sqrt{p_i^+}, \sqrt{p_i^-}\right) < 0. \end{cases} \tag{B10}$$

We prove that the $\tilde{a}_i$ values approximate the a_i values, specifically

$$|\tilde{a}_i - a_i| \leq \varepsilon' + \varepsilon_{\text{tsp}} + |b_i|. \tag{B11}$$

We will prove the above claim using a case-by-case analysis. Assume that $a_i \geq 0$; the case $a_i < 0$ will proceed similarly.

First, consider the case $\sqrt{p_i} \leq 2\,\varepsilon'/3 + \varepsilon_{\text{tsp}}$. In this case, $\tilde{a}_i = 0$ and $a_i \leq \sqrt{p}|\tilde{v}_i| \leq \sqrt{p_i} + \varepsilon'/3 \leq \varepsilon' + \varepsilon_{\text{tsp}}$, so $|\tilde{a}_i - a_i| \leq \varepsilon + \varepsilon_{\text{tsp}}$.

Second, consider the case $f_i(\sqrt{p_i^+}, \sqrt{p_i^-}) \geq a_i$. From the definition of $\tilde{a}_i$ and assertion (1), we have $\tilde{a}_i \leq \sqrt{p_i} \leq \sqrt{p}|\tilde{v}_i| + \varepsilon'/3$ and thus

$$\begin{aligned} \tilde{a}_i - a_i &\leq \sqrt{p}|\tilde{v}_i| - a_i + \frac{\varepsilon'}{3} \\ &= \sqrt{a_i^2 + b_i^2} - a_i + \frac{\varepsilon'}{3} \leq |b_i| + \frac{\varepsilon'}{3}. \end{aligned} \tag{B12}$$

We also have [again invoking assertion (1)]

$$a_i - \tilde{a}_i \leq a_i - \sqrt{p_i} \leq a_i - \sqrt{p}|\tilde{v}_i| + \frac{\varepsilon'}{3} \leq \frac{\varepsilon'}{3} \tag{B13}$$

and thus, $|a_i - \tilde{a}_i| \leq |b_i| + \varepsilon'/3$.

Finally, consider the case $f_i(\sqrt{p_i^+}, \sqrt{p_i^-}) < a_i$. Defining $\tilde{\varepsilon} = 2\,\varepsilon'/3 + \varepsilon_{\text{tsp}}$, we can lower bound $f_i(\sqrt{p_i^+}, \sqrt{p_i^-})$:

$$\begin{aligned} f_i(\sqrt{p_i^+}, \sqrt{p_i^-}) &= \frac{(2\sqrt{p_i^+})^2 - (2\sqrt{p_i^+})^2}{4\sqrt{p_i}} \\ &\geq \frac{(r_i^+ - \tilde{\varepsilon})^2 - (r_i^- + \tilde{\varepsilon})^2}{4\sqrt{p_i}} \\ &= \frac{(r_i^+)^2 - (r_i^-)^2}{4\sqrt{p_i}} - \tilde{\varepsilon}\frac{r_i^+ + r_i^-}{2\sqrt{p_i}} \\ &= a_i - \tilde{\varepsilon}\frac{r_i^+ + r_i^-}{2\sqrt{p_i}}. \end{aligned} \tag{B14}$$

Here, in the second line, we have used Eq. (B9) and the fact that $r_i^+ \geq \sqrt{p_i} \geq 2\,\varepsilon'/3 + \varepsilon_{\text{tsp}}$. We now upper bound $r_i^+ + r_i^-$:

$$\begin{aligned} r^+ + r^- &= \sqrt{(a_i + \sqrt{p_i})^2 + b_i^2} + \sqrt{(a_i - \sqrt{p_i})^2 + b_i^2} \\ &\leq |a_i + \sqrt{p_i}| + |a_i - \sqrt{p_i}| + 2|b_i| \\ &= 2\max(a_i, \sqrt{p_i}) + 2|b_i| \\ &\leq 2(\sqrt{p_i} + \varepsilon'/3 + |b_i|), \end{aligned} \tag{B15}$$

where in the fourth line we have used $a_i \leq \sqrt{a_i^2 + b_i^2} = \sqrt{p}|\tilde{v}_i| \leq \sqrt{p_i} + \varepsilon'/3$ [assertion (1)]. Therefore,

$$\begin{aligned} f_i(\sqrt{p_i^+}, \sqrt{p_i^-}) &= a_i - \tilde{\varepsilon}\frac{r_i^+ + r_i^-}{2\sqrt{p_i}} \\ &\geq a_i - \tilde{\varepsilon}\frac{2(\sqrt{p_i} + \varepsilon'/3 + |b_i|)}{2\sqrt{p_i}} \\ &= a_i - \tilde{\varepsilon} - \frac{\tilde{\varepsilon}}{\sqrt{p_i}}(\varepsilon'/3 + |b_i|)) \\ &\geq a_i - (\varepsilon' + \varepsilon_{\text{tsp}} + |b_i|), \end{aligned} \tag{B16}$$

where in the fourth line we have used $\tilde{\varepsilon}/\sqrt{p_i} \leq 1$. This implies that

$$\begin{aligned} |\tilde{a}_i - a_i| &= a_i - \tilde{a}_i \\ &\leq a_i - \min(f_i(\sqrt{p_i^+}, \sqrt{p_i^-}), \sqrt{p_i}) \\ &\leq \varepsilon' + \varepsilon_{\text{tsp}} + |b_i|. \end{aligned} \tag{B17}$$

Here, we have used $a_i - \sqrt{p_i} \leq \sqrt{p}|\tilde{v}_i| - \sqrt{p_i} \leq \varepsilon'/3$.

TABLE XII. The fit parameters for the Frobenius condition number for the four horizontal-axis locations considered on the scaling plot of Fig. 13. The uncertainties correspond to one-standard-deviation errors on the parameter estimates from the fit. We note that both versions have similar empirical scaling, although the fits are better for IF-QIPM-QR. The constant prefactors are superior for the IF-QIPM-QR version, but calculating the QR decomposition requires a one-time classical cost proportional to $\mathcal{O}(L^3)$.

Duality gap	IF-QIPM	IF-QIPM-QR
1.0	$\kappa_F(G) \sim n^{0.57\pm0.60}$	$\kappa_F(G) \sim n^{0.228\pm0.002}$
0.1	$\kappa_F(G) \sim n^{0.58\pm0.28}$	$\kappa_F(G) \sim n^{0.66\pm0.03}$
0.01	$\kappa_F(G) \sim n^{0.81\pm0.53}$	$\kappa_F(G) \sim n^{0.73\pm0.03}$
0.001	$\kappa_F(G) \sim n^{1.01\pm0.77}$	$\kappa_F(G) \sim n^{0.98\pm0.04}$

TABLE XIII. The fit parameters for the square of the inverse of the required tomography precision to stay near the central path, corresponding to Fig. 14. The uncertainties correspond to one-standard-deviation errors on the parameter estimates from the fit.

Duality gap	IF-QIPM	IF-QIPM-QR
1.0	$\xi^{-2} \sim \mathcal{O}(n^{-0.01\pm0.02})$	$\xi^{-2} \sim \mathcal{O}(n^{-0.11\pm0.07})$
0.1	$\xi^{-2} \sim \mathcal{O}(n^{-0.99\pm0.41})$	$\xi^{-2} \sim \mathcal{O}(n^{-0.46\pm0.11})$
0.01	$\xi^{-2} \sim \mathcal{O}(n^{0.53\pm0.91})$	$\xi^{-2} \sim \mathcal{O}(n^{0.89\pm0.15})$
0.001	$\xi^{-2} \sim \mathcal{O}(n^{0.93\pm0.66})$	$\xi^{-2} \sim \mathcal{O}(n^{0.90\pm0.15})$

We have shown that $|\tilde{a}_i - a_i| \leq \varepsilon' + \varepsilon_{\text{tsp}} + |b_i|$ for all cases. Therefore,

$$\begin{aligned}\|\tilde{\mathbf{a}} - \mathbf{a}\|_2^2 &\leq \sum_i \left[(\varepsilon' + \varepsilon_{\text{tsp}})^2 + 2|b_i|(\varepsilon' + \varepsilon_{\text{tsp}}) + b_i^2\right] \\ &\leq L(\varepsilon' + \varepsilon_{\text{tsp}})^2 + 2(\varepsilon' + \varepsilon_{\text{tsp}})\sqrt{L\sum_i b_i^2} + \sum_i b_i^2 \\ &= \left(\sqrt{L}(\varepsilon' + \varepsilon_{\text{tsp}}) + \sqrt{\sum_i b_i^2}\right)^2 \end{aligned} \tag{B18}$$

and hence

$$\begin{aligned}\|\tilde{\mathbf{a}} - \sqrt{p}\mathbf{v}\|_2 &\leq \|\tilde{\mathbf{a}} - \mathbf{a}\|_2 + \|\mathbf{a} - \sqrt{p}\mathbf{v}\|_2 \\ &\leq \sqrt{L}(\varepsilon' + \varepsilon_{\text{tsp}}) + \sqrt{\sum_i b_i^2} + \sqrt{\sum_i (\sqrt{p}v_i - a_i)^2} \\ &\leq \sqrt{L}(\varepsilon' + \varepsilon_{\text{tsp}}) + \sqrt{2p}\varepsilon_{\text{QLSP}}, \end{aligned} \tag{B19}$$

where we have used $\sum_i ((v_i - a_i/\sqrt{p})^2 + b_i^2/p) \leq \varepsilon_{\text{QLSP}}^2$. Since $\tilde{\mathbf{v}}' \propto \tilde{\mathbf{a}}$, for some proportionality factor λ we have $\|\lambda\tilde{\mathbf{v}}' - \mathbf{v}\| \leq \sqrt{2L}(\varepsilon' + \varepsilon_{\text{tsp}}) + \sqrt{2}\varepsilon_{\text{QLSP}}$, where we have used $p \geq 1/2$. A bit of geometry will show that if $\|\mathbf{c} - \mathbf{d}\|_2 \leq \gamma < 1$ and $\|\mathbf{d}\|_2 = 1$, then $\|\mathbf{c}/\|\mathbf{c}\|_2 - \mathbf{d}\|_2 \leq g(\gamma) \equiv 2\sin(\frac{1}{2}\sin^{-1}\gamma) = \sqrt{1+\gamma} - \sqrt{1-\gamma}$. Applying this with $\mathbf{c} = \lambda\tilde{\mathbf{v}}'$ and $\mathbf{d} = \mathbf{v}$, we obtain

$$\begin{aligned}&\|\tilde{\mathbf{v}}' - \mathbf{v}\|_2 \\ &\quad \leq g(\sqrt{2L}(\varepsilon' + \varepsilon_{\text{tsp}}) + \sqrt{2}\varepsilon_{\text{QLSP}}) \\ &\quad < g(\sqrt{2L}\varepsilon') \\ &\quad + (\sqrt{2L}\varepsilon_{\text{tsp}} + \sqrt{2}\varepsilon_{\text{QLSP}})\left.\frac{dg}{dx}\right|_{x=\sqrt{2L}(\varepsilon'+\varepsilon_{\text{tsp}})+\sqrt{2}\varepsilon_{\text{QLSP}}} \\ &\quad < \varepsilon + 1.58\sqrt{L}\varepsilon_{\text{tsp}} + 1.58\varepsilon_{\text{QLSP}}, \end{aligned} \tag{B20}$$

as claimed. In the second inequality, we have used the convexity of g; and in the third inequality, we have used the fact that $g(\sqrt{2L}\varepsilon') = \varepsilon$, $\sqrt{2L}(\varepsilon' + \varepsilon_{\text{tsp}}) + \sqrt{2}\varepsilon_{\text{QLSP}} < \varepsilon + \sqrt{2L}\varepsilon_{\text{tsp}} + \sqrt{2}\varepsilon_{\text{QLSP}} \leq 1/2$, and $\sqrt{2}g'(1/2) < 1.58$. ■

TABLE XIV. The estimated scaling of the quantum algorithm as a function of the portfolio size for the two feasible versions of the quantum algorithm, corresponding to Fig. 15. The uncertainties correspond to one-standard-deviation errors on the parameter estimates from the fit.

Duality gap	IF-QIPM	IF-QIPM-QR
1.0	$\mathcal{O}(n^{1.41\pm0.01})$	$\mathcal{O}(n^{2.07\pm0.15})$
0.1	$\mathcal{O}(n^{1.23\pm0.40})$	$\mathcal{O}(n^{1.77\pm0.15})$
0.01	$\mathcal{O}(n^{2.87\pm0.91})$	$\mathcal{O}(n^{3.13\pm0.18})$
0.001	$\mathcal{O}(n^{3.54\pm0.64})$	$\mathcal{O}(n^{3.50\pm0.10})$

APPENDIX C: NULL-SPACE MATRIX FOR PORTFOLIO OPTIMIZATION

In Sec. III C, an inexact-feasible IPM has been described that requires as input a matrix B with columns that form a basis for the null space of the feasibility equations for the self-dual SOCP that appears in Eq. (19). A straightforward way to find such a B, in general, would be to perform a QR decomposition of the constraint matrix, costing classical $O(N^3)$ runtime (or, using techniques for fast matrix multiplication, between $O(N^2)$ and $O(N^3)$ time [89,90]). The upshot is that B need only be computed once and does not change with each iteration of the algorithm but, depending on other parameters of the problem, this classical runtime could dominate the overall complexity. Alternatively, in many specific cases, including ours, a valid matrix B can be determined by inspection. For example, suppose that we have a $(N-K)\times N$ matrix Q_A with full column rank for which $AQ_A = 0$, a $K \times (K-1)$ matrix P with full column rank for which $\bar{\mathbf{b}}^\intercal P = 0$, and a point $\mathbf{x}_0$ for which $A\mathbf{x}_0 = \mathbf{b}$. Then, letting $\gamma = \mathbf{b}^\intercal\bar{\mathbf{b}}/||\bar{\mathbf{b}}||^2$, a valid choice for B is

$$B = \begin{array}{c} \mathbf{x} \\ \mathbf{y} \\ \tau \\ \theta \\ \mathbf{s} \\ \kappa \end{array} \begin{pmatrix} 0 & Q_A & \mathbf{e} & \mathbf{x}_0 \\ P & \bar{\mathbf{b}}\dfrac{\bar{\mathbf{c}}^\intercal Q_A}{||\bar{\mathbf{b}}||^2} & -\dfrac{(r+1)}{||\bar{\mathbf{b}}||^2}\bar{\mathbf{b}} & \dfrac{\bar{\mathbf{c}}^\intercal\mathbf{x}_0 - \bar{z}}{||\bar{\mathbf{b}}||^2}\bar{\mathbf{b}} \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ -A^\intercal P & -A^\intercal\bar{\mathbf{b}}\dfrac{\bar{\mathbf{c}}^\intercal Q_A}{||\bar{\mathbf{b}}||^2} & \dfrac{r+1}{||\bar{\mathbf{b}}||^2}A^\intercal\bar{\mathbf{b}} + \mathbf{e} & \dfrac{-\bar{\mathbf{c}}^\intercal\mathbf{x}_0 + \bar{z}}{||\bar{\mathbf{b}}||^2}A^\intercal\bar{\mathbf{b}} + \mathbf{c} \\ \mathbf{b}^\intercal P & (\gamma-1)\mathbf{c}^\intercal Q_A - \gamma\mathbf{e}^\intercal Q_A & 1-\gamma(r+1) & -\gamma\bar{z} + (\gamma-1)\mathbf{c}^\intercal\mathbf{x}_0 - \gamma\mathbf{e}^\intercal\mathbf{x}_0 \end{pmatrix}. \tag{C1}$$

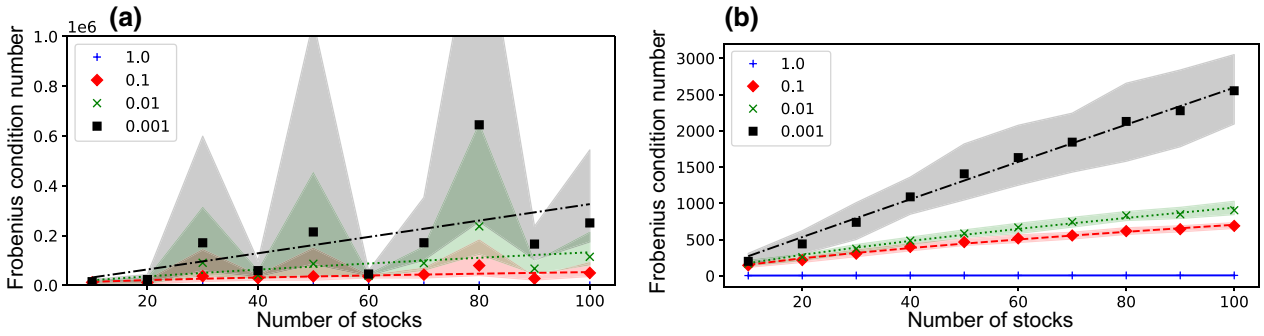

FIG. 13. The median Frobenius condition number for 128 randomly sampled stock portfolios from the DWCF index as a function of the portfolio size for duality gaps of 1.0, 0.1, 0.01, and 0.001: (a) IF-QIPM; (b) IF-QIPM-QR. The error bars show the 68th percentile, which corresponds to one standard deviation if the distribution is Gaussian. We find that a linear trend appears to work quite well for the IF-QIPM-QR case but that the IF-QIPM is quite noisy. For each duality gap, we also plot a power-law fit of the form an^b and report the values of b in Table XII.

The leftmost column in the above block matrix corresponds to $K-1$ basis vectors formed by choosing $\mathbf{y}$ to be a vector perpendicular to $\bar{\mathbf{b}}$ and $\mathbf{x}=\mathbf{0}$, $\tau=\theta=0$. The second column corresponds to $N-K$ vectors formed by choosing $\mathbf{x}$ to be in the null space of A and letting $\tau=\theta=0$, with $\mathbf{y}=(\bar{\mathbf{c}}^\intercal\mathbf{x}/||\bar{\mathbf{b}}||^2)\bar{\mathbf{b}}$. The third column corresponds to the vector formed by choosing $\mathbf{x}=\mathbf{e}$, $\tau=\theta=1$ and then $\mathbf{y}=-((r+1)/||\bar{\mathbf{b}}||^2)\bar{\mathbf{b}}$. The final column corresponds to choosing $\mathbf{x}=\mathbf{x}_0$, $\tau=1$, $\theta=0$, and $\mathbf{y}=((\bar{\mathbf{c}}^\intercal\mathbf{x}_0-\bar{z})/||\bar{\mathbf{b}}||^2)\bar{\mathbf{b}}$. In each case, the choices of $\mathbf{x}$, $\mathbf{y}$, τ, and θ uniquely determine the values of $\mathbf{s}$ and κ. Note that in practice, the second and fourth block rows of B can be ignored because in Eq. (22) they are left multiplied by a matrix the second and fourth block columns of which are zero.

What remains is to specify P, Q_A, and $\mathbf{x}_0$ for the case of PO, given in Eq. (10). Finding a valid matrix P is straightforward. Note that from Eq. (10), we have $\mathbf{b}=(1;\bar{\mathbf{w}}+\boldsymbol{\zeta};\bar{\mathbf{w}}-\boldsymbol{\zeta};\mathbf{0})$. For $j=1,\ldots,2n$, we let $\mathbf{p}_j$ have a 1 in its first entry and a $-1/b_{j+1}$ in its $(j+1)$th entry, with zeros elsewhere. For $j=2n+1,\ldots,2n+m$, we let $\mathbf{p}_j$ have a single 1 in its $(j+1)$th entry and zeros elsewhere. Thus, the $\mathbf{p}_j$ are independent and $\mathbf{b}^\intercal\mathbf{p}_j=0$ for all j. We then define the matrix P by $P=(\mathbf{p}_1,\ldots,\mathbf{p}_{2n+m})$. Similarly, we can generate the columns of a valid matrix Q_A as follows. Given a choice of $\mathbf{w}$ such that $\mathbf{1}^\intercal\mathbf{w}=0$, we choose $\boldsymbol{\phi}=-\mathbf{w}$, $\boldsymbol{\rho}=\mathbf{w}$, $t=0$, and $\boldsymbol{\eta}=M\mathbf{w}$. As there are $n-1$ linearly independent choices of $\mathbf{w}$ [e.g., the vectors $(1;-1;0;0;\ldots;0)$, $(0;1;-1;0;\ldots;0)$, $(0;0;1;-1;\ldots;0)$, etc.], this leads to $n-1$ linearly independent columns of Q_A. A final nth column can be formed by choosing $t=1$ and $\mathbf{w}=\boldsymbol{\phi}=\boldsymbol{\rho}=\mathbf{0}$ and $\boldsymbol{\eta}=\mathbf{0}$. Finally, the point $\mathbf{x}_0$ can be chosen by letting $\mathbf{w}=\bar{\mathbf{w}}$, $\boldsymbol{\phi}=\boldsymbol{\rho}=\boldsymbol{\zeta}$, $t=0$, and $\boldsymbol{\eta}=M\bar{\mathbf{w}}$.

APPENDIX D: ALTERNATIVE SEARCH DIRECTIONS

The solution $(\Delta\mathbf{x};\Delta\mathbf{y};\Delta\tau;\Delta\theta;\Delta\mathbf{s};\Delta\varkappa)$ to the Newton systems in Eqs. (19) and (22) is one possible *search direction* for the IPM. Alternative search directions can be found by applying a scale transformation to the convex set. We

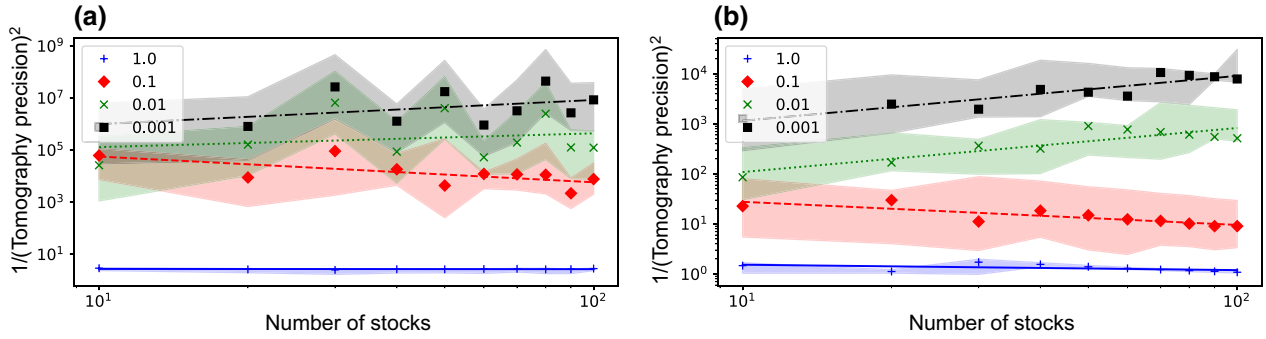

FIG. 14. Median value of the square of the required inverse-tomography precision required to remain in the neighborhood of the central path for 128 randomly sampled stock portfolios from the DWCF index as a function of portfolio size for duality gaps of 1.0, 0.1, 0.01, and 0.001: (a) IF-QIPM; (b) IF-QIPM-QR. The error bars show the 68th percentile, which corresponds to one standard deviation if the distribution is Gaussian. For each duality, gap, we also plot a linear fit on the log-log data, and report the corresponding slope in Table XIII.

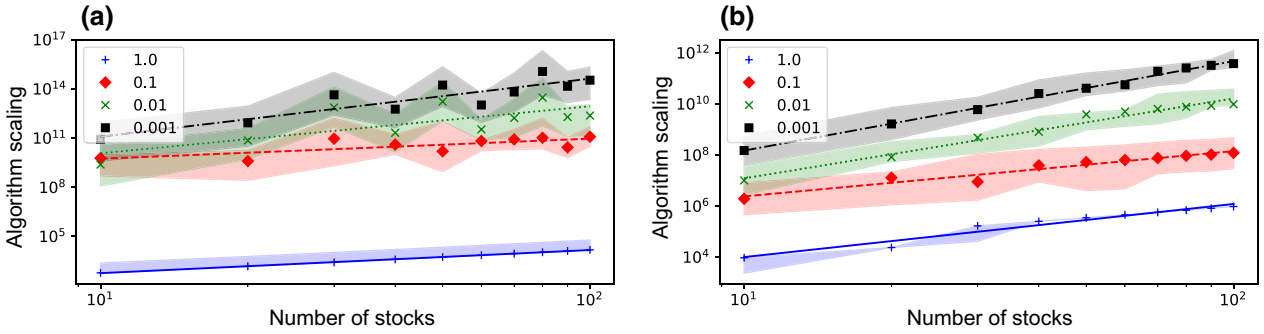

FIG. 15. The median value of the estimated algorithm scaling factor, computed as the median of $n^{1.5}\kappa_F/\xi^2$ for 128 randomly sampled stock portfolios from the DWCF index as a function of the portfolio size for duality gaps of 1.0, 0.1, 0.01, and 0.001: (a) IF-QIPM; (b) IF-QIPM-QR. The error bars show the 68th percentile, which corresponds to one standard deviation if the distribution is Gaussian. For each duality gap, we also plot a linear fit on the log-log data and report the corresponding slope in Table XIV.

follow Ref. [53] and, for the k-dimensional second-order cone $\mathcal{Q}^k$, we define the set

$$\mathcal{G}^k = \left\{\lambda T : \lambda > 0, T^\intercal \begin{pmatrix} 1 & 0 \\ 0 & -I \end{pmatrix} T = \begin{pmatrix} 1 & 0 \\ 0 & -I \end{pmatrix}\right\}. \quad \text{(D1)}$$

For the product $\mathcal{Q}$ of multiple cones, we let the set $\mathcal{G}$ consist of direct sums of entries from $\mathcal{G}^k$. This definition implies that the matrices $G \in \mathcal{G}$ map the set $\mathcal{Q}$ onto itself. Thus, for a fixed choice $G \in \mathcal{G}$, we may consider a change of variables $\mathbf{x}' = G^\intercal \mathbf{x}$, $\mathbf{s}' = G^{-1}\mathbf{s}$, $\mathbf{y}' = \mathbf{y}$. We let X' and S' be the arrowhead matrices for $\mathbf{x}'$ and $\mathbf{s}'$ and, following the same logic as above, we arrive at a Newton system

$$\begin{pmatrix} S'G^\intercal & 0 & 0 & 0 & X'G^{-1} & 0 \\ 0 & 0 & \varkappa & 0 & 0 & \tau \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\tau \\ \Delta\theta \\ \Delta\mathbf{s} \\ \Delta\varkappa \end{pmatrix} = \begin{pmatrix} \sigma\mu\mathbf{e} - X'S'\mathbf{e} \\ \sigma\mu - \varkappa\tau \end{pmatrix}. \quad \text{(D2)}$$

The solution to this linear set of equations [along with the feasibility equations of Eq. (19)] will be distinct for different choices of G. The choice $G = I$ recovers Eq. (22) and is called the Alizadeh-Haeberly-Overton (AHO) direction. Reference [53] has shown that the IPM can reduce the duality gap by a constant factor after $O(\sqrt{r})$ iterations for any choice of G. However, some choices of G can yield additional potentially desirable properties; e.g., the Nesterov-Todd search direction scales the cone such that $\mathbf{x}' = \mathbf{s}'$. However, in our numerical simulations of the QIPM, we have not observed any obvious benefits of choosing a search direction other than the AHO direction.

APPENDIX E: NUMERICAL RESULTS FOR FEASIBLE QIPMs

In Sec. VI, we have presented numerical results for the "II-QIPM," for which intermediate points could be infeasible. Here, we also present some results for two variants of the "feasible" QIPM, inspired by the work of Ref. [14], denoted by "IF-QIPM" and "IF-QIPM-QR," as summarized in Table II. The IF-QIPM uses the null-space basis B outlined in Appendix C, whereas the IF-QIPM-QR version uses a null-space basis B determined using a QR decomposition. In all cases, we have simulated the algorithm for enough iterations to reduce the duality gap to 10^{-3}, whereas for the II-QIPM we have simulated down to 10^{-7}.

In Figs. 13–15, we present results for the feasible IPMs that are analogous to those displayed in Figs. 9–11 for the infeasible case. We find that the IF-QIPM-QR has the best performance, though this must be weighed against the fact that an expensive QR decomposition must be classically precomputed to implement this method. However, the advantage of the IF-QIPM-QR method is not large enough for any of the qualitative conclusions in Sec. VII to change. The IF-QIPM method has the worst performance, which we believe is due to the fact that the null-space basis found by inspection turns out to be a very ill-conditioned matrix (its condition number was observed to be in the vicinity of 1000). Additionally, the IF-QIPM appears to have the largest instance-to-instance variation of any of the methods, leading to lower-quality numerical fits.

[1] S. M. Stigler, Gauss and the invention of least squares, Ann. Stat. **9**, 465 (1981).
[2] B. Apolloni, C. Carvalho, and D. de Falco, Quantum stochastic optimization, Stoch. Process. Their Appl. **33**, 233 (1989).
[3] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm (2014).

[4] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, A. Kandala, A. Mezzacapo, P. Müller, W. Riess, G. Salis, J. Smolin, I. Tavernelli, and K. Temme, Quantum optimization using variational algorithms on near-term quantum devices, Quantum Sci. Technol. **3**, 030503 (2018).

[5] F. G. Brandao and K. M. Svore, in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (2017), p. 415.

[6] F. G. S. L. Brandão, A. Kalev, T. Li, C. Y.-Y. Lin, K. M. Svore, and X. Wu, in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 132, edited by C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019), p. 27:1.

[7] J. van Apeldoorn, A. Gilyén, S. Gribling, and R. de Wolf, Quantum SDP-solvers: Better upper and lower bounds, Quantum **4**, 230 (2020).

[8] J. van Apeldoorn and A. Gilyén, in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 132, edited by C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019), p. 99:1.

[9] F. G. S L. Brandão, R. Kueng, and D. Stilck França, Faster quantum and classical SDP approximations for quadratic binary optimization, Quantum **6**, 625 (2022).

[10] I. Kerenidis and A. Prakash, A quantum interior point method for LPs and SDPs, ACM Trans. Quantum Comput. **1**, 1 (2020).

[11] B. Augustino, G. Nannicini, T. Terlaky, and L. F. Zuluaga, Quantum interior point methods for semidefinite optimization, arXiv preprint ArXiv:2112.06025 (2021).

[12] B. Huang, S. Jiang, Z. Song, R. Tao, and R. Zhang, A faster quantum algorithm for semidefinite programming via robust IPM framework, arXiv preprint ArXiv:2207.11154 (2022).

[13] I. Kerenidis, A. Prakash, and D. Szilágyi, Quantum algorithms for second-order cone programming and support vector machines, Quantum **5**, 427 (2021).

[14] B. Augustino, M. Mohammadisiahroudi, T. Terlaky, and L. F. Zuluaga, An inexact-feasible quantum interior point method for second-order cone optimization (2022), in preparation. https://engineering.lehigh.edu/sites/engineering.lehigh.edu/files/_DEPARTMENTS/ise/pdf/tech-papers/21/21T_009a.pdf.

[15] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**, 150502 (2009).

[16] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, SIAM J. Comput. **46**, 1920 (2017).

[17] Y. Subasi, R. D. Somma, and D. Orsucci, Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing, Phys. Rev. Lett. **122**, 060504 (2019).

[18] P. C. Costa, D. An, Y. R. Sanders, Y. Su, R. Babbush, and D. W. Berry, Optimal Scaling Quantum Linear-Systems Solver via Discrete Adiabatic Theorem, PRX Quantum **3**, 040303 (2022).

[19] D. An and L. Lin, Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm, ACM Trans. Quantum Comput. **3**, 1 (2022).

[20] N. Wiebe, D. Braun, and S. Lloyd, Quantum Algorithm for Data Fitting, Phys. Rev. Lett. **109**, 050505 (2012).

[21] P. Rebentrost and S. Lloyd, Quantum computational finance: Quantum algorithm for portfolio optimization, arXiv preprint ArXiv:1811.03975 (2018).

[22] I. Kerenidis, A. Prakash, and D. Szilágyi, in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies* (2019), p. 147.

[23] S. Palmer, S. Sahin, R. Hernandez, S. Mugel, and R. Orus, Quantum portfolio optimization with investment bands and target volatility, arXiv preprint ArXiv:2106.06735 (2021).

[24] S. Mugel, C. Kuchkovsky, E. Sanchez, S. Fernandez-Lorenzo, J. Luis-Hita, E. Lizaso, and R. Orus, Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks, Phys. Rev. Res. **4**, 013006 (2022).

[25] A. Domahidi, E. Chu, and S. Boyd, in *2013 European Control Conference (ECC)* (IEEE, 2013), p. 3071.

[26] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Quantum machine learning: A classical perspective, Proc. R. Soc. A: Math., Phys. Eng. Sci. **474**, 20170551 (2018).

[27] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, New J. Phys. **14**, 123011 (2012).

[28] D. Litinski and F. v. Oppen, Lattice surgery with a twist: Simplifying Clifford gates of surface codes, Quantum **2**, 62 (2018).

[29] D. Litinski, A game of surface codes: Large-scale quantum computing with lattice surgery, Quantum **3**, 128 (2019).

[30] C. Chamberland and E. T. Campbell, Universal Quantum Computing with Twist-Free and Temporally Encoded Lattice Surgery, PRX Quantum **3**, 010331 (2022).

[31] E. Knill, Fault-tolerant postselected quantum computation: Schemes (2004).

[32] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, Phys. Rev. A **71**, 022316 (2005).

[33] B. D. Clader, A. M. Dalzell, N. Stamatopoulos, G. Salton, M. Berta, and W. J. Zeng, Quantum resources required to block-encode a matrix of classical data (2022), ArXiv:2206.03505.

[34] Recently, an alternative method for pure-state tomography has been proposed in Ref. [73] with superior asymptotic query complexity, reducing $\mathcal{O}(L\ln(L)/\xi^2)$ to $\mathcal{O}(L\ln(L)/\xi)$. However, the protocol is more complicated than our approach and it requires additional gate overhead to implement. Furthermore, for the values of ξ and L that we consider in Table I, a conservative estimate of the improvement from this method (ignoring potentially large constants) only yields about a 2-orders-of-magnitude improvement in our final estimates of the T-depth and T-count—not enough to change our results qualitatively.

Thus, we do not incorporate this method into our analysis but we remark that we do expect a marginal improvement in our final counts.

[35] H. M. Markowitz, Portfolio selection, J. Finance **7**, 77 (1952).

[36] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investments* (John Wiley & Sons, New York, 1959).

[37] The Nobel Prize, Press release: This year's laureates are pioneers in the theory of financial economics and corporate finance (1990), https://www.nobelprize.org/prizes/economic-sciences/1990/press-release/.

[38] H. M. Markowitz, The early history of portfolio theory: 1600–1960, Financial Anal. J. **55**, 5 (1999).

[39] Typically, investment banks hold long positions, while hedge funds build portfolios with short positions that have higher risk due to the uncertainty of the price to buy the asset at the end of the period.

[40] For instance, we can add constraints to allow short positions, component-wise short sale limits, or a total short sale limit. Another variant of this is a constraint for a collateralization requirement, which limits the total of short positions to a fraction of the total long positions. Often, buying or selling an asset results in a transaction fee that is proportional to the amount of asset that is bought or sold. Linear transaction costs or maximum-transaction amounts are often included as constraints in PO. Diversification constraints can limit portfolio risk by limiting the exposure to individual positions and groups of assets within particular sectors.

[41] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, UK, 2004).

[42] S. J. Wright, *Primal-Dual Interior-Point Methods* (Society for Industrial and Applied Mathematics, Philadelphia, 1997).

[43] W. T. Ziemba and R. G. Vickson, *Stochastic Optimization Models in Finance, 2006 Edition* (World Scientific, Hackensack, NJ, 2006).

[44] G. Cornuéjols, J. Peña, and R. Tütüncü, *Optimization Methods in Finance* (Cambridge University Press, Cambridge, UK, 2018), 2nd ed.

[45] MOSEK ApS, *MOSEK Portfolio Optimization Cookbook Release 1.0.0* (2021).

[46] F. Alizadeh and D. Goldfarb, Second-order cone programming, Math. Program. **95**, 3 (2003).

[47] I. Kerenidis and A. Prakash, Quantum recommendation systems (2016), ArXiv:1603.08675.

[48] O. Ledoit and M. Wolf, The power of (non-)linear shrinking: A review and guide to covariance matrix estimation, J. Financial Econ. **20**, 187 (2020).

[49] Alternatively, the absolute-value constraints could be straightforwardly encoded with n second-order cone constraints of dimension 2; these formulations are equivalent up to a simple coordinate change, and we opt to use one-dimensional cones for their simplicity of presentation.

[50] Note that we would have had $r = 2n + 1$ cones if we had represented the absolute-value constraints using dimension-2 cones.

[51] Y. Ye, M. J. Todd, and S. Mizuno, An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm, Math. Oper. Res. **19**, 53 (1994).

[52] E. D. Andersen, C. Roos, and T. Terlaky, On implementing a primal-dual interior-point method for conic quadratic optimization, Math. Program. **95**, 249 (2003).

[53] R. D. Monteiro and T. Tsuchiya, Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions, Math. Program. **88**, 61 (2000).

[54] V. Strassen, *et al.*, Gaussian elimination is not optimal, Num. Math. **13**, 354 (1969).

[55] J. Alman and V. V. Williams, in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)* (SIAM, 2021), p. 522.

[56] T. Strohmer and R. Vershynin, A randomized Kaczmarz algorithm with exponential convergence, J. Fourier Anal. Appl. **15**, 262 (2009).

[57] Better asymptotic scaling for QR decomposition can be accomplished using fast matrix multiplication [89].

[58] A. Krishnamoorthy and D. Menon, in *2013 signal processing: Algorithms, architectures, arrangements, and applications (SPA)* (IEEE, 2013), p. 70.

[59] In this formulation, the quantum state $|\mathbf{v}\rangle$ corresponds to the normalized solution vector of the normalized linear system $G\mathbf{u} = \mathbf{h}$. Thus, the state $|\mathbf{v}\rangle$ does not carry information on the norm of the solution $\|\mathbf{u}\|$. This norm is related to $\mathbf{v}$ by the relationship $\|\mathbf{u}\| = \|\mathbf{h}\|/\|G\mathbf{v}\|$.

[60] A. Ambainis, Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations (2010), ArXiv:1010.4458.

[61] L. Lin and Y. Tong, Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems, Quantum **4**, 361 (2020).

[62] A. Dranov, J. Kellendonk, and R. Seiler, Discrete time adiabatic theorems for quantum mechanical systems, J. Math. Phys. **39**, 1340 (1998).

[63] Y. Dong, X. Meng, K. B. Whaley, and L. Lin, Efficient phase-factor evaluation in quantum signal processing, Phys. Rev. A **103**, 042419 (2021).

[64] Whereas the methods from Ref. [63, Sec. III] do not have a quantified worst-case convergence guarantee, they work very well in practice by typically running in time $\mathrm{polylog}(d\delta^{-1})$ for precision $\delta \in (0, 1]$ and $d = \mathcal{O}(\kappa_F(G)\log(1/\varepsilon_{\mathrm{QLSP}}))$, the degree of the underlying polynomial. Alternatively, one might resort to the provable methods from Ref. [76] that are known to run with complexity $\mathcal{O}(d^3\mathrm{polylog}(d\delta^{-1}))$ (for a discussion, see also Ref. [68]).

[65] G. H. Low, V. Kliuchnikov, and L. Schaeffer, Trading T-gates for dirty qubits in state preparation and unitary synthesis (2018), ArXiv:1812.00954.

[66] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum Random Access Memory, Phys. Rev. Lett. **100**, 160501 (2008).

[67] C. T. Hann, G. Lee, S. Girvin, and L. Jiang, Resilience of Quantum Random Access Memory to Generic Noise, PRX Quantum **2**, 020311 (2021).

[68] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019 (Association for Computing Machinery, New York, 2019), p. 193.

[69] S. Chakrabarti, A. M. Childs, T. Li, and X. Wu, Quantum algorithms and lower bounds for convex optimization, Quantum **4**, 221 (2020).

[70] In our setting, the matrices to block-encode are typically dense, which is why the general constructions from Ref. [33] are sufficient. However, in the event that the relevant data has some structure—e.g., if it is sparse—more adapted strategies such as Ref. [91] can be preferable.

[71] C. M. Dawson and M. A. Nielsen, The Solovay-Kitaev algorithm, arXiv preprint ArXiv:0505030 (2005).

[72] N. J. Ross and P. Selinger, Optimal ancilla-free Clifford+T approximation of z-rotations (2016), ArXiv:1403.2975.

[73] J. van Apeldoorn, A. Cornelissen, A. Gilyén, and G. Nannicini, Quantum tomography using state-preparation unitaries, arXiv preprint ArXiv:2207.08800 (2022).

[74] R. O'Donnell and J. Wright, in *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16 (Association for Computing Machinery, New York, 2016), p. 899.

[75] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, Quantum **3**, 163 (2019).

[76] J. Haah, Product decomposition of periodic functions in quantum signal processing, Quantum **3**, 190 (2019).

[77] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, Preconditioned Quantum Linear System Algorithm, Phys. Rev. Lett. **110**, 250504 (2013).

[78] It is perhaps related to whether the instance is nondegenerate and obeys strict complementarity. A known consequence of these properties is that the Jacobian matrix is nonsingular (implying a nondivergent condition number) at the optimum, as discussed in Ref. [46, Sec. 6].

[79] S. Chakraborty, A. Gilyén, and S. Jeffery, in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 132, edited by C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019), p. 33:1.

[80] Indeed, recently, Ref. [92] has proposed a different QLSS implementation based on Ref. [17] with slightly suboptimal $\mathcal{O}(\kappa \log(\kappa/\epsilon))$ asymptotic scaling but with superior actual performance for all $\kappa < 10^{32}$. According to Ref. [17, Fig. 1], substituting their QLSS for the one we have analyzed would reduce the contant prefactor at $\kappa = 10^4$ from our estimated $2 \times 1.31 \times 2305$ (where in this comparison we include the extra factor of 2 arising from the 1/2 success probability of eigenstate filtering) to about 900, saving about a factor of 6 on our resource estimate.

[81] M. Roetteler, M. Naehrig, K. M. Svore, and K. Lauter, in *International Conference on the Theory and Application of Cryptology and Information Security* (Springer, 2017), p. 241.

[82] J. Lee, D. W. Berry, C. Gidney, W. J. Huggins, J. R. McClean, N. Wiebe, and R. Babbush, Even More Efficient Quantum Computations of Chemistry Through Tensor Hypercontraction, PRX Quantum **2**, 030305 (2021).

[83] C. Jones, Low-overhead constructions for the fault-tolerant Toffoli gate, Phys. Rev. A **87**, 022328 (2013).

[84] K. C. Chen, W. Dai, C. Errando-Herranz, S. Lloyd, and D. Englund, Scalable and High-Fidelity Quantum Random Access Memory in Spin-Photon Networks, PRX Quantum **2**, 030319 (2021).

[85] N. Jiang, Y.-F. Pu, W. Chang, C. Li, S. Zhang, and L.-M. Duan, Experimental realization of 105-qubit random access quantum memory, npj Quantum Inf. **5**, 1 (2019).

[86] We expect that exploiting the sparsity of the matrix would lead to a reduced logical-qubit count and T-count but not a reduced T-depth. In fact, it could lead to non-negligible increases in the T-depth, since the shallowest block-encoding constructions from Ref. [33] are hyperoptimized for low depth and are explicitly not compatible with exploiting sparsity.

[87] Separate from the QLSS, a relatively small number of state-preparation queries is needed in tomography to create the state in Eq. (49) but this number does not scale with κ and we neglect it in this back-of-the-envelope analysis.

[88] R. Babbush, J. R. McClean, M. Newman, C. Gidney, S. Boixo, and H. Neven, Focus beyond Quadratic Speedups for Error-Corrected Quantum Advantage, PRX Quantum **2**, 010103 (2021).

[89] C. Camarero, Simple, fast and practicable algorithms for Cholesky, LU and QR decomposition using fast rectangular matrix multiplication, arXiv preprint ArXiv:1812.02056 (2018).

[90] P. A. Knight, Fast rectangular matrix multiplication and QR decomposition, Linear Algebra Appl. **221**, 69 (1995).

[91] D. Camps, L. Lin, R. V. Beeumen, and C. Yang, Explicit quantum circuits for block-encodings of certain sparse matrices (2022), ArXiv:2203.10236.

[92] D. Jennings, M. Lostaglio, S. Pallister, A. T. Sornborger, and Y. Subaşı, Efficient quantum linear solver algorithm with detailed running costs, arXiv preprint ArXiv:2305.11352 (2023).