





Constrained optimization via quantum Zeno dynamics

Dylan Herman ^{1,2}✉, Ruslan Shaydulin^{1,2}, Yue Sun ^{1,2}, Shouvanik Chakrabarti¹, Shaohan Hu¹, Pierre Minssen ¹, Arthur Rattew¹, Romina Yalovetzky ¹ & Marco Pistoia¹

Constrained optimization problems are ubiquitous in science and industry. Quantum algorithms have shown promise in solving optimization problems, yet none of the current algorithms can effectively handle arbitrary constraints. We introduce a technique that uses quantum Zeno dynamics to solve optimization problems with multiple arbitrary constraints, including inequalities. We show that the dynamics of quantum optimization can be efficiently restricted to the in-constraint subspace on a fault-tolerant quantum computer via repeated projective measurements, requiring only a small number of auxiliary qubits and no post-selection. Our technique has broad applicability, which we demonstrate by incorporating it into the quantum approximate optimization algorithm (QAOA) and variational quantum circuits for optimization. We evaluate our method numerically on portfolio optimization problems with multiple realistic constraints and observe better solution quality and higher in-constraint probability than state-of-the-art techniques. We implement a proof-of-concept demonstration of our method on the Quantinuum H1-2 quantum processor.

¹Global Technology Applied Research, JPMorgan Chase, New York, NY 10017, USA. ²These authors contributed equally: Dylan Herman, Ruslan Shaydulin, Yue Sun. ✉email: dylan.a.herman@jpmchase.com

Combinatorial optimization is widely considered to be one of the most promising problem domains for quantum algorithms. The ubiquity of hard optimization problems in science and industry amplifies the impact of any improvements in algorithmic performance. In practice, the optimization problems often have many constraints, such as the regulatory constraints when optimizing a portfolio or logistic constraints when optimizing flight crew assignments. Being able to incorporate a diverse range of constraints is an essential criterion for the applicability of a quantum algorithm to industrial problems.

A commonly considered class of quantum optimization algorithms uses a parameterized quantum evolution to drive the quantum system towards a state encoding the solution of the optimization problem. This class of algorithms includes the quantum approximate optimization algorithm (QAOA)^{1,2} and variational algorithms for optimization^{3,4}. While these algorithms are often discussed as promising approaches for noisy near-term devices^{5–9}. Therefore, in this paper we primarily view these algorithms as targeting fault-tolerant quantum processors.

One of the main challenges in applying these quantum algorithms to commercially-relevant optimization problems is the need to enforce the constraints. Concretely, the goal is to prepare a quantum state such that upon measuring it, a high-quality solution that satisfies the constraints is obtained with high probability. Two commonly considered approaches are to encode the constraint into the objective using a penalty term and to directly restrict the parameterized quantum evolution to the in-constraint subspace. In the first approach, a penalty term is added to the objective so that optimizing the objective requires satisfying the constraint. While such approaches are flexible enough to satisfy most constraints, the quality of the result is sensitive to the choice of the penalty strength¹⁰. As tuning the penalty strength is difficult in general, this approach often leads to sub-optimal performance in practice¹¹. This observation motivates the second approach, i.e., restricting the quantum evolution to the in-constraint subspace.

A number of techniques have been proposed to ensure that the parameterized quantum evolution respects the constraints of the problem. Hadfield et al.^{12,13} proposed the quantum alternating operator ansatz algorithm, which applies pairs of alternating operators to an in-constraint initial state. The first alternating operator (*phase operator*) is diagonal in the computational basis and encodes the objective, and the second operator (*mixing operator* or *mixer*) is non-diagonal and restricts the transitions of probability amplitudes to the computational basis states corresponding to the in-constraint solutions. The problem of constructing a Hamiltonian preserving arbitrary constraints is NP-complete even for linear constraints¹⁴, though explicit constructions are available for some combinatorial optimization problems^{12,13,15,16}. In general, constraint-preserving mixers are difficult to implement, even when constructions are available^{17,18}. The cost of implementing the algorithm on hardware can be reduced for a restricted class of problems by combining the phase and mixing operators¹⁹. If a uniform superposition of in-constraint states can be prepared efficiently, a Grover operator can be used as the mixer^{20–22}. Finally, for problems with an indexable set of feasible states (such as those with Hamming-weight constraints), a continuous-time quantum walk in the solution space can be used as a mixer^{23–25}. However, none of these techniques are sufficiently flexible to handle the general case of multiple arbitrary constraints directly. The parity optimization framework^{26–30} can natively handle polynomial equality constraints for QAOA-like circuits. However, this framework introduces an auxiliary qubit for every unique monomial term that appears, leading to large space overhead for complex objectives and constraints. All of the techniques mentioned above consider

QAOA-like alternating operator circuits, and are not easy to generalize to other variational algorithms.

In this work, we introduce an approach for enforcing multiple arbitrary constraints in quantum optimization. We restrict the quantum evolution to the in-constraint subspace by repeated projective measurements. In each measurement, the value of the constraint is computed onto an auxiliary register, which is then measured. Our technique uses quantum Zeno dynamics, wherein the evolution of the system is restricted to the subspace defined by the repeated projective measurements and transitions outside of this subspace are suppressed. Our approach is applicable to any problem in NPO (the NP optimization complexity class), as the only restriction we impose on the constraints is the existence of an efficient oracle for testing them. We provide explicit constructions for arbitrary combinatorial constraints. We demonstrate the effectiveness of the proposed technique by using it to enforce constraints in QAOA with various, unconstrained, mixing operators and the layer variational quantum eigensolver (L-VQE)³¹, which is a variational quantum algorithm for optimization. We show analytically that our technique is guaranteed to obtain the optimal in-constraint solution when applied to the digital simulation of the quantum adiabatic algorithm, or equivalently to QAOA in the constrained subspace with sufficiently large depth. We derive an analytical form of the scaling of the number of measurements required to maintain a constant minimum success probability for any parameterized quantum evolution. Furthermore, we provide numerical evidence that our technique, applied to QAOA for the portfolio optimization problem with a budget constraint, provides significant performance improvements over the state-of-the-art method of enforcing the constraint by introducing a penalty term. While the results we derive are for fault-tolerant quantum processors, high-fidelity near-term devices may be able to implement the algorithms without realizing full error-correction. To demonstrate an end-to-end realization of our technique, we implement QAOA with Zeno dynamics on the Quantinuum H1-2 trapped-ion quantum processor for proof-of-concept portfolio optimization problems. These experiments complement our numerical simulations by using explicit constructions and compilations of circuits, including those for checking the constraints. In the hardware experiments, we observe performance improvements from increasing the number of measurements, up to a two-qubit circuit depth of 148.

Results

Quantum Zeno dynamics for constrained optimization. We now introduce our approach to enforcing constraints in quantum optimization by repeated non-selective projective measurements. Our method is general, though here we focus on algorithms utilizing parameterized states of the form

$$|\psi(\theta)\rangle = U(\theta)|s\rangle = \prod_{j=1}^m e^{-i\theta_j H_j} |s\rangle, \quad (1)$$

where H_j is some Hamiltonian, e.g., a tensor product of single-qubit Pauli operators, and $|s\rangle$ is the initial state, which lies in the system Hilbert space \mathcal{H} .

A constrained combinatorial optimization problem has a set of feasible states \mathcal{F} , which is a subset of the n -dimensional Boolean cube \mathbb{B}^n . Let $P_{\mathcal{F}}$ denote the orthogonal projector onto the subspace spanned by computational basis states corresponding to feasible solutions in \mathcal{F} . We discuss the construction of this operator in the *Methods* Section. The measurement \mathcal{P} is a super-operator as defined as

$$\mathcal{P}\rho = \sum_{j=1}^k P_j \rho P_j, \quad (2)$$

where $\sum_{j=1}^k P_j = I$, and P_j is a projection onto some subspace $\mathcal{H}_j = P_j \mathcal{H}$ of dimensionality $\text{Tr}(P_j) \geq 1$. Without loss of generality, we can assume $P_1 = P_{\mathcal{F}}$, and define $P_{\mathcal{G}} := I - P_{\mathcal{F}} = \sum_{j=2}^k P_j$.

We give our main result in Theorem 1, which we use to derive the number of measurements required to enforce constraints in parameterized evolutions of the form given by Equation (1).

Theorem 1. Let \mathcal{P} be the measurement defined in Equation (2). Suppose a system is evolved from some initial state $\rho_0 = P_j \rho_0 P_j$ under the action of a Hamiltonian H , whose distinct eigenvalues are $\xi_{\min} = \xi_1 < \xi_2 < \dots < \xi_d = \xi_{\max}$ for time θ . For $\delta \leq 0.19$, if N applications of \mathcal{P} are performed at equally-spaced time intervals with

$$N = \left\lceil \frac{[\theta(\xi_{\max} - \xi_{\min})]^2}{\ln(1 - 2\delta)^{-2}} \right\rceil, \quad (3)$$

then the probability of measuring a state in \mathcal{H}_j at time θ is lower bounded by $1 - \delta$, i.e.,

$$\text{Tr}[P_j \rho(\theta)] \geq 1 - \delta, \quad (4)$$

where

$$\rho(\theta) = \mathcal{U}(\theta) \rho_0 \mathcal{U}(\theta)^\dagger, \quad \mathcal{U}(\theta) = [\mathcal{P} e^{-iH\theta/N}]^N. \quad (5)$$

Proof. See the *Methods* Section.

Remark 1. Note that since $2\|H\|_2 \geq |\xi_{\max} - \xi_{\min}|$, the bound can be reformulated in terms of the spectral norm of the Hamiltonian. This may be useful as the spectral norm may be easier to bound in practice for complicated Hamiltonians.

Assume that the initial state $|s\rangle$ respects the constraints, that is $P_{\mathcal{F}}|s\rangle = |s\rangle$. We apply a parameterized unitary $\mathcal{U}(\theta)$ to the initial state following Equation (1). To enforce the constraints, we can insert measurements into the parameterized evolution as follows:

$$\mathcal{U}_Z(\theta) = \prod_{k=1}^L \left[\mathcal{P} \prod_{j=1}^{m_k} e^{-i(\theta_{r(k,j)}/N_k) H_{r(k,j)}} \right]^{N_k}, \quad (6)$$

where $r(k, j) = \sum_{t=1}^{k-1} m_t + j$ and each sequence of m_k parameterized evolutions, without a measurement, is called a *block*. We define $N_k = 0$ to mean that no measurement is performed and no $\theta_{r(k,j)}$ is not scaled for that block. The following corollary provides a sufficient N_k for each block to ensure a desired minimum in-constraint probability. The asymptotic dynamics, i.e. when $N_k \rightarrow \infty, \forall k$ and also called the Zeno limit, will be different depending on how the blocks are chosen.

Corollary 1. Let \mathcal{P} be the measurement defined in Equation (1). Let the parameterized evolution defined in Equation (6) evolve the system from some initial state $\rho_0 = P_j \rho_0 P_j$. Then, in order to ensure that

$$\text{Tr}[P_j \mathcal{U}_Z(\theta) \rho_0 \mathcal{U}_Z(\theta)^\dagger] \geq 1 - \delta,$$

it suffices to choose

$$N_k = \left\lceil \frac{4L[\sum_{j=1}^{m_k} |\theta_{r(k,j)}|]^2 \max_j \|H_{r(k,j)}\|_2^2}{\tau(\delta)} \right\rceil, \quad (7)$$

where

- $\tau(\delta) = \ln(1 - 2\delta)^{-2}$ if $H_{r(k,j)}$ pairwise commute,
- $\tau(\delta) = \ln(1 - \delta)^{-1.78}$ otherwise,

and $\delta \leq 0.19$. In addition, the asymptotic dynamics is

$$\prod_{k=1}^L e^{-iP H_k \cdot \theta_k P}, \quad (8)$$

where \mathcal{P} acts element-wise on the vector $H_k = (H_{(k,1)}, \dots, H_{(k,m_k)})^\top$ and $\theta_k = (\theta_{(k,1)}, \dots, \theta_{(k,m_k)})$.

Proof. See the *Methods* Section.

Remark 2. For combinatorial optimization problems, constraint-preserving measurements that correspond to different constraints always commute. Thus $\mathcal{P}_{\mathcal{F}}$ can be implemented as a composition of measurements corresponding to different constraints.

While the previous results indicate that N_k can grow inverse polynomially with the desired error probability, the following result (Corollary 2) shows that fixing δ and applying a simple repetition scheme suffices to suppress the failure probability arbitrarily below δ with only logarithmic overhead. Thus, the overall procedure can be made efficient. The purpose of the Zeno framework is to ensure that we can obtain a state that has an overlap with \mathcal{H}_j that is lower bounded by a constant and prepare this state with an overhead that is $O(\text{polylog}(\dim \mathcal{H}))$.

Corollary 2. Let \mathcal{P} be the measurement defined in Equation (2). Let the parameterized evolution defined in Equation (6) evolve the system from some initial state $\rho_0 = P_j \rho_0 P_j$. In addition, suppose that the number of measurements N_k was chosen, using Corollary 1, to ensure that $\text{Tr}[P_j \rho_Z(\theta)] = \text{Tr}[P_j \mathcal{U}_Z(\theta) \rho_0 \mathcal{U}_Z(\theta)^\dagger]$ is lower bounded by a constant independent of the system size, and then in order to ensure that \mathcal{P} applied to $\rho_Z(\theta)$ prepares a state in \mathcal{H}_j with a probability at least $1 - \epsilon$, it suffices to prepare and measure at most $\log(1/\epsilon)$ copies of $\rho_Z(\theta)$.

Proof. Suppose $\text{Tr}[P_j \rho_Z(\theta)] = c$. Since we can efficiently check whether the post-measurement state obtained from applying \mathcal{P} to $\rho_Z(\theta)$ is in \mathcal{H}_j , $\log(1/\epsilon)/\log(1/(1-c)) < \log(1/\epsilon)$ repetitions suffice to ensure that the outcome of at least one of the repetitions is in \mathcal{H}_j with probability at least $1 - \epsilon$.

These results imply that for most practical cases, e.g. when H_j are Pauli operators as in the cases of QAOA and hardware-efficient parameterized circuits, the number of measurements scales at most quadratically in the circuit depth and width, i.e., as $O(\text{polylog}(\dim \mathcal{H}))$. Thus, QZD can be used to efficiently constrain parameterized evolution for quantum optimization.

Constrained QAOA via Zeno dynamics. We now discuss the application of QZD to QAOA. In a QAOA circuit, the phase operator $U_C(\gamma)$ is diagonal in the computational basis and cannot violate constraints. More specifically, it evolves the current state, for time γ , under the diagonal operator $C = \sum_{\mathbf{x} \in \mathbb{B}^n} f(\mathbf{x}) |\mathbf{x}\rangle \langle \mathbf{x}|$, which encodes the values of the objective function f on \mathbb{B}^n . The Hermitian mixing operator B transitions probability amplitude between elements of \mathbb{B}^n and, in general, does not respect the problem constraints. Therefore the measurements only need to be added to the mixing operator. Since a p -layer QAOA circuit consists of p applications of the phase and mixing operators in an alternating fashion, the full circuit combined with the Zeno framework then becomes

$$\mathcal{U}_{Z\text{-QAOA}}(\beta, \gamma) = \prod_{j=1}^p [\mathcal{U}_B(\beta_j, N_j) U_C(\gamma_j)], \quad (9)$$

where

$$\mathcal{U}_B(\beta_j, N_j) = \left[\mathcal{P} e^{-i\beta_j B/N_j} \right]^{N_j}. \quad (10)$$

In the notation of Equation (6), this corresponds to setting all $m_k = 1$, and setting $N_k = 0$ for blocks containing the cost operator. While there are other valid choices for the blocks, the

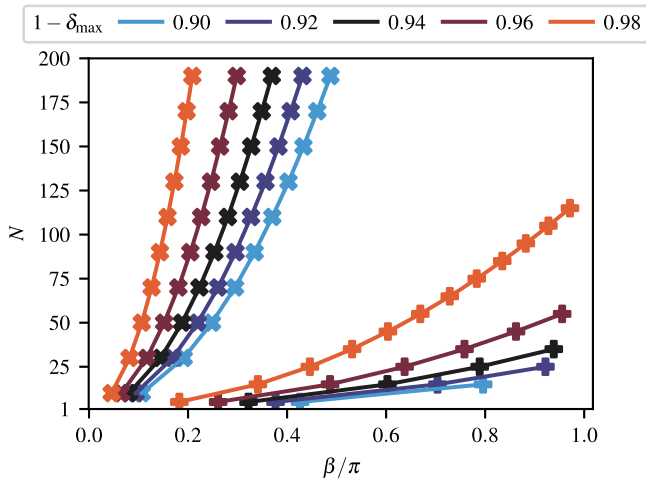


Fig. 1 Scaling of the number of Zeno measurements. Number of measurements, obtained from Corollary 3, required in QAOA with Zeno dynamics to maintain a maximum out-of-constraint probability of δ_{\max} (hence, a minimum in-constraint probability of $1 - \delta_{\max}$) for the $B = \sum_j x_j$ (\times marker) and $B = |+\rangle\langle+|$ ($+$ marker) mixers with 3 qubits. Color denotes the minimum in-constraint probability $1 - \delta_{\max}$, as indicated by the legend. Note that this is the scaling required to ensure the desired minimum in-constraint probability for the worst-case initial state (i.e., Equation (37)) and is potentially more pessimistic than what is observed in practice. Many more measurements are required for $B = \sum_j x_j$ as the number of measurements grows quadratically with number of qubits. Note that due to periodicity, the evolution time, β , can be constrained to $|\beta| \leq \frac{\pi}{2}$ for $B = \sum_j x_j$ and $|\beta| \leq \pi$ for $B = |+\rangle\langle+|$.

decomposition we have chosen is sufficient to achieve an efficient scheme.

As the mixing operator B is known, we can explicitly derive the number of measurements required to maintain a constant success probability. We observe that for any mixer this number of measurements grows linearly with the number of QAOA layers, and for commonly considered mixers, the number of measurements grows no more than quadratically with the number of qubits.

Corollary 3. Let $\mathcal{U}_{\text{Z-QAOA}}(\beta, \gamma)$ denote the QAOA circuit on n qubits with N measurements added to each mixing operator as defined in Equation (9). Let the initial state $\rho_0 = |s\rangle\langle s|$ be in-constraint. Then N_j measurements suffice to maintain at least a $1 - \delta$ probability of obtaining an in-constraint measurement outcome, where

- if $B = \sum_{k=1}^n x_k$, then $N_j = \left\lceil \frac{p\beta_j^2 n^2}{\ln[1-2\delta]^{-\frac{1}{2}}} \right\rceil$
- if $B = |+\rangle\langle+|$, then $N_j = \left\lceil \frac{p\beta_j^2}{\ln[1-2\delta]^{-\frac{1}{2}}} \right\rceil$,

and $\delta \leq 0.19$.

Proof. The proof follows from Theorem 1 by noting that for $B = \sum_{k=1}^n x_k$ the minimum and maximum eigenvalues are $-n$ and n , respectively, and for $B = |+\rangle\langle+|$ the only eigenvalues are one and zero. For QAOA with p layers, the number of measurements increases by a factor of p . Note that while we could of instead used Corollary 1, using Theorem 1 directly results in N_k being lower by a constant for $B = |+\rangle\langle+|$.

Note that the scaling rule of Corollary 3 implies that the number of measurements will change with β_j and thus each mixer layer.

Figure 1 visualizes how the number of measurements required to maintain a given minimum in-constraint probability, according to Corollary 3, grows with the evolution time β for the

$B = \sum_j x_j$ (\times marker) and $B = |+\rangle\langle+|$ ($+$ marker) mixing operators for $p = 1$ QAOA with a 3-qubit initial state $|s\rangle$. As the phase operator is diagonal, there is no dependency on it. We note that the number of measurements for the mixer $B = \sum_j x_j$ grows with number of qubits and is therefore larger than for $B = |+\rangle\langle+|$. Note that when following the scaling rules of Corollary 3, the number of measurements is multiplied by the number of QAOA layers p .

In the *Results* Section, we observe that for realistic constraints, the number of measurements is significantly lower. This is because the worst-case $P_{\mathcal{F}}$ and $|s\rangle$, i.e., from Equation (37) in the proof of Lemma 1, are far from those encountered in practice. Specifically, the worst-case $P_{\mathcal{F}}$ is rank one (i.e., only one state is in-constraint). A larger in-constraint subspace leads to a lower sufficient number of measurements. Moreover, in practice the initial state is unlikely to align perfectly with the worst case presented in Equation (37). We also observe in our experiments that the required number of measurements has only a weak dependence on the number of QAOA layers p for the problem instances considered. Therefore, one could consider a significantly relaxed and simplified version of the rules provided in Corollary 3 as follows:

$$N_j = \left\lceil \frac{\beta_j^2}{\eta} \right\rceil, \quad (11)$$

where η is some hyperparameter to be fine tuned. One could always efficiently estimate the in-constraint probability of a QAOA circuit with a fixed η by measuring a single auxiliary qubit indicating whether the final state output by the circuit is in-constraint. In the portfolio optimization experiments, we successfully use an η for the $B = \sum_j x_j$ mixer that is orders of magnitude larger than predicted by Corollary 3, requiring a correspondingly smaller number of measurements.

QAOA with Zeno dynamics in the adiabatic limit. If the initial state $|s\rangle$ is the ground state of the mixer Hamiltonian B , QAOA is known to be able to prepare the ground state of the cost Hamiltonian C and thereby solve the problem exactly in the limit of an infinite number of QAOA layers by approximating adiabatic evolution². We now show that this limiting behavior is preserved for constrained QAOA with Zeno dynamics.

Now consider QAOA with constraints enforced by measurement \mathcal{P} as defined in Equation (2), in the Zeno limit, when the number of measurements is taken to infinity, the operator describing the asymptotic dynamics is a sum of the original mixer B projected onto the subspaces defined by the projectors constituting \mathcal{P} , i.e.,

$$H_Z = \mathcal{P}B = \sum_{j=1}^k P_j B P_j.$$

Concretely, consider the task of using QAOA to approximate the adiabatic evolution under the following time-dependent Hamiltonian:

$$H_s(t) = (1 - s(t))B + s(t)C, \quad (12)$$

where $s: [0, T] \rightarrow [0, 1]$ is the interpolating schedule function. A common schedule function is the linear schedule defined by

$$s(t) = \frac{t}{T}, \quad (13)$$

where T is the evolution time scale. Suppose $T \gg O((\min_s \Delta_n(s))^{-2})$, where $\Delta_n(s)$ is the instantaneous minimum difference between the n -th eigenvalue and any other eigenvalue of $H(s)$. If $\forall s$, it holds that $\Delta_n(s) \neq 0$, then the quantum

adiabatic theorem³² implies:

$$\mathcal{T} \exp\left(i \int_0^T H_s(t) dt\right) |\phi_n(0)\rangle = |\phi_n(T)\rangle. \quad (14)$$

In the Zeno case, we consider

$$H_s(t) = (1 - s(t))H_Z + s(t)PC. \quad (15)$$

Consider the QAOA operator with only one measurement per layer, i.e., $\forall j, N_j = 1$ in (9):

$$\mathcal{U}(p) = \prod_{j=1}^p \mathcal{P} \mathcal{U}_B(\beta_j) \mathcal{U}_C(\gamma_j). \quad (16)$$

Now it is easy to recover the parameters β_j, γ_j giving the limit. From the definition of the product integral³³ it follows that

$$\begin{aligned} & \mathcal{T} \exp\left(i \int_0^T H_s(t) dt\right) \\ &= \lim_{p \rightarrow \infty} \prod_{j=1}^p \exp\left(i \frac{T}{p} H_s\left(\frac{jT}{p}\right)\right) \\ &= \lim_{p \rightarrow \infty} \prod_{j=1}^p \exp\left(i \frac{T}{p} \left[\left(1 - \frac{j}{p}\right) \mathcal{P} B + \left(\frac{j}{p}\right) \mathcal{P} C\right]\right) \\ &= \lim_{p \rightarrow \infty} \prod_{j=1}^p \mathcal{P} \exp\left(i \frac{T}{p} \left(1 - \frac{j}{p}\right) B\right) \exp\left(i \frac{jT}{p^2} C\right), \end{aligned} \quad (17)$$

where the third equality follows from expanding to the first order in $\frac{T}{p}$ and that $\frac{j}{p}$ and $1 - \frac{j}{p}$ are bounded by 1. Also, since the evolution is in a finite-dimensional space, B and C have bounded operator norms.

Thus if $\rho_n(0) = |\psi_n(0)\rangle\langle\psi_n(0)|$ is an n -th eigenstate of H_Z then

$$\rho_n(T) = \lim_{p \rightarrow \infty} \mathcal{U}(p) \rho_n(0), \quad (18)$$

where $\rho_n(T)$ is pure and is an n -th eigenstate of $\mathcal{P}C$. Thus with $\beta_j = -\frac{T}{p}(1 - \frac{j}{p})$ and $\gamma_j = -\frac{jT}{p^2}$ as $p \rightarrow \infty$, QAOA with Zeno dynamics approaches the adiabatic limit and recovers the optimal solution.

Mitigating mixer limitations in the Zeno limit. While the evolution under $\mathcal{P}_{\mathcal{F}} B \mathcal{P}_{\mathcal{F}}$ is guaranteed to preserve the in-constraint subspace, it may inhibit transitions between states in \mathcal{F} that were allowed with B . This is because states in \mathcal{F} may be connected by B through a path that passes through states not in \mathcal{F} . To see this, consider a simple example of the two-qubit mixer $B_2 = x_1 + x_2$ and the in-constraint space $\mathcal{F} = \{|01\rangle, |10\rangle\}$. In the Zeno limit, the mixing operator evolution in the in-constraint subspace is generated by $\mathcal{P}_{\mathcal{F}} B_2 \mathcal{P}_{\mathcal{F}}$, which equals the zero matrix. Thus, the propagator corresponding to the projected mixer becomes the identity operator and the dynamics become trivial. In general, if there is no path between two computational basis states $|j\rangle, |k\rangle \in \mathcal{F}$ in the graph defined by B , the continuous-time quantum walk defined by the mixing operator cannot move probability amplitude from $|k\rangle$ to $|j\rangle$. Whether the transitions between in-constraint states are suppressed in the Zeno limit is in general dependent on the in-constraint space \mathcal{F} .

One way to avoid the issue of suppressed transitions is by choosing a mixer B with a complete connectivity graph among computational basis states, i.e., $B = |+\rangle\langle+|$. This mixer is also known as the complete-graph mixer^{20,34}. It has been conjectured³⁴ that mixers with high connectivity, such as the $B = |+\rangle\langle+|$, can at best produce a Grover-like speedup since they do not make use of the structure of the cost operator. While it is unclear if this conjecture is true, we emphasize that our approach

can utilize any mixer and can efficiently enforce constraints as long as the difference between the maximum and minimum eigenvalues of the mixer is polynomial in the number of qubits.

Numerical experiments. We now present the numerical experiments showing the power of the proposed method. The technique we propose is general, though in this section we consider only the problem of portfolio optimization (with both equality and inequality constraints) and only the QAOA and L-VQE algorithms. The parameters in QAOA and VQE were optimized using COBYLA³⁵ initialized with a large number of random initial points. We compare the results to the state-of-the-art method of encoding constraints by introducing a penalty into the objective, and observe significant improvements in both approximation ratio and in-constraint probability. In addition to better performance, the proposed method does not require complicated tuning of the penalty factor.

Benchmark: portfolio optimization. The daily operation of a large financial institution requires solving many classically-hard optimization problems^{36–38}. Among such problems, one of the most important is portfolio optimization. Modern portfolio theory³⁹ considers the task of finding a portfolio with a desired trade-off between risk and expected return. This task is typically formulated as an optimization problem, which is hard to solve classically in many settings, such as when the variables are required to only take on a discrete set of values. When designing an algorithm for portfolio optimization, a central consideration is the ability to incorporate a general class of constraints. Such constraints can come from regulatory or business considerations, with examples ranging from portfolio-level constraints (including budget and total number of assets) to asset-level constraints (such as minimum holding size).

The particular constrained portfolio optimization problems we study numerically arise from the discrete mean-variance Markowitz model³⁹ and have the following objective function

$$\min_{\mathbf{x} \in \mathcal{F}} q \mathbf{x}^T \Sigma \mathbf{x} - \boldsymbol{\mu}^T \mathbf{x}, \quad (19)$$

where \mathcal{F} is defined by some set of constraints on the portfolio. We consider two sets of problems. In the first set, we impose an inequality constraint on the total size of the portfolio ($\sum_j x_j \leq C$). In the second set of problems, in addition to the inequality constraint on portfolio size, we include a constraint on the total expected return ($\sum_j \mu_j x_j \geq R$). For each of the two sets of constraints, we consider seven instances with between four and ten assets, for a total of fourteen instances. In all problem instances $\mathcal{F} \subset \mathbb{B}^n$, where n is the number of assets.

Zeno dynamics improves quantum optimization performance. Figure 2 presents the comparison between QAOA with Zeno dynamics and QAOA with constraints enforced using a penalty factor on the fourteen problem instances described in the previous subsection. The penalty method is described in the *Methods* Section. The solution quality is measured in terms of the approximation ratio r , a value between 0 and 1, with larger r being better. The approximation ratio is formally defined in the *Methods* Section. We consider QAOA with mixers $B = \sum_j x_j$ (\star marker) and $B = |+\rangle\langle+|$ (\dagger marker), and optimize the QAOA parameters exhaustively. To improve the performance of parameter optimization, we follow ref. ⁴⁰ and rescale the cost function so that the gradients with respect to β and γ are roughly of the same magnitude.

For instances with a single constraint (see dotted lines in Fig. 2a–d) we perform extensive tuning of the penalty factor λ . For multi-constraint problems, the tuning becomes prohibitively

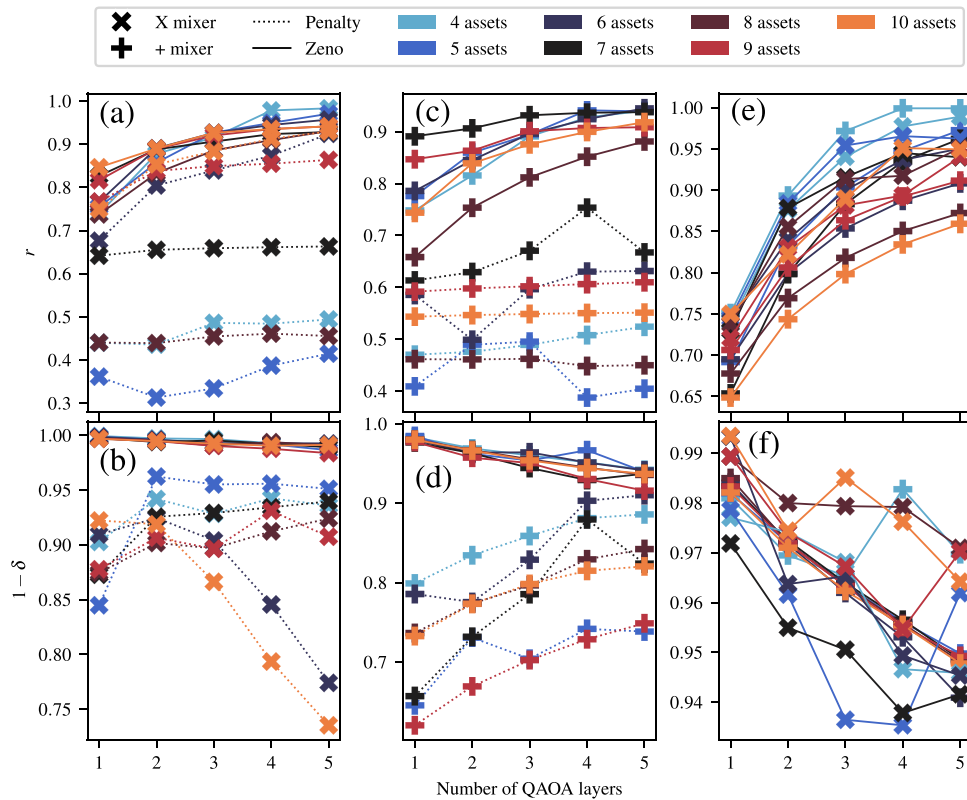


Fig. 2 Performance of QAOA with Zeno dynamics and QAOA with constraints enforced using penalty terms. Approximation ratio r and out-of-constraint probability δ (correspondingly $1 - \delta$ in-constraint probability) achieved by QAOA with constraints enforced using penalty terms (dotted lines) on problems (a–d) with a single constraint, and by QAOA with Zeno dynamics (solid lines) on problems with a single (a–d) and multiple (e, f) constraint(s). The markers \times and $+$ indicate whether QAOA used the $B = \sum_j x_j$ mixer or $B = |+\rangle\langle+|$ mixer, respectively. For all single constraint problems, QAOA with Zeno dynamics produces a superior approximation ratio and in-constraint probability (solid line is above dotted line with the same color). As penalty factor tuning is prohibitively difficult for problems with multiple constraints (see the Results Section), for these problems only Zeno dynamics results are presented.

expensive. Therefore, we exclude QAOA with constraints enforced through penalties from the comparison for problems with multiple constraints. The choice of the penalty factor and the difficulty of its optimization are discussed in detail in the next subsection.

We observe that Zeno dynamics (see solid lines in Fig. 2a–d) enables consistently better solution quality and in-constraint probability as compared to QAOA with constraints enforced using a penalty (dotted lines) for all problems considered. Furthermore, Fig. 2b shows that for 6 and 10 assets the in-constraint probability drops off rapidly with the number of QAOA layers if the penalty factor is kept constant. This highlights an important limitation of enforcing the constraints via penalties, namely that the penalty factor must be tuned independently for each QAOA depth. In contrast, for QAOA with Zeno dynamics we obtain an explicit rule for how η , from (11), should change with the QAOA depth (see Corollary 3). However, for the numerics shown in Fig. 2, we fix η to ensure a constant minimum in-constraint probability per layer. We observe good performance despite η being a depth-independent constant in this case. We note that since η was held constant while p varied, the in-constraint probability slowly decreases with the number of layers as predicted by Corollary 3. For $B = |+\rangle\langle+|$ mixer, this results in an average number of measurements of ≈ 77 for 6 assets and ≈ 35 for 7 assets.

Since multiple constraints can be efficiently handled in the Zeno framework, in Fig. 2e, f, we include the performance of QAOA with Zeno dynamics on problems with multiple

constraints (one on the budget and one on the total expected return). The results show that the Zeno-enhanced QAOA is able to achieve a similar performance as it did for the single-constraint problems, with sufficiently high p .

We note that the in-constraint probability can be improved arbitrarily for the Zeno dynamics approach by decreasing η , without the need to re-optimize the QAOA parameters. This is due to the objective function landscape becoming independent of η as the Zeno limit is approached. In fact, we observe that transferring parameters from a smaller to a larger number of measurements (larger to smaller η) works well even for practically relevant values of η . Figure 3 shows the approximation ratio r and in-constraint probability with directly optimized QAOA parameters and with pre-optimized parameters transferred from a fixed value of $\eta = 1.6$ (marked with a star in the plot). We observe that for sufficiently small η , transfer works well and the difference in approximation ratio is negligible. Specifically, parameter transfer using the $B = \sum_j x_j$ mixer and a total of 33, 75, and 200 measurements results in in-constraint probabilities of at least 85%, 89%, and 96%, respectively for the nine-assets, single-constraint problem at $p = 5$. At the same time, if the number of measurements is very small (η large), the objective function landscape is very different from the landscape in the Zeno limit, and the parameter transfer does not work well. We remark that while the in-constraint probability increases monotonically as η decreases, no such guarantee is given for approximation ratio r . In fact, in Fig. 3 we observe that depending on the problem and the circuit depth, r can either increase or decrease with η .

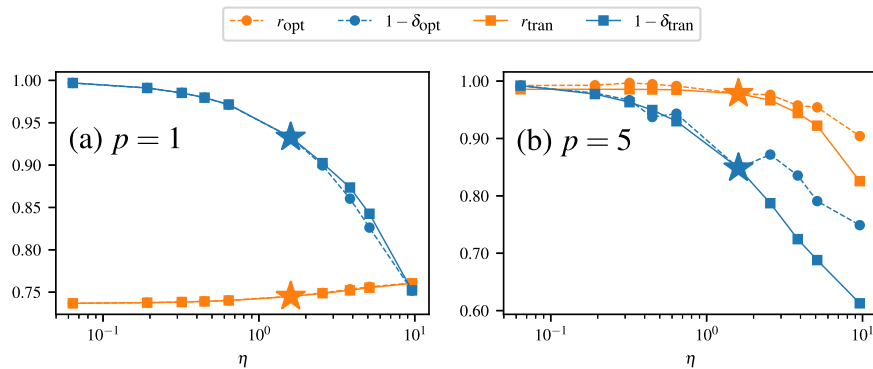


Fig. 3 Transferability of parameters in QAOA with Zeno dynamics. Performance of a 1-layer (a) and 5-layer (b) QAOA with Zeno dynamics and mixer $B = \sum_j x_j$ with directly optimized parameters (r_{opt} , $1 - \delta_{\text{opt}}$) and with parameters transferred from a fixed value of $\eta = 1.6$ (r_{tran} , $1 - \delta_{\text{tran}}$). The source is marked with a star. Corresponding to each case, r signifies the approximation ratio and δ the out-of-constraint probability. For values of the hyperparameter η , which controls the number of measurements and is defined in Equation (11), smaller than 1.6, the difference between performance with optimized and transferred parameters is negligible (dashed line very close to the solid line).

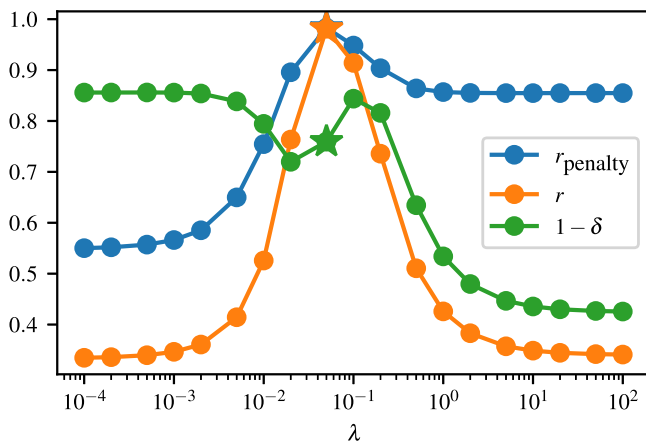


Fig. 4 Transferability of parameters in QAOA with penalty terms. Performance of QAOA with $B = \sum_j x_j$ mixer and constraints enforced through penalties with parameters transferred from a fixed value of penalty factor $\lambda = 0.1$ (source marked with a star). The out-of-constraint probability is δ . The approximation ratio r (Equation (23)), unlike r_{penalty} (Equation (24)), excludes the penalty objective and drops off to random guess if transferring parameters to values of λ sufficiently different from source.

Note that the same approach of boosting the in-constraint probability without re-optimizing the QAOA parameters does not work if the constraints are enforced using penalties. Figure 4 shows that transferring parameters from a fixed value of penalty factor (marked with a star) leads to the approximation ratio rapidly dropping off to random guess. It is however possible that better performance may be achieved by leveraging more sophisticated parameter transfer strategies, such as the rescaling rule proposed for the weighted MaxCut problem^{41,42} or machine learning methods⁴³.

While for QAOA with Zeno dynamics the approximation ratio r given in Equation (23) increases monotonically with the number of QAOA layers, this is not guaranteed for QAOA with constraints enforced through penalties. This is because the QAOA parameters are chosen with respect to the objective with penalties and the increased expressivity of the higher-depth circuit is only guaranteed to improve the performance with respect to that objective. Figure 5 shows that this is indeed the case and the approximation ratio r_{penalty} given in Equation (24) increases with the number of QAOA layers as expected.

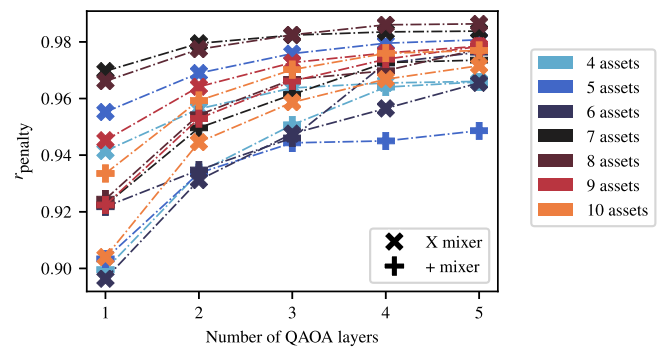


Fig. 5 Approximation ratio of QAOA with penalty terms using different numbers of QAOA layers. The approximation ratio (as defined in Equation (24)) for the full objective with penalty terms increases monotonically with the number of QAOA layers, as expected. However, the in-constraint approximation ratio (as defined in Equation (23)) is not guaranteed to change monotonically, as seen in Fig. 2a, c. Color denotes the number of assets in the optimization problem, as shown in the legend. The markers \times and $+$ indicate whether QAOA used the $B = \sum_j x_j$ mixer or $B = |+ \rangle \langle +|$ mixer, respectively.

Finally, we include the results for Zeno-enhanced L-VQE with $L = 1$ in Equation (6). The structure of L-VQE is presented in Equation (26) and further described in the *Methods* Section. However, instead of using Corollary 1 to determine a sufficient value for the number of measurements N , we heuristically set $N = 100$. Table 1 presents the results. As expected, L-VQE achieves high approximation ratio, while Zeno dynamics enables high in-constraint probability. As the total number of measurements is kept fixed for all problems and parameter values, slightly lower in-constraint probability is observed for higher qubit counts. As is the case for QAOA, the in-constraint probability can be increased by increasing the number of measurements.

Penalty factor tuning is difficult. An important advantage of our method is the simplicity of hyperparameter tuning, as only η in Equation (11) needs to be chosen. This choice is made easy by Theorem 1 and its corollaries, which imply the monotonic increase of in-constraint probability with decrease in η . This is in sharp contrast with the penalty approach, where the performance crucially depends on the penalty strength, which is hard to tune in general. We now present how the penalty strength was chosen

for the experiments above, and discuss the challenges that arose in doing so.

Figure 6 presents the performance of QAOA on a single-constraint problem enforced using a penalty term with varying penalty factors λ . In the plot, the in-constraint probability $1 - \delta$ monotonically increases with λ , while the approximation ratio r decreases. This indicates a trade-off between r and the out-of-constraint probability δ , and hence hyperparameter tuning on λ must be performed in order to obtain a good approximation ratio while meeting requirements on the minimum in-constraint probability. We also observe that for QAOA with small p , $1 - \delta$ tends to levels off at a value far below what is achievable by using Zeno dynamics. For example, the top figure in Fig. 6 shows that the highest in-constraint probability achievable with $p = 1$ is around 80% for the problem tested. Given that the approximation ratio with the penalty term r_{penalty} is above 0.9 for the high λ regime, it indicates that the maximum achievable in-constraint probability may be limited by the expressivity of the variational circuit. On the other hand, constraints enforced by Zeno dynamics do not suffer from such problems, as the in-constraint probability can be arbitrarily boosted regardless of the expressivity of the variational circuit (see Fig. 3). In the numerical experiments, we choose the value of λ independently for each problem instance with the goal of obtaining a high in-constraint probability $1 - \delta$. Since we show that the factor λ

trades off r and δ , both cannot be improved at the same time. This suggests that there does not exist a choice of λ such that QAOA with the penalty method outperforms QAOA with Zeno dynamics.

For problems with multiple constraints, hyperparameter tuning should generally be performed on each penalty factor λ_i included in the relaxed objective (Equation (21)). This means that hyperparameter tuning can quickly become infeasible, as the search space for all λ_i 's grows exponentially with the number of penalty terms. We show in Fig. 7 how hyperparameter tuning works with two penalty factors: λ_1 and λ_2 , which correspond to penalty terms enforcing the budget constraint and the return constraint respectively. The figure shows the in-constraint probability of the optimal solution obtained with varying λ_1 and λ_2 . Similar to the single-constraint case, maximal approximation ratio r and maximal in-constraint probability $1 - \delta$ cannot be simultaneously achieved. Specifically, the solutions with the maximal r and maximal $1 - \delta$ have very different values in λ_1 and λ_2 . Moreover, unlike Fig. 6, Fig. 7 clearly shows the non-monotonic behavior of $1 - \delta$ in both λ_1 and λ_2 . In fact, we observe a similar behavior across many of the single- and multi-constraint problems that we have tested, and for both the $B = \sum x_j$ and $B = |+\rangle\langle +|$ mixers. This indicates that tuning the penalty factors is indeed difficult in the general case.

Table 1 Performance of L-VQE with Zeno dynamics on the benchmark problems.

# assets	Single		Multiple	
	r	$1 - \delta$	r	$1 - \delta$
4	0.995	0.964	0.9996	0.980
5	0.995	0.913	0.977	0.909
6	0.972	0.895	0.964	0.963
7	0.979	0.870	0.917	0.936
8	0.956	0.887	0.948	0.944
9	0.967	0.844	0.961	0.974
10	0.914	0.811	0.910	0.960

Layer variational quantum eigensolver (L-VQE) enhanced with Zeno dynamics obtains high approximation ratio r (Equation (23)) and high in-constraint probability $1 - \delta$. The algorithm was applied to both problems with a single constraint and multiple constraints. The Zeno-enhanced L-VQE circuit was constructed by inserting measurements after all parameterized gates have been applied. This corresponds to $L = 1$ in Equation (6). The number of measurements was heuristically set to 100.

Hardware experiments. While the numerical experiments presented earlier show evidence of the performance of our technique, they do not make use of any concrete circuit implementations of the constraint-checking oracles. In this section, we consider optimized circuit implementations of constraint-checking oracles for two proof-of-concept portfolio optimization problems on noisy quantum hardware. This enables us to validate all of the hardware features, such as mid-circuit measurements and quantum conditional logic (QCL), that are required to implement the efficient oracle construction presented in the *Methods* Section.

We execute QAOA with Zeno dynamics on the Quantinuum H1-2 trapped-ion quantum processor. Our implementation uses constraint-checking oracles that perform quantum arithmetic in the Fourier domain, following directly the construction in the *Methods* Section. We observe that increasing the number of measurements improves the in-constraint probability $1 - \delta$, as expected. The improvement from additional measurements continues up to a two-qubit gate depth of 148, at which point the hardware noise prevents further improvements.

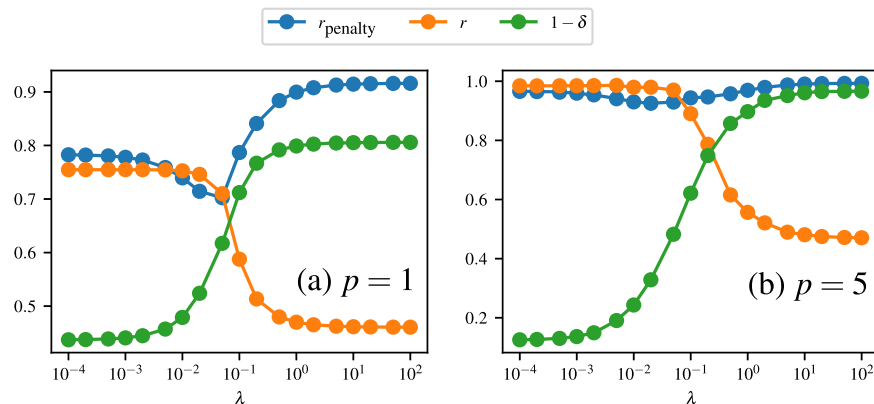


Fig. 6 Difficulty of penalty factor tuning for QAOA with a single penalty term. Performance of QAOA with a single constraint enforced through a penalty term with varying penalty factors λ . A trade-off occurs between the approximation ratio r (Equation (23)) and the in-constraint probability $1 - \delta$. As shown in **a** the maximum in-constraint probability is limited by the expressivity of the QAOA circuit at low depth (1 QAOA layer, or $p = 1$). With 5 layers (**b**), QAOA is able to achieve better performance in terms of the penalized objective, as indicated by the approximation ratio r_{penalty} (Equation (24)). However, there is still a significant trade-off between the true objective r and in-constraint probability.

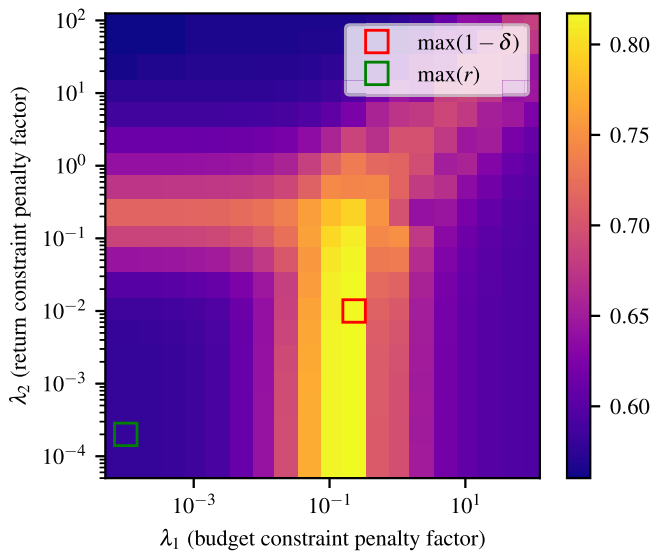


Fig. 7 Difficulty of penalty factor tuning for QAOA with two penalty terms. In-constraint probability of optimized solution using QAOA applied to an objective with two penalty functions, associated with separate constraints. The corresponding penalty factors are indicated by λ_1 and λ_2 , respectively. One is a maximum budget constraint and the other is minimum return constraint. The value δ is the out-of-constraint probability, and r is the approximation ratio (as defined in Equation (23)). The square highlighted in red corresponds to the maximum in-constraint probability ($1 - \delta$) over all combinations of the two penalty factors, and the square highlighted in green corresponds to the maximum r . This highlights that both large in-constraint probability and large approximation ratio cannot be obtained. The figure shows results for the $B = |+\rangle\langle +|$ mixer and 3-layer QAOA, though we observe similar behavior for all mixers and QAOA depths considered.

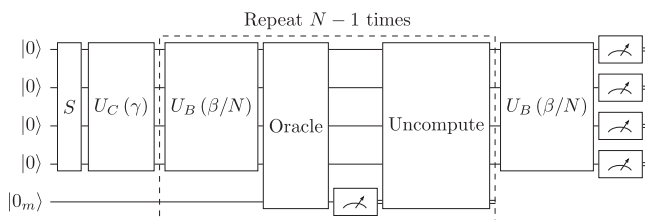


Fig. 8 Quantum circuit for QAOA with Zeno dynamics. QAOA circuit with Zeno dynamics used in hardware runs for one-layer QAOA ($p = 1$) on four-asset problems. The operator S prepares a uniform superposition over feasible states.

The experiments presented in this Section utilize $p = 1$ QAOA and the $B = \sum_j x_j$ mixer. We use the cost function of the four-assets portfolio optimization problem used in the numerics described in the *Results* Section, but apply different constraints. We consider two instances with linear constraints, one with an equality constraint and one with an inequality constraint. Figure 8 shows a high-level circuit diagram. For each problem, the QAOA parameters are first optimized using a noiseless simulator. All circuit executions use 2000 shots and no error mitigation.

The first portfolio optimization instance we consider has an equality constraint on the four binary variables x_1, x_2, x_3, x_4 : $2x_1 - x_2 - x_3 = 0$. As discussed in the *Methods* Section, the semiclassical quantum Fourier transform (QFT) can be utilized for equality constraints. The semiclassical QFT makes use of QCL and midcircuit measurements, which are features supported by the H1-2 device. This results in an oracle that uses only one

auxiliary qubit, and thus the circuit uses five qubits in total. The circuit for the oracle is shown in Fig. 9. We note that the uncomputation step consists of resetting the one auxiliary qubit to the $|+\rangle$ state.

As a comparison, we also implement the coherent QFT (Fig. 10) on three qubits, resulting in seven qubits in total. After applying the oracle and measuring, all auxiliary qubits are reset to the ground state for the uncompute step. Figure 11a shows the in-constraint probability as a function of the number of projective measurements. Figure 12a shows the distributions of measurement outcomes of QAOA for varying numbers of measurements (N), with the outcomes (computational basis states) ordered by the objective function value. For both implementations, the in-constraint probability improves with the number of measurements up to $N \approx 15$. For a higher number of measurements, the hardware noise arising from high circuit depth prevents further improvements in the in-constraint probability $1 - \delta$.

While the QCL and non-QCL implementations both perform similarly, we do note a reduction in the number of two-qubit gates and auxiliary qubits. For QCL and $N = 15$, the two-qubit gate depth was 122 and the count was 123. Without QCL, for $N = 15$, the two-qubit gate depth was 148 and the count was 165. The similar performance between QCL and non-QCL versions despite the difference in gate count may be due to the higher impact of measurement error on the QCL implementation.

The second portfolio optimization instance we consider has a cardinality (Hamming-weight) inequality constraint $\sum_{j=1}^4 x_j \leq 2$. For this problem, it is necessary to utilize the coherent QFT, and thus QCL does not lead to a resource-requirement reduction. The QFT adder is used to compute $\sum_j x_j - 3$, which requires four qubits to accommodate the range. In addition, unlike the equality-constraint case, the inverse oracle is necessary for uncomputation. The system is in-constraint when the most-significant qubit, i.e., the sign bit, is a one. The circuit for the oracle is shown in Fig. 10. Similar to the previous run, we plot the in-constraint probability for varying numbers of measurements (Fig. 11b), as well as, the measurement distributions obtained from QAOA (Fig. 12b). For $N = 3$, the two-qubit gate depth is 112 and the count is 186. Similarly to the experiments with the equality constraint, the in-constraint probability $1 - \delta$ improves until $N = 3$. For a higher number of measurements, the hardware noise prevents further improvements.

Note that the performance deteriorates at a significantly lower N for the inequality constraint problem than equality. This occurs even though the two-qubit circuit depth is lower for the inequality case and the two-qubit gate count is not significantly higher. Besides the inclusion of an additional qubit, one potential reason for this is that for the inequality constraint, only one of the auxiliary qubits is measured and then the inverse oracle is applied. This allows for errors to accumulate more and propagate to the rest of circuit. However, in the equality constraint case, after applying the oracle, all auxiliary qubits are measured and then reset to the ground state. In addition, the total gate count happens to be significantly higher for the inequality constraint case.

Discussion

In this work, we propose an approach for enforcing constraints in quantum optimization and demonstrate its effectiveness by applying it to constrained instances of portfolio optimization in simulation and on a trapped-ion quantum processor. Our technique has two major advantages: the ability to enforce a very general class of constraints and the simplicity of hyperparameter tuning. Two important downsides of our approach are the

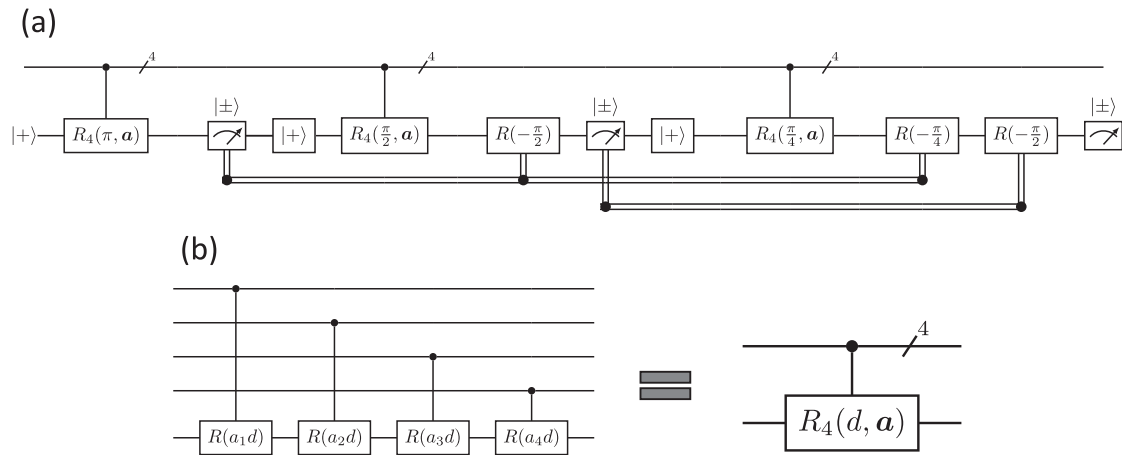


Fig. 9 QFT adder with QCL. Quantum circuits for (a) semiclassical quantum Fourier transform adder with quantum conditional logic (QCL) used in the hardware experiments involving equality constraints and (b) the four-qubit rotation gate used in a. Note that $R(a)$ denotes a phase gate. For the equality-constraint experiment executed on the H1-2 quantum device, we set $\mathbf{a} := (a_1, a_2, a_3, a_4) = (2, -1, -1, 0)$. The uncomputation step consists of resetting the auxiliary qubit to the $|+\rangle$ state.

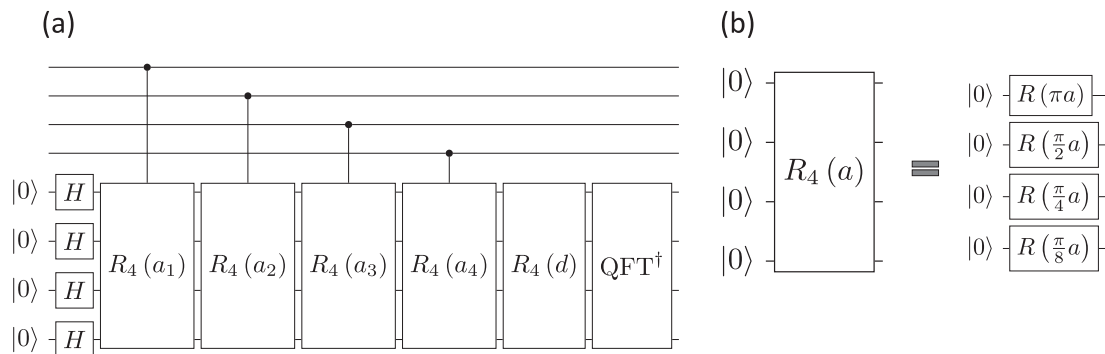


Fig. 10 QFT adder without QCL. Quantum circuits for (a) the quantum Fourier transform (QFT) adder used in the hardware experiments and (b) the four-qubit rotation gate used in a. Note that $R(a)$ denotes a phase gate. For the inequality-constraint experiment, we set $a_1 = a_2 = a_3 = a_4 = 1$, $d = -3$ and used four qubits for precision. For the equality-constraint experiment, without quantum conditional logic (QCL), we set $a_1 = 2$, $a_2 = a_3 = -1$, $a_4 = d = 0$ and used only three qubits for precision. For the inequality constraint, the inverse of the oracle is applied after measuring the qubit encoding the sign. However, for the equality constraint, since all auxiliary qubits are measured, we do not need to apply the inverse QFT operator and can simply reset all auxiliary qubits to the ground state. Note that here the inverse QFT operator (QFT†) does not include swaps as the reordering has been done by rearranging the banks of controlled rotations.

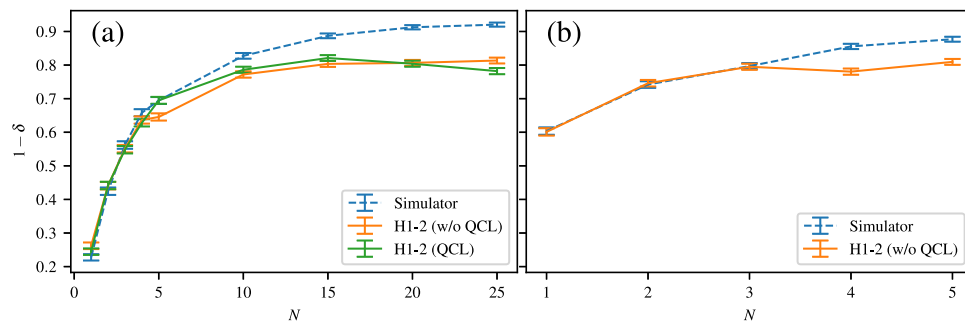


Fig. 11 Simulation and hardware experiment results using QAOA with Zeno dynamics. QAOA with $p = 1$ and Zeno dynamics was applied to solve a four-asset problem with an equality constraint $2x_1 - x_2 - x_3 = 0$ (a) and inequality constraint $\sum_{j=1}^4 x_j \leq 2$ (b). The circuits were executed on a classical simulator and on the H1-2 quantum device. The oracles are implemented using arithmetic in the Fourier domain. For the equality constraint (a), the quantum conditional logic (QCL) implementation of the Fourier adder used one auxiliary qubit, and the version without QCL used three auxiliary qubits. The Fourier adder used for the inequality constraint (b) used four auxiliary qubits. Error bars indicate the standard error of the mean arising from finite sampling (2000 shots). The in-constraint probability $1 - \delta$ grows with the number of measurements (N).

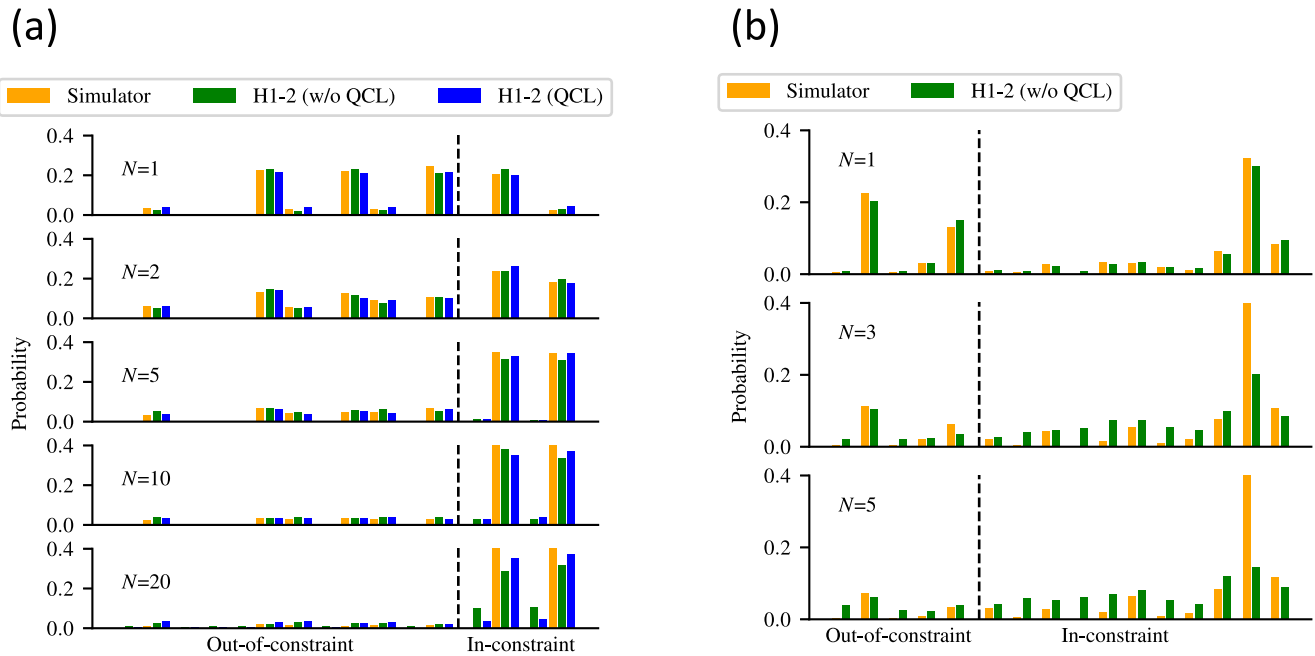


Fig. 12 Effectiveness of constraint enforcement using QAOA with Zeno dynamics in simulation and hardware experiments. Distribution of final measurement results obtained from QAOA applied to the equality- (a) and inequality-constrained (b) problem for different numbers of measurements (N). For the equality-constrained problem experiments were executed both with and without quantum conditional logic (QCL). Each column corresponds to a computational basis state (either in-constraint or out-of-constraint), and the columns are ordered by objective value (to the right is better). The circuits were executed on a classical simulator and on the H1-2 quantum device. There is strong agreement between the hardware results and results from noise-free simulation.

complexity of implementing the measurement and the possibility of the measurements resulting in trivial dynamics.

Implementing the oracle for a constraint in general requires quantum arithmetic and may lead to high gate count for more complex constraints. However, the asymptotic efficiency of our approach makes it viable for fault-tolerant quantum devices. Additionally, reductions in the cost of implementing quantum arithmetic, such as techniques utilizing quantum conditional logic, can further reduce the overhead of the proposed method.

Moreover, for noisy quantum devices, additional performance improvements can be obtained by leveraging advanced algorithm-specific error mitigation techniques such as the ones recently proposed for QAOA^{44,45}. Such techniques may help bridge the gap between the noisy near-term devices and the error correction likely required to execute circuits of sufficient depth to provide performance improvements over classical algorithms^{46–48}.

As discussed in the *Results* Section, restricting the evolution to the Zeno subspace may result in trivial dynamics for certain mixers. Therefore an important consideration when applying the proposed technique is evaluating whether the particular choice of mixer has this behavior. As this effect would apply generally to all instances with a given class of constraints, the mixer only needs to be analyzed once for a class of problems.

Methods

Preliminaries. We begin by briefly introducing the relevant concepts and setting the notation. We undertake the task of minimizing an objective function f defined on the Boolean cube, \mathbb{B}^n , over the set of feasible solutions $\mathcal{F} \subseteq \mathbb{B}^n$:

$$\min_{\mathbf{x} \in \mathcal{F}} f(\mathbf{x}). \quad (20)$$

We consider sets \mathcal{F} of the form $\mathcal{F} = \{\mathbf{x} \in \mathbb{B}^n \mid \bar{g}_j(\mathbf{x}) = 0 \ \forall j\}$, where $\bar{g}_j(\mathbf{x})$ is an oracle that returns 0 if \mathbf{x} satisfies the j -th constraint and a value strictly greater-than 0 otherwise. This general definition includes most commonly considered problems such as those with equality and inequality constraints.

This constrained optimization problem can be solved by relaxing the constraints and introducing penalty terms as follows:

$$\min_{\mathbf{x} \in \mathbb{B}^n} f_{\text{penalty}} = \min_{\mathbf{x} \in \mathbb{B}^n} f(\mathbf{x}) + \sum_j \lambda_j \bar{g}_j(\mathbf{x}), \quad (21)$$

where $\lambda_j \in \mathbb{R}^+$ are the penalty factors.

Specifically, for an equality constraint $g(\mathbf{x}) = 0$, the penalty function may be written as

$$\bar{g}(\mathbf{x}) = [g(\mathbf{x})]^2.$$

On the other hand, an inequality constraint $g(\mathbf{x}) \geq 0$ can be converted into an equivalent equality constraint $g(\mathbf{x}) - \hat{s} = 0$ by introducing a *slack variable* $\hat{s} \in [0, g_{\max}]$, where $g_{\max} = \max_{\mathbf{x} \in \mathcal{F}} g(\mathbf{x})$. If we assume $g(\mathbf{x})$ can be discretized with a spacing of Δ_g , then \hat{s} can be implemented using $n_{\text{slack}} = \lceil \log_2(g_{\max}/\Delta_g) \rceil$ binary variables $\mathbf{s} = (s_1, \dots, s_{n_{\text{slack}}})^T$, and the resultant equality constraint is $g(\mathbf{x}) - \Delta_g \sum_{j=1}^{n_{\text{slack}}} 2^{j-1} s_j = 0$. Therefore the penalty function for an inequality constraint can be written as

$$\bar{g}(\mathbf{x}; \mathbf{s}) = \left[g(\mathbf{x}) - \Delta_g \sum_{j=1}^{n_{\text{slack}}} 2^{j-1} s_j \right]^2.$$

The magnitudes of the penalty factors λ_j control how much the constraint violations are penalized. Intuitively, a higher value of λ_j should lead to a higher in-constraint probability. However, in practice, the relationship between the penalty factor, the in-constraint probability and the solution quality may be non-monotonic. This makes choosing λ_j harder. We discuss the difficulty of tuning the penalty factors in the *Results* Section.

Quantum algorithms for approximate optimization. In this work, we focus on the class of quantum optimization algorithms that use a parameterized quantum evolution to prepare a state, such that the corresponding measurement outcomes contain a high-quality, valid solution to the original optimization problem with high probability. This parameterized state, a restatement of Equation (1), is prepared by applying a parameterized evolution $U(\boldsymbol{\theta})$ to some initial state $|s\rangle$:

$$|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|s\rangle = \prod_{j=1}^m e^{-i\theta_j H_j} |s\rangle, \quad (22)$$

where H_j is some Hamiltonian, e.g., a tensor product of single-qubit Pauli operators.

Let $C = \sum_{\mathbf{x} \in \mathbb{B}^n} f(\mathbf{x}) |\mathbf{x}\rangle \langle \mathbf{x}|$ be the operator encoding the objective function f on qubits and $C_{\text{penalty}} = \sum_{\mathbf{x} \in \mathbb{B}^n} f_{\text{penalty}}(\mathbf{x}) |\mathbf{x}\rangle \langle \mathbf{x}|$ be the operator encoding the relaxed objective function (21). The figures of merit used to evaluate the quality of a

parameter θ^* obtained by algorithms that employ parameterized circuit (22) are approximation ratios, defined as follows:

$$r = \frac{\langle \psi(\theta^*) | C_{\mathcal{F}} | \psi(\theta^*) \rangle - f^{\min}}{f^{\min} - f^{\max}} \quad (23)$$

and

$$r_{\text{penalty}} = \frac{\langle \psi(\theta^*) | C_{\text{penalty}} | \psi(\theta^*) \rangle - f_{\text{penalty}}^{\min}}{f_{\text{penalty}}^{\min} - f_{\text{penalty}}^{\max}}, \quad (24)$$

where $C_{\mathcal{F}} = \sum_{\mathbf{x} \in \mathcal{F}} f(\mathbf{x}) |\mathbf{x}\rangle \langle \mathbf{x}|$, $f^{\min} = \min_{\mathbf{x} \in \mathcal{F}} f(\mathbf{x})$, $f^{\max} = \max_{\mathbf{x} \in \mathcal{F}} f(\mathbf{x})$, $f_{\text{penalty}}^{\min} = \min_{\mathbf{x} \in \mathbb{B}^n} f_{\text{penalty}}(\mathbf{x})$, and $f_{\text{penalty}}^{\max} = \max_{\mathbf{x} \in \mathbb{B}^n} f_{\text{penalty}}(\mathbf{x})$.

This class of algorithms includes QAOA^{1,2,48} and its generalization, the quantum alternating operator ansatz algorithm¹³. In both algorithms, the parameterized quantum evolution is performed by applying pairs of alternating operators:

$$|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \prod_{j=1}^p [U_B(\beta_j) U_C(\gamma_j)] |s\rangle, \quad (25)$$

where $U_C(\gamma_j) = e^{-i\gamma_j C}$ is the phase operator, and $U_B(\beta_j)$ is the mixing operator. In the special case of QAOA, the initial state $|s\rangle$ is the uniform superposition over all computational basis states and the mixing operator U_B is set to be $U_B(\beta_j) = e^{-i\beta_j B}$, where $B = \sum_k x_k$ is a sum of single-qubit Pauli- x operators. In quantum alternating operator ansatz, U_B and $|s\rangle$ are allowed to be arbitrary, and are typically set such that the resulting state $|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle$ preserves the constraints, in the sense that every measurement outcome \mathbf{x} belongs to \mathcal{F} . In this paper, we consider QAOA with an arbitrary mixing Hamiltonian B , defined in ref. ¹³ as Hamiltonian-based QAOA. In all other sections of this paper, unless it is specified otherwise, the acronym QAOA is used to denote this version of the algorithm.

In addition to QAOA, we consider the layer variational quantum eigensolver (L-VQE)³¹, which is a version of VQE with the hardware-efficient layered parameterized circuit tailored towards optimization problems. L-VQE uses the parameterized circuit of the form

$$\prod_{j=1}^p [U_{\text{NN}}(\boldsymbol{\theta}_j)] V(\boldsymbol{\theta}_0) |0\rangle, \quad (26)$$

where U_{NN} consists of nearest-neighbor CNOT's and single-qubit Ry's, and V is a layer of single-qubit Ry's. The reader is referred to ref. ³¹ for the precise definition of the circuit. While the circuit includes non-parameterized CNOT's, it is easy to write it equivalently in the form of Equation (22) by pushing Ry through the control of the CNOT and noting that $\text{Ry}(\theta) = e^{-i\frac{\theta}{2} \sigma_y}$ and $\text{cnot}_{1,2} \text{Ry}_2(\theta) \text{cnot}_{1,2} = e^{-i\frac{\theta}{2} \sigma_{y,2}}$. Here, y_j and z_j denote a single-qubit Pauli- y and Pauli- z , respectively, acting on the j -th qubit.

Quantum Zeno dynamics. The quantum Zeno effect (QZE)^{49,50} is named after Zeno's paradox⁵¹, which regards the continuous observation of a moving arrow. Zeno's paradox states that an arrow cannot move if no time has elapsed since the point it was last observed. If the time difference between observations is Δt , continuous observation occurs in the limit of $\Delta t \rightarrow 0$. Under continuous observation, no time elapses between observations, and during each observation the arrow is not moving; thus, no overall movement is possible. The analog in quantum mechanics is a consequence of the Schrödinger equation. We first introduce a simpler one-dimensional version, in which the quantum state is restricted from evolving due to repeated measurements, and then present a more general case in which the dynamics of the system are restricted to a particular subspace, called a Zeno subspace.

Suppose a time-dependent quantum state is evolved in a finite-dimensional Hilbert space \mathcal{H} from some initial state $|\psi_0\rangle$ under the action of some Hamiltonian H for time t . Define a projective measurement \mathcal{P} given by a pair of complement projections $P = |\psi_0\rangle \langle \psi_0|$ and $Q = I - P$, which acts on a density operator ρ as

$$\mathcal{P}\rho = P\rho P + Q\rho Q.$$

If we carry out N repeated projective measurements \mathcal{P} at a time interval of t/N , then the probability that the system remains in the initial state is

$$\begin{aligned} p(t) &= \|P e^{-iHt/N} |\psi_0\rangle\|_2^{2N} \\ &= [| \langle \psi_0 | e^{-iHt/N} | \psi_0 \rangle |^2]^N \\ &= [1 - (t/N \tau_Z)^2]^N + O(N^{-2}) \xrightarrow{N \rightarrow \infty} 1, \end{aligned}$$

where $\tau_Z^{-2} = \langle \psi_0 | H^2 | \psi_0 \rangle - \langle \psi_0 | H | \psi_0 \rangle^2$ is called the Zeno time and quantifies how often the measurements need to be taken. As the frequency at which the measurements are performed increases without bound, the probability of remaining in the initial state approaches one.

Quantum Zeno dynamics (QZD)^{52–55} considers the more general case where the evolution of the state is constrained to a subspace of dimension greater than one. Thus the projective measurement \mathcal{P} can contain multiple projections with

ranks all greater than one. Specifically, a restatement of Equation (2),

$$\mathcal{P}\rho = \sum_{j=1}^k P_j \rho P_j, \quad (27)$$

where $\sum_{j=1}^k P_j = I$, and P_j is a projection onto some subspace $\mathcal{H}_j = P_j \mathcal{H}$ of dimensionality $\text{Tr}(P_j) \geq 1$. Informally, QZD states that if the evolution starts in \mathcal{H}_j and the measurement \mathcal{P} is performed sufficiently often, then the system will remain in \mathcal{H}_j with high probability.

Consider an initial state ρ_0 , after N projective measurements by \mathcal{P} , the state of the system is given by

$$\rho(t) = \mathcal{U}(t) \rho_0 \mathcal{U}(t)^\dagger, \quad (28)$$

where $\mathcal{U}(t) = (P e^{-iHt/N})^N$ and $p(t) = \text{Tr}(P_j \rho(t))$ is the probability of the system remaining in \mathcal{H}_j after evolving for time t . Note that

$$\begin{aligned} \mathcal{U}(t) &= (P e^{-iHt/N})^N \\ &= (P [I - iHt/N + O(N^{-2})])^N \\ &= (I - iPHt/N + O(N^{-2}))^N \\ &= (I - iPHt/N)^N + O(N^{-1}) \end{aligned} \quad (29)$$

$$\xrightarrow{N \rightarrow \infty} e^{-iPHt} P, \quad (30)$$

and the dynamics of the system are governed by $H_Z = PH$, called the Zeno Hamiltonian. Moreover, as $N \rightarrow \infty$, transitions between different subspaces $\{\mathcal{H}_1, \dots, \mathcal{H}_k\}$ of \mathcal{H} are suppressed. This implies if $\rho_0 = P_j \rho_0 P_j$ for some $j \in [k] := \{1, \dots, k\}$, then in the limit of $N \rightarrow \infty$, called the Zeno limit, it follows that $p(t) \rightarrow 1$, and thus the state will remain in \mathcal{H}_j throughout the evolution. For a more detailed discussion the reader is referred to refs. ^{54,55}.

QZE has many applications in algorithms and error mitigation. Childs et al.⁵⁶ propose a version of Grover's search based on QZD that utilizes frequent measurements instead of slow adiabatic evolution. This alternative approach to slow evolution was also observed in ref. ⁵⁷. Somma et al.^{58,59} develop a quantum-enhanced version of the simulated annealing algorithm. Their approach makes use of QZD to ensure that the evolution remains in the instantaneous quantum Gibbs state for varying temperature. Boixo et al.⁶⁰ show that for Grover's algorithm and simulated annealing based on QZD, one could use frequent randomized evolutions instead of measurements (the randomization method). The randomization method has also been used to implement algorithms for quantum linear systems^{61,62}. Finally, dynamical decoupling, also called bang-bang decoupling⁶³, is a popular error-mitigation technique that uses QZE to suppress decoherence^{55,64–68}.

Proof of Theorem 1. In this Section we derive our main result, Theorem 1, for the number of measurements required to maintain a constant success probability. We start by deriving the required lemmas.

Lemma 1. Let H be a Hermitian matrix. Then

$$\begin{aligned} \min_{P, |\psi\rangle \in \text{Im}(P)} \|P e^{-i\theta H} |\psi\rangle\|_2^2 &= \cos^2\left(\frac{\xi_{\max} - \xi_{\min}}{2} \theta\right) \\ \forall \theta \in \mathbb{R}, |\theta| &\leq \frac{\pi}{\xi_{\max} - \xi_{\min}}, \end{aligned}$$

where P is an orthogonal projector and ξ_{\max} and ξ_{\min} are the largest and smallest eigenvalues of H .

Proof. Suppose H has the following eigendecomposition

$$H = \sum_{k=1}^d \xi_k Q_k,$$

where ξ_k are the unique eigenvalues of H (including 0 if H is not full rank) and $\{Q_k\}_{k=1}^d$ is the complete set of projectors onto the corresponding eigenspaces. Therefore

$$\begin{aligned} p(\theta) &= \|P e^{-i\theta H} |\psi\rangle\|_2^2 \\ &\geq \| |\psi\rangle \langle \psi | P e^{-i\theta H} | \psi \rangle \|_2^2 \\ &= | \langle \psi | e^{-i\theta H} | \psi \rangle |^2 \\ &= \left| \sum_{j,k=1}^d e^{i\theta(\xi_j - \xi_k)} \langle \psi | Q_j | \psi \rangle \langle \psi | Q_k | \psi \rangle \right| \\ &= \left| \sum_{j,k=1}^d \cos(\theta(\xi_j - \xi_k)) \langle \psi | Q_j | \psi \rangle \langle \psi | Q_k | \psi \rangle \right| \\ &= \sum_{j,k=1}^d c_{jk} x_j x_k, \end{aligned} \quad (31)$$

where $c_{jk} = \cos(\theta(\xi_j - \xi_k))$, $x_j = \langle \psi | Q_j | \psi \rangle \geq 0$. Note that the second to the last

equality follows from

$$e^{i\theta(\xi_j - \xi_k)} x_j x_k + e^{i\theta(\xi_k - \xi_j)} x_k x_j = \cos(\theta(\xi_j - \xi_k)) x_j x_k + \cos(\theta(\xi_k - \xi_j)) x_k x_j.$$

Let C be the matrix with elements c_{jk} at the j -th row and k -th column. Then using simple trigonometric identities, it can be shown that

$$C = \mathbf{v}(\mathbf{v})^T + \mathbf{v}\left(\frac{\pi}{2} - \theta\right)\mathbf{v}\left(\frac{\pi}{2} - \theta\right)^T \quad (32)$$

where

$$\mathbf{v}(\theta) = (\cos(\xi_1\theta), \dots, \cos(\xi_d\theta))^T. \quad (33)$$

Since C is the sum of positive semi-definite matrices, it too is positive semi-definite.

Therefore, minimizing $p(\theta)$ is equivalent to solving the following convex constrained minimization problem

$$\min_{\mathbf{x} \in \mathcal{S}} \mathbf{x}^T C \mathbf{x}, \text{ where } \mathcal{S} := \{\mathbf{x} \in \mathbb{R}_+^d \mid \|\mathbf{x}\|_1 = 1\}, \quad (34)$$

$\mathbf{x} = (x_1, \dots, x_d)^T$ and thus a sufficient condition⁶⁹, Theorem 2.2.5 for \mathbf{x}^* to be the optimum is

$$\mathbf{x}^{*T} C (\mathbf{x} - \mathbf{x}^*) \geq 0, \forall \mathbf{x} \in \mathcal{S} \quad (35)$$

Consider the following trial solution

$$\begin{aligned} x_{\min}^* &= x_{\max}^* = \frac{1}{2}, \\ x_j^* &= 0 \quad \forall j \notin \{\min, \max\}. \end{aligned} \quad (36)$$

We have that $\forall \mathbf{x} \in \mathcal{S}$

$$\begin{aligned} 2\mathbf{x}^{*T} C (\mathbf{x} - \mathbf{x}^*) &= (1 + c_{\max, \min})(x_{\max} + x_{\min} - 1) + \sum_{j \notin \{\min, \max\}} x_j (c_{\max, j} + c_{\min, j}) \\ &= (1 - x_{\max} - x_{\min}) \left[\sum_{j \notin \{\min, \max\}} (c_{\max, j} + c_{\min, j}) - (1 + c_{\max, \min}) \right] \end{aligned}$$

Also for $|\theta| \leq \frac{\pi}{\xi_{\max} - \xi_{\min}}$, we have $c_{j,k} \geq c_{\max, \min}$, and thus

$$\begin{aligned} 1 + c_{\min, \max} &= 2\cos^2\left(\frac{\xi_{\max} - \xi_{\min}}{2}\theta\right) \\ &\leq 2\cos\left(\frac{\xi_{\max} - \xi_{\min}}{2}\theta\right)\cos\left(\frac{\xi_{\max} + \xi_{\min} - 2\xi_j}{2}\theta\right) \\ &= c_{\max, j} + c_{\min, j}. \end{aligned}$$

Combining the above results, we obtain that $2\mathbf{x}^{*T} C (\mathbf{x} - \mathbf{x}^*) \geq 0$. Thus our choice is optimal.

After, plugging in the optimal choice and noting that all steps are equalities in (31) when $P = |\psi\rangle\langle\psi|$, we obtain:

$$\min_{P, |\psi\rangle \in \text{Im}(P)} \|Pe^{-i\theta H} |\psi\rangle\|_2^2 = \mathbf{x}^{*T} C \mathbf{x}^* = \cos^2\left(\frac{\xi_{\max} - \xi_{\min}}{2}\theta\right).$$

Additionally, the result implies that minimization occurs when

$$|\psi\rangle = |\pm_H\rangle := \frac{1}{\sqrt{2}}|\xi_{\max}\rangle \pm \frac{1}{\sqrt{2}}|\xi_{\min}\rangle \quad (37)$$

for any $|\xi_{\max}\rangle \in \text{Im}(Q_{\max})$ and $|\xi_{\min}\rangle \in \text{Im}(Q_{\min})$.

Note as observed in the proof of Lemma 1, the lower bound on the in-constraint probability bound is saturated when the initial state is chosen to be either $|\pm_H\rangle$ or $|\pm_{-H}\rangle$ in Equation (37), and P is the projector onto the chosen initial state.

Lemma 2. Let H be a Hermitian matrix. Then

$$\begin{aligned} \min_{P, |\psi\rangle \in \text{Im}(P)} \left\| P \left(\mathcal{P} e^{-i\frac{\theta}{N} H} \right)^N |\psi\rangle \right\|_2^2 &= \frac{1}{2} + \frac{1}{2} \left[2p^*\left(\frac{\theta}{N}\right) - 1 \right]^2, \\ \forall \theta \in \mathbb{R}, |\theta| &\leq \frac{\pi N}{\xi_{\max} - \xi_{\min}}, \end{aligned}$$

where \mathcal{P} is a projective measurement as defined in Equation (27) with projectors P and $I - P$,

$$p^*\left(\frac{\theta}{N}\right) = \cos^2\left(\frac{\xi_{\max} - \xi_{\min}}{2N}\theta\right),$$

and ξ_{\max} and ξ_{\min} are the largest and smallest eigenvalues of H .

Proof. Consider a fixed θ and some N that satisfies the hypothesis. The stochastic process formed by random variables indicating whether the system is in $\text{Im}(P)$ or its complement after each evolution segment $\mathcal{P}e^{-i\frac{\theta}{N}H}$ form a two-state Markov chain. According to Lemma 1, the probability of remaining in a state on the chain at any point in time is at least

$$p^*\left(\frac{\theta}{N}\right) := \cos^2\left(\frac{\xi_{\max} - \xi_{\min}}{2N}\theta\right), \quad (38)$$

and this minimum probability is attained at each segment when $|\psi\rangle$ is (37) and $P = |\psi\rangle\langle\psi|$. Because, in this case, the evolution lies in the two-dimensional space spanned by $|\pm_H\rangle$, the result is a Markov chain with transition matrix

$$A(k) = \bar{A} = \begin{pmatrix} p^* & 1 - p^* \\ 1 - p^* & p^* \end{pmatrix}, \forall k \in [N], \quad (39)$$

and $\forall k > N, A(k) = I$.

Therefore the probability of the state remaining in $\text{Im}(P)$ after N steps of the chain is $\bar{A}_{0,0}^N$, or the first diagonal element of the matrix \bar{A} after raising it to the N -th power. Applying diagonalization on \bar{A} , we obtain

$$\bar{A}_{0,0}^N = \frac{1 + (2p^* - 1)^N}{2}. \quad (40)$$

We now proceed to derive Theorem 1 using the above lemmas.

Proof of Theorem 1. For all $\theta \in \mathbb{R}$, such that

$$|\theta| < \frac{N}{\xi_{\max} - \xi_{\min}}, \quad (41)$$

it follows that

$$\begin{aligned} \cos^2\left(\frac{\xi_{\max} - \xi_{\min}}{2N}\theta\right) &\geq \left(1 - \frac{1}{2} \left[\frac{\theta(\xi_{\max} - \xi_{\min})}{2N} \right]^2 \right)^2 \\ &\geq 1 - \frac{[\theta(\xi_{\max} - \xi_{\min})]^2}{4N^2}. \end{aligned}$$

If we combine this result with Lemma 2, then we obtain

$$\begin{aligned} \frac{1}{2} + \frac{1}{2} [2p^*\left(\frac{\theta}{N}\right) - 1]^N &\geq \frac{1}{2} + \frac{1}{2} \left(1 - \frac{[\theta(\xi_{\max} - \xi_{\min})]^2}{2N^2}\right)^N \\ &\geq \frac{1}{2} + \frac{1}{2} \exp\left(-\frac{[\theta(\xi_{\max} - \xi_{\min})]^2}{2N}\right) \end{aligned} \quad (42)$$

To lower bound this by $1 - \delta$, we can choose N as stated in Theorem 1. Note that to ensure Equation (41) we must have

$$\frac{[\theta(\xi_{\max} - \xi_{\min})]^2}{N} < N, \quad (43)$$

and thus

$$\frac{1}{2} + \frac{1}{2} \exp\left(-\frac{[\theta(\xi_{\max} - \xi_{\min})]^2}{2N}\right) > \frac{1}{2} + \frac{1}{2} \exp\left(-\frac{N}{2}\right). \quad (44)$$

At the minimum of value of N , we have

$$\frac{1}{2} + \frac{1}{2} \exp\left(-\frac{1}{2}\right) \lesssim 0.81. \quad (45)$$

Proof of Corollary 1

Proof. For simplicity, consider a single block of size m :

$$\mathcal{U}_Z(\theta) = \left[\prod_{j=1}^m e^{-i(\theta_j/N)H_j} \right]^N. \quad (46)$$

First, suppose that the elements of $\{H_j\}_{j=1}^m$ do not all pairwise commute. Then, according to⁷⁰, Proposition 9:

$$\left\| \prod_{j=1}^m e^{-i(\theta_j/N)H_j} - e^{-i\sum_{j=1}^m (\theta_j/N)H_j} \right\|_2 \leq \frac{1}{2N^2} \sum_{j=1}^m \left\| \left[\sum_{f=j+1}^m \theta_f H_f, \theta_j H_j \right] \right\|_2 \quad (47)$$

This implies that

$$\begin{aligned} \left\| \mathcal{U}_Z(\theta) - \left[\mathcal{P} e^{-i\sum_{j=1}^m (\theta_j/N)H_j} \right]^N \right\|_2 &\leq \frac{1}{2N} \sum_{j=1}^m \left\| \left[\sum_{f=j+1}^m \theta_f H_f, \theta_j H_j \right] \right\|_2 \\ &\leq \frac{\left[\sum_{j=1}^m |\theta_j| \right]^2 \max_j \|H_j\|_2^2}{N}. \end{aligned} \quad (48)$$

Then

$$\|P_G \mathcal{U}_Z(\theta) |\psi\rangle\|_2^2 \leq \left(\left\| P_G \left[\mathcal{P} e^{-i\sum_{j=1}^m (\theta_j/N)H_j} \right]^N |\psi\rangle \right\|_2 + \frac{\left[\sum_{j=1}^m |\theta_j| \right]^2 \max_j \|H_j\|_2^2}{N} \right)^2. \quad (49)$$

If we choose

$$N = \left\lceil \frac{4 \left[\sum_{j=1}^m |\theta_j| \right]^2 \max_j \|H_j\|_2^2}{\ln(1 - \delta)^{-2\alpha}} \right\rceil, \quad (50)$$

then for $\alpha \leq 1$, Theorem 1 with Remark 1 implies that the out-of-constraint probability is at most

$$\|P_{\mathcal{G}}\mathcal{U}_Z(\theta)|\psi\rangle\|_2^2 \leq \frac{\delta}{2} + \alpha \frac{\sqrt{\delta}}{2} \ln(1-\delta)^{-2} + \frac{\alpha^2}{16} \ln^2(1-\delta)^{-2} \quad (51)$$

$$\leq \frac{\delta}{2} + \frac{\delta}{2} \left[\alpha + \frac{\alpha^2}{8} \right], \quad (52)$$

where $\delta \leq 0.19$. If $\alpha = 0.89$, then

$$\|P_{\mathcal{G}}\mathcal{U}_Z(\theta)|\psi\rangle\|_2^2 < \delta. \quad (53)$$

To compensate for the decay of the success probability after L blocks, each N_k must be multiplied by L .

Lastly, for the asymptotic dynamics, from Equation (29)–(30) we get

$$\begin{aligned} \mathcal{U}_Z(\theta) &= \left[\mathcal{P} \prod_{j=1}^m \left(I - i(\theta_j/N)H_j + O(N^{-2}) \right) \right]^N \\ &= \left[\mathcal{P} \left(I - i \sum_{j=1}^m (\theta_j/N)H_j + O(N^{-2}) \right) \right]^N \end{aligned} \quad (54)$$

$$\xrightarrow{N \rightarrow \infty} e^{-i \sum_{j=1}^m \mathcal{P} H_j \theta_j \mathcal{P}} = e^{-i \mathcal{P} \mathbf{H} \cdot \boldsymbol{\theta} \mathcal{P}}. \quad (55)$$

Thereby the dynamics are described by the Zeno Hamiltonian $\mathbf{H}_Z = \mathcal{P} \mathbf{H}$, where \mathcal{P} acts element-wise on the vector $\mathbf{H} = (H_1, \dots, H_m)^T$. The limiting dynamics of L blocks is the product of these limits.

If the elements of $\{H_j\}_{j=1}^m$ pairwise commute, then there is no Trotter error, and $\alpha = 1$ without the need to halve δ . The limiting dynamics follows trivially as well.

Realizing oracles for combinatorial constraints. In this Section, we review the constructions of quantum oracles for implementing polynomial inequality and equality constraints. We use the constructions provided in this Section in the experiments on a trapped-ion quantum computer described in the *Results* Section. Since any function on the Boolean cube can be expressed as a polynomial it suffices to only demonstrate constructions for polynomial constraints⁷¹. In addition, since we are considering problems in NPO we can assume the existence of a polynomially-sized classical circuit for evaluating any constraints to sufficient precision. Given that all classical basis gates can be represented as polynomials, we can represent our constraint as the composition of polynomially many polynomial functions. Of course, one could also directly implement the classical circuit in a reversible fashion on a quantum device efficiently. For the remainder of this Section, we consider a polynomial function g :

$$g(\mathbf{b}) = \sum_{k=1}^K d_k \prod_{l \in S_k} b_l, \quad (56)$$

where $S_k \subseteq [n]$ and $d_k \in \mathbb{R}$. In addition for $S_k = \emptyset$, $\prod_{l \in S_k} b_l = 1$.

Without loss of generality we can assume that equality constraints are of the form $g(\mathbf{b}) = 0$ and inequality constraints are of the form $g(\mathbf{b}) \geq 0$. We assume that there exists an oracle that computes the value of $g(\mathbf{b})$ into a quantum register (constructions of such oracles are briefly reviewed in the *Methods* Section). For an equality constraint, we implement the constraint-enforcing measurement by simply measuring the entire register. A projection onto the in-constraint subspace implies that we have observed a 0 in the register. For an inequality constraint, we measure the qubit corresponding to the sign, a 0 corresponds to a successful projection, and apply the inverse of the oracle post measurement.

While the above procedure works in general, there are further optimizations that can be made by utilizing quantum conditional logic (QCL). We give an example of such an optimization in the *Results* Section. Further optimizations are possible for double-sided inequalities of the form $0 \leq g(\mathbf{b}) < a$, where a is a power of 2. To implement the measurement corresponding to this double-sided inequality, we only need to measure higher-order bits. Since the results of these high-order bits are now classical, we can replace the part of the inverse-oracle circuit controlled on these bits with classically-conditioned single-qubit gates. Lastly, because all constraint-preserving measurements can be implemented separately and thus auxiliary qubits can be reused, the required number of auxiliary qubits to implement all constraint-preserving measurements is equal to the maximum amount of auxiliary qubits required by any oracle call.

In the subsections that follow, we present efficient constructions of oracles that can be used to implement polynomial functions. Both of these use techniques that have been presented in prior work. Here we include a brief review for completeness and present the resource analysis for our setting.

Review of classical reversible arithmetic circuits. The design of reversible versions of classical arithmetic circuits has been extensively explored and highly optimized constructions are available^{72–74}. Such constructions allow one to implement unitary operations for performing arithmetic on quantum registers. Consider fixed-point arithmetic of m bits including digits both before and after the decimal point. Suppose polynomial g has K terms. For each coefficient d_k , we require an n -qubit controlled m -bit adder. A controlled m -bit adder can be implemented with $O(m)$ T

gates⁷⁵. Since a multi-controlled Toffoli can be implemented with a T count of $O(n)$ ^{76,77} and thus the overall multi-controlled adder can be implemented with a T count of $O(n + m)$. The T count for implementing g is $O(K(n + m))$.

Review of quantum Fourier arithmetic. For smaller quantum devices, a more resource efficient approach is to switch to the Fourier basis using the quantum Fourier transform (QFT) and perform the arithmetic in the Fourier basis. This approach has worse asymptotic complexity in terms of T-gate counts, but requires fewer qubits and CNOT gates. We use this approach in the hardware experiments discussed in the *Results* Section. The discussion in this Section is based on ref. ²¹, though the idea of using the QFT for quantum arithmetic is well-known, see e.g.^{78–80}.

For $s \in [2^m]$, the QFT on \mathbb{Z}_{2^m} is defined as follows:

$$\text{QFT}_{2^m} : |s\rangle \mapsto \sum_{k \in [2^m]} e^{-i2\pi ks/2^m} |k\rangle. \quad (57)$$

It can be shown⁸¹ that the right-hand side of (57) is a product state and can be expressed in the following form:

$$\bigotimes_{k=1}^m \frac{|0\rangle + e^{-i\pi \frac{s}{2^{m-k}}} |1\rangle}{\sqrt{2}} = F_m\left(\frac{s}{2^m}\right) |+\rangle^{\otimes m}, \quad (58)$$

where

$$F_m(\theta) := \bigotimes_{k=1}^m R(\pi 2^k \theta) \quad (59)$$

implements the desired operation. In addition, $R(\alpha)$ denotes the phase gate $|0\rangle\langle 0| + e^{i\alpha}|1\rangle\langle 1|$. The angle θ is restricted to $[-\frac{1}{2}, \frac{1}{2}]$ to avoid overflow and allow for representing negative numbers. Thus, when implementing a polynomial g , we require that its range match the range of g , i.e., $\|g\|_\infty \leq \frac{1}{2}$. This can always be satisfied by scaling g accordingly.

As an example, we can add two integers a and b , with the conditions $a, b, a + b \in \{-2^{m-1}, \dots, 0, \dots, 2^{m-1} - 1\}$, as follows:

$$\text{QFT}_{2^m}^\dagger F_m\left(\frac{a}{2^m}\right) F_m\left(\frac{b}{2^m}\right) |+\rangle^{\otimes m} = |a + b\rangle. \quad (60)$$

Note, the value in the quantum register is really the two's complement of $a + b$. We define the following controlled operation:

$$F_m(\mathbf{b}, \theta) := |\mathbf{b}\rangle\langle \mathbf{b}| \otimes F_m(\theta) + (I - |\mathbf{b}\rangle\langle \mathbf{b}|) \otimes I, \quad (61)$$

where $\mathbf{b} \in \mathbb{B}^n$. For $S_k \subseteq [n]$, let $\mathbf{I}_{S_k} \in \mathbb{B}^n$ denote the indicator vector of S_k . The process for (approximately) loading the value of the polynomial (56) into a quantum register is:

$$(I \otimes \text{QFT}_{2^m}^\dagger) \prod_{k=1}^K F_m(\mathbf{I}_{S_k}, d_k) |\mathbf{b}\rangle |+\rangle^{\otimes m} = |\mathbf{b}\rangle |\tilde{g}(\mathbf{b})\rangle, \quad (62)$$

where by the assumption on the range of g , $|\tilde{g}(\mathbf{b}) - g(\mathbf{b})| \leq 2^{-m}$. The result is stored in an auxiliary quantum register of size $O(m)$. The operation $F_m(\mathbf{b}, \theta)$ requires m n -controlled rotation gates. Thus overall it requires Km n -controlled rotation gates. An $O(n)$ -controlled Toffoli can be implemented with $O(n)$ T gates^{76,77} and each controlled rotation can be ϵ -approximately implemented with $O(\log(1/\epsilon))$ T's^{82,83}. Thus, assuming a fixed rotation-gate approximation error the total cost is $O(Kmn)$.

The operation QFT_{2^m} requires $O(m^2)$ gates to be implemented exactly⁸¹ and can be implemented approximately, for a fixed approximation error, on a fault-tolerant device with $O(m \log(m))$ T gates⁸³. For equality constraints, since we will be measuring the entire register containing the value $\tilde{g}(\mathbf{b})$, we swap the coherent implementation of the inverse QFT for the semiclassical variant^{84,85}. This semiclassical version of the QFT replaces all two-qubit gates with classically-controlled single qubit gates and requires only a single auxiliary qubit that is repeatedly measured and reset to compute the bits of $\tilde{g}(\mathbf{b})$. Thus, this approach benefits from both mid-circuit measurements and QCL. A fault-tolerant version of this circuit can be approximately implemented with $O(m \log(m))$ T gates⁸⁶. Thus in a fault-tolerant setting the overall T count of the QFT-based approach is $O(Kmn + m \log(m))$.

Initial state construction. Our proposed approach is flexible with regards to the choice of the initial state, any initial state that is in-constraint suffices. Thus, unlike ref. ²⁰, when using the complete-graph mixer our approach does not require repeated applications of a unitary and its inverse for preparing the uniform superposition of in-constraint states. However, the initial state we use in experiments discussed in the *Results* Section is the uniform superposition over all computational basis states encoding in-constraint solutions. In general, this superposition is hard to prepare. However, there exist constructions for a wide range of practically relevant cases. If the set of feasible solutions is efficiently indexable, (ref. ²⁴, Section IIIB) gives an efficient procedure for the initial state preparation. In the specific case of a Hamming-weight equality or inequality constraint, the uniform superposition over feasible states is a superposition of Dicke states with corresponding Hamming weights, which can be constructed

efficiently⁸⁷. Since, our technique does not require the state preparation method be reversible, we can make use of repeat-until-success schemes.

Parameter optimization. The Zeno framework we propose works well with standard techniques used to optimize parameterized quantum circuits. Specifically, as long as each N_r is large enough to ensure the desired minimum in-constraint probability is $1 - \delta$ (c.f. Corollary 1) for the given parameter range, the direction of steepest descent will still result in a circuit with the same minimum in-constraint probability. Here we make an assumption that θ remains bounded throughout optimization, which is a valid assumption in practice. This means that both gradient-based and gradient-free local optimization methods can be used with Zeno-augmented hybrid quantum-classical algorithms. A commonly used way to optimize parameterized quantum circuits is to use the parameter-shift rule^{88,89} in conjunction with a gradient-based optimizer. We now show that the Zeno framework works efficiently with the parameter-shift rule.

We consider the task of finding a minimum-eigenvalue state of an observable M using a parameterized quantum evolution consisting of generating Hamiltonians that are also unitary, e.g. L-VQE. We utilize the measurement scheme presented in Equation (6) with the condition that $\forall k, m_k = 1$. (Following similar arguments as⁸⁸, Section 3), we obtain

$$\begin{aligned} \frac{\partial}{\partial \theta_r} \text{Tr} \{ M U_Z(\theta) \rho U_Z^\dagger(\theta) \} &= \sum_{k=1}^{N_r} \text{Tr} \{ M_k P_{N_r}^H e^{-i \frac{\theta_r}{N_r} H_r} \rho_k + \text{h.c.} \} \\ &= \frac{1}{N_r} \sum_{k=1}^{N_r} \left[\text{Tr} \{ M U_Z^{+(r,k)} \rho U_Z^{\dagger,+(r,k)} \} - \text{Tr} \{ M U_Z^{-(r,k)} \rho U_Z^{\dagger,-(r,k)} \} \right], \end{aligned} \quad (63)$$

where M_k and ρ_k contain terms that have not been differentiated, and $U_Z^{\pm(r,k)}$ is the same as $U_Z(\theta)$ except that the evolution at the $\sum_{t=1}^{r-1} N_t + k$ -th step has a phase shift of $\pm \frac{\pi}{N_r}$. Thus, whereas the normal parameter-shift requires two expectation evaluations per parameter, Zeno would require $2N_r$. This is the same additional overhead as in the case of a circuit with gates that share parameters.

It also easy to see that the gradient is biased towards minimizing $M_{\mathcal{F}} = P_{\mathcal{F}} M P_{\mathcal{F}}$, i.e. the in-constraint Hamiltonian, as follows:

$$\begin{aligned} \frac{\partial}{\partial \theta_r} \text{Tr} \{ M U_Z(\theta) \rho U_Z^\dagger(\theta) \} &= \text{Pr}_{\mathcal{F}} \frac{\partial}{\partial \theta_r} \text{Tr} \left\{ M_{\mathcal{F}} \frac{U_Z(\theta) \rho U_Z^\dagger(\theta)}{\text{Pr}_{\mathcal{F}}} \right\} \\ &+ \text{Pr}_{\mathcal{G}} \frac{\partial}{\partial \theta_r} \text{Tr} \left\{ M_{\mathcal{G}} \frac{U_Z(\theta) \rho U_Z^\dagger(\theta)}{\text{Pr}_{\mathcal{G}}} \right\}, \end{aligned} \quad (64)$$

where $\text{Pr}_{\mathcal{F}}$ is the probability of projecting onto \mathcal{F} when measuring the parameterized evolution with \mathcal{P} . Lastly, Corollary 1 can be used to ensure $\text{Pr}_{\mathcal{F}} > 1 - \delta$.

Data availability

We make all the data presented in this paper available online at <https://doi.org/10.5281/zenodo.7125969>.

Code availability

We make the code required to reproduce the figures presented in this paper as well as the code executed on quantum hardware available online at <https://doi.org/10.5281/zenodo.7125969>.

Received: 6 June 2023; Accepted: 31 July 2023;

Published online: 18 August 2023

References

- Hogg, T. & Portnov, D. Quantum optimization. *Inform. Sci.* **128**, 181–197 (2000).
- Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028* (2014).
- Cerezo, M. et al. Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625–644 (2021).
- Rattew, A. G., Hu, S., Pistoia, M., Chen, R. & Wood, S. A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver. <https://arxiv.org/abs/1910.09694> (2019).
- Kandala, A. et al. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **549**, 242–246 (2017).
- Shaydulin, R. & Pistoia, M. QAOA with $n \cdot p \geq 200$. *arXiv:2303.02064* (2023).
- You, X., Chakrabarti, S. & Wu, X. A convergence theory for over-parameterized variational quantum eigensolvers. *arXiv:2205.12481* (2022).
- Boulebnane, S. & Montanaro, A. Solving Boolean satisfiability problems with the quantum approximate optimization algorithm. <https://doi.org/10.48550/arXiv.2208.06909> (2022).
- Shaydulin, R. et al. Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem. *arXiv preprint arXiv:2308.02342* (2023).
- Wang, Z., Rubin, N. C., Dominy, J. M. & Rieffel, E. G. XY-mixers: Analytical and numerical results for the quantum alternating operator ansatz. *Phys. Rev. A* **101**, <https://doi.org/10.1103/physreva.101.012320> (2020).
- Niroula, P. et al. Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer. *Sci. Rep.* **12**, 17171 (2022).
- Hadfield, S. Quantum algorithms for scientific computing and approximate optimization. Columbia university PhD dissertation, *arXiv:1805.03265* (2018).
- Hadfield, S. et al. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* **12**, 34 (2019).
- Leipold, H. & Spedalieri, F. M. Constructing driver Hamiltonians for optimization problems with linear constraints. *Quantum Sci. Technol.* **7**, 015013 (2021).
- Stollenwerk, T., Hadfield, S. & Wang, Z. Toward quantum gate-model heuristics for real-world planning problems. *IEEE Transac. Quantum Eng.* **1**, 1–16 (2020).
- Hen, I. & Sarandy, M. S. Driver hamiltonians for constrained optimization in quantum annealing. *Phys. Rev. A* **93**, <https://doi.org/10.1103/physreva.93.062312> (2016).
- Cook, J., Eidenbenz, S. & Bärttschi, A. The quantum alternating operator ansatz on maximum k-vertex cover. *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 83–92 (2020).
- Fuchs, F. G., Lye, K. O., Nilsen, H. M., Stasik, A. J. & Sartor, G. Constrained mixers for QAOA. *arXiv preprint arXiv:2203.06095* (2022).
- LaRose, R., Rieffel, E. & Venturelli, D. Mixer-phaser ansätze for quantum optimization with hard constraints. *arXiv preprint arXiv:2107.06651* (2021).
- Bärttschi, A. & Eidenbenz, S. Grover mixers for qaoa: Shifting complexity from mixer design to state preparation. *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 72–82 (2020).
- Gilliam, A., Woerner, S. & Goniçulea, C. Grover adaptive search for constrained polynomial binary optimization. *Quantum* **5**, 428 (2021).
- Golden, J., Bärttschi, A., O'Malley, D. & Eidenbenz, S. Threshold-based quantum optimization. *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 137–147 (2021).
- Marsh, S. & Wang, J. B. A quantum walk-assisted approximate algorithm for bounded NP optimisation problems. *Quantum Inform. Process.* **18**, 1–18 (2019).
- Marsh, S. & Wang, J. B. Combinatorial optimization via highly efficient quantum walks. *Phys. Rev. Res.* **2**, <https://doi.org/10.1103/physrevresearch.2.023302> (2020).
- Slate, N., Matwiejew, E., Marsh, S. & Wang, J. B. Quantum walk-based portfolio optimisation. *Quantum* **5**, 513 (2021).
- Ender, K., ter Hoeven, R., Niehoff, B. E., Drieb-Schön, M. & Lechner, W. Parity quantum optimization: Compiler. <https://arxiv.org/abs/2105.06233> (2021).
- Drieb-Schön, M., Javanmard, Y., Ender, K. & Lechner, W. Parity quantum optimization: Encoding constraints. <https://arxiv.org/abs/2105.06235> (2021).
- Fellner, M., Ender, K., ter Hoeven, R. & Lechner, W. Parity quantum optimization: Benchmarks. <https://arxiv.org/abs/2105.06240> (2021).
- Ender, K., Messinger, A., Fellner, M., Dlska, C. & Lechner, W. Modular parity quantum approximate optimization. *PRX Quantum* **3**, 030304 (2022).
- Dominguez, F. et al. Encoding-independent optimization problem formulation for quantum computing. <https://arxiv.org/abs/2302.03711> (2023).
- Liu, X. et al. Layer VQE: A variational approach for combinatorial optimization on noisy quantum computers. *IEEE Transactions on Quantum Engineering* **3**, 1–20 (2022).
- Childs, A. M. Lecture notes on quantum algorithms. *Lecture notes at University of Maryland*. <https://www.cs.umd.edu/~amchilds/qa/qa.pdf> (2017).
- Dollard, J. D. & Friedman, C. N. *Product Integration with Application to Differential Equations*. Encyclopedia of Mathematics and its Applications (Cambridge University Press, 1984).
- McClean, J. R. et al. Low-depth mechanisms for quantum optimization. *PRX Quantum* **2**, <https://doi.org/10.1103/prxquantum.2.030312> (2021).
- Powell, M. J. D. *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*. (Springer Netherlands, Dordrecht, 1994).
- Herman, D. et al. Quantum computing for finance. *Nat. Rev. Phys.* <https://doi.org/10.1038/s42254-023-00603-1> (2023).
- Yalovetzky, R., Minssen, P., Herman, D. & Pistoia, M. NISQ-HHL: Portfolio optimization for near-term quantum hardware. <https://arxiv.org/abs/2110.15958> (2021).
- He, Z. et al. Alignment between initial state and mixer improves QAOA performance for constrained portfolio optimization. *arXiv preprint arXiv:2305.03857* (2023).

39. Markowitz, H. M. Portfolio selection. *J. Financ.* **7**, 77–91 (1952).
40. Boulebnane, S., Lucas, X., Meyder, A., Adaszewski, S. & Montanaro, A. Peptide conformational sampling using the quantum approximate optimization algorithm. *arXiv preprint arXiv:2204.01821* (2022).
41. Shaydulin, R., Lotshaw, P. C., Larson, J., Ostrowski, J. & Humble, T. S. Parameter transfer for quantum approximate optimization of weighted maxcut. *arXiv preprint arXiv:2201.11785* (2022).
42. Sureshbabu, S. H. et al. Parameter setting in quantum approximate optimization of weighted problems. <https://doi.org/10.48550/arXiv.2305.15201> (2023).
43. Khairy, S., Shaydulin, R., Cincio, L., Alexeev, Y. & Balaprakash, P. Learning to optimize variational quantum circuits to solve combinatorial problems. *Proc. AAAI Conf. Artif. Intell.* **34**, 2367–2375 (2020).
44. Shaydulin, R. & Galda, A. Error mitigation for deep quantum optimization circuits by leveraging problem symmetries. <https://doi.org/10.1109/qce52317.2021.00046> (2021).
45. Kakkar, A., Larson, J., Galda, A. & Shaydulin, R. Characterizing error mitigation by symmetry verification in QAOA. <https://arxiv.org/abs/2204.05852> (2022).
46. Takagi, R., Endo, S., Minagawa, S. & Gu, M. Fundamental limits of quantum error mitigation. *npj Quantum Inform.* **8**, 114 (2022).
47. Farhi, E., Gamarnik, D. & Gutmann, S. The quantum approximate optimization algorithm needs to see the whole graph: A typical case. <https://doi.org/10.48550/arXiv.2004.09002> (2020).
48. Sanders, Y. R. et al. Compilation of fault-tolerant quantum heuristics for combinatorial optimization. *PRX Quantum* **1**, 020312 (2020).
49. Misra, B. & Sudarshan, E. C. G. The Zeno's paradox in quantum theory. *J. Math. Phys.* **18**, 756–763 (1977).
50. Presilla, C., Onofrio, R. & Tambini, U. Measurement quantum mechanics and experiments on quantum Zeno effect. *Annal. Phys.* **248**, 95–121 (1996).
51. Huggett, N. Zeno's Paradoxes. In Zalta, E. N. *The Stanford Encyclopedia of Philosophy* (Metaphysics Research Lab, Stanford University, 2019), Winter 2019 edn.
52. Facchi, P., Gorini, V., Marmo, G., Pascazio, S. & Sudarshan, E. Quantum Zeno dynamics. *Phys. Lett. A* **275**, 12–19 (2000).
53. Facchi, P. & Pascazio, S. Quantum Zeno subspaces. *Phys. Rev. Lett.* **89**, <https://doi.org/10.1103/PhysRevLett.89.080401> (2002).
54. Facchi, P. & Pascazio, S. Quantum Zeno dynamics: mathematical and physical aspects. *J. Phys. A* **41**, 493001 (2008).
55. Burgarth, D., Facchi, P., Nakazato, H., Pascazio, S. & Yuasa, K. Quantum Zeno dynamics from general quantum operations. *Quantum* **4**, 289 (2020).
56. Childs, A. M. et al. Quantum search by measurement. *Phys. Rev. A* **66**, 032314 (2002).
57. Aharonov, D. & Ta-Shma, A. Adiabatic quantum state generation and statistical zero knowledge. <https://arxiv.org/abs/quant-ph/0301023> (2003).
58. Somma, R., Boixo, S. & Barnum, H. Quantum simulated annealing. <https://arxiv.org/abs/0712.1008> (2007).
59. Somma, R. D., Boixo, S., Barnum, H. & Knill, E. Quantum simulations of classical annealing processes. *Phys. Rev. Lett.* **101**, <https://doi.org/10.1103/PhysRevLett.101.130504> (2008).
60. Boixo, S., Knill, E. & Somma, R. Eigenpath traversal by phase randomization. *Quantum Info. Comput.* **9**, 833–855 (2009).
61. Subaşı, Y., Somma, R. D. & Orsucci, D. Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing. *Phys. Rev. Lett.* **122**, 060504 (2019).
62. Lin, L. & Tong, Y. Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems. *Quantum* **4**, 361 (2020).
63. Viola, L. & Lloyd, S. Dynamical suppression of decoherence in two-state quantum systems. *Phys. Rev. A* **58**, 2733–2744 (1998).
64. Facchi, P., Lidar, D. A. & Pascazio, S. Unification of dynamical decoupling and the quantum Zeno effect. *Phys. Rev. A* **69**, <https://doi.org/10.1103/PhysRevA.69.032314> (2004).
65. Halimeh, J. C., Lang, H., Mildenberger, J., Jiang, Z. & Hauke, P. Gauge-symmetry protection using single-body terms. *PRX Quantum* **2**, 040311 (2021).
66. Halimeh, J. C., Zhao, H., Hauke, P. & Knolle, J. Stabilizing disorder-free localization. <https://arxiv.org/abs/2111.02427> (2021).
67. Halimeh, J. C. et al. Stabilizing lattice gauge theories through simplified local pseudogenerators. *Phys. Rev. Res.* **4**, 033120 (2022).
68. Halimeh, J. C. et al. Enhancing disorder-free localization through dynamically emergent local symmetries. *PRX Quantum* **3**, 020345 (2022).
69. Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, vol. 87 (Springer Science & Business Media, 2003).
70. Childs, A. M., Su, Y., Tran, M. C., Wiebe, N. & Zhu, S. Theory of trotter error with commutator scaling. *Physical Review X* **11**, 011020 (2021).
71. O'Donnell, R. *Analysis of Boolean Functions* (Cambridge University Press, 2014).
72. Häner, T., Roetteler, M. & Svore, K. M. Factoring using $2n+2$ qubits with Toffoli based modular multiplication. <https://arxiv.org/abs/1611.07995> (2016).
73. Häner, T., Rötteler, M. & Svore, K. M. Optimizing quantum circuits for arithmetic. *arXiv preprint arXiv:1805.12445* (2018).
74. Häner, T., Soeken, M., Roetteler, M. & Svore, K. M. Quantum circuits for floating-point arithmetic. <https://arxiv.org/abs/1807.02023> (2018).
75. Gidney, C. Halving the cost of quantum addition. *Quantum* **2**, 74 (2018).
76. Jones, C. Low-overhead constructions for the fault-tolerant Toffoli gate. *Phys. Rev. A* **87**, <https://doi.org/10.1103/physreva.87.022328> (2013).
77. Maslov, D. Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization. *Phys. Rev. A* **93**, 022311 (2016).
78. Draper, T. G. Addition on a quantum computer. <https://arxiv.org/abs/quant-ph/0008033> (2000).
79. Ruiz-Perez, L. & Garcia-Escartin, J. C. Quantum arithmetic with the quantum Fourier transform. *Quantum Inform. Process.* **16**, <https://doi.org/10.1007/s11128-017-1603-1> (2017).
80. Şahin, E. Quantum arithmetic operations based on quantum Fourier transform on signed integers. *Int. J. Quantum Inform.* **18**, 2050035 (2020).
81. Nielsen, M. A. & Chuang, I. *Quantum computation and quantum information* (Cambridge University Press, 2010).
82. Bocharov, A., Roetteler, M. & Svore, K. M. Efficient synthesis of universal repeat-until-success quantum circuits. *Phys. Rev. Lett.* **114**, <https://doi.org/10.1103/physrevlett.114.080502> (2015).
83. Nam, Y., Su, Y. & Maslov, D. Approximate quantum Fourier transform with $O(n \log(n))$ T gates. *npj Quantum Inform.* **6**, <https://doi.org/10.1038/s41534-020-0257-5> (2020).
84. Griffiths, R. B. & Niu, C.-S. Semiclassical Fourier transform for quantum computation. *Phys. Rev. Lett.* **76**, 3228 (1996).
85. Parker, S. & Plenio, M. B. Efficient factorization with a single pure qubit and $\log n$ mixed qubits. *Phys. Rev. Lett.* **85**, 3049–3052 (2000).
86. Goto, H. Resource requirements for a fault-tolerant quantum fourier transform. *Phys. Rev. A* **90**, 052318 (2014).
87. Bärtschi, A. & Eidenbenz, S. Short-depth circuits for Dicke state preparation. *arXiv preprint arXiv:2207.09998* (2022).
88. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J. & Killoran, N. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A* **99**, 032331 (2019).
89. Wierichs, D., Izaac, J., Wang, C. & Lin, C. Y.-Y. General parameter-shift rules for quantum gradients. *Quantum* **6**, 677 (2022).

Acknowledgements

The authors wish to thank Antonio Mezzacapo from IBM for his invaluable contributions to this project. Special thanks also to Tony Uttley, Jenni Strabley and Brian Neyenhuis from Quantinuum for their assistance on the execution of the experiments on the Quantinuum H1-2 trapped-ion quantum processor. *Disclaimer:* This paper was prepared for information purposes with contributions from the Global Technology Applied Research center of JPMorgan Chase. This paper is not a product of the Research Department of JPMorgan Chase or its affiliates. Neither JPMorgan Chase nor any of its affiliates make any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, but not limited to, the completeness, accuracy, reliability of information contained herein and the potential legal, compliance, tax or accounting effects thereof. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

Author contributions

M.P. led the overall project. D.H., R.S., Y.S., and R.Y. developed the simulation code and performed numerical experiments. D.H. performed the experiments on trapped-ion quantum processors. D.H., R.S., Y.S., S.C., and A.R. developed the theoretical results. S.H. and P.M. contributed to technical discussions. All authors contributed to the writing of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Dylan Herman.

Peer review information *Communications Physics* thanks the anonymous reviewers for their contribution to the peer review of this work.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023