Editors' Suggestion

Scalable Mitigation of Measurement Errors on Quantum Computers

Paul D. Nation,* Hwajung Kang, Neereja Sundaresan, and Jay M. Gambetta
IBM Quantum, Yorktown Heights, New York 10598, USA

(Received 6 September 2021; accepted 8 October 2021; published 8 November 2021)

We present a method for mitigating measurement errors on quantum computing platforms that does not form the full assignment matrix, or its inverse, and works in a subspace defined by the noisy input bit strings. This method accommodates both uncorrelated and correlated errors and allows for the computation of accurate error bounds. Additionally, we detail a matrix-free preconditioned iterative-solution method that converges in $\mathcal{O}(1)$ steps that is performant and uses orders of magnitude less memory than direct factorization. We demonstrate the validity of our method and mitigate errors in a few seconds on numbers of qubits that would otherwise be impractical.

DOI: 10.1103/PRXQuantum.2.040326

I. INTRODUCTION

Recently, rapid developments in the fabrication, control, and deployment of quantum computing systems have brought qubit counts to approximately 100, where it might be possible to show advantage over classical computation methods in one or more limited cases [1–3]. However, such breakthroughs are hampered by noise and errors that conspire to limit the effectiveness of quantum computers at tackling problems of appreciable scale. To counteract these effects, researchers have turned to mitigation methods that approximately cancel quantum gate [4–10] and measurement assignment [11–20] errors. For short-depth quantum circuits that can be executed on current-generation hardware, measurement errors play an outsized role and their correction is critical to many near-term experiments [21–28].

In the canonical situation where initialization noise is minimal, measurement errors over N qubits can be treated classically and satisfy

$$\vec{p}_{\text{noisy}} = A\vec{p}_{\text{ideal}}, \tag{1}$$

where $\vec{p}_{\text{noisy}}$ is a vector of noisy probabilities returned by the quantum system, $\vec{p}_{\text{ideal}}$ is the probabilities in the absence of measurement errors (but still includes, e.g., gate errors), and A is the $2^N \times 2^N$ complete assignment matrix (A-matrix), where element $A_{\texttt{row},\texttt{col}}$ is the probability of bit string `col` being converted to bit string `row` by the measurement-error process (for examples, see Appendix A). While computing A requires executing 2^N circuits, it is often the case that errors on multiple qubits can be well approximated using at most $\mathcal{O}(N)$ calibration circuits.

Equation (1) has a solution $\vec{p}_{\text{ideal}}$ that is readily found using direct lower-upper (LU) factorization. However, direct methods necessarily generate quasiprobability distributions due to finite sampling that contains negative values but still sums to one, these values being incompatible with the requirement of $\vec{p}_{\text{ideal}}$ being a probability vector. Consequently, a bounded-minimization approach solving $||A\vec{p}_{\text{ideal}} - \vec{p}_{\text{noisy}}||_2^2$, where $\vec{p}_{\text{ideal}}$ is constrained to be positive, is often used in place of a direct solution [12–14,16,29]. Although physically appealing, the run times of these methods are orders of magnitude longer than those of direct techniques. Alternatively, it has been shown that quasiprobabilities can be used provided that one mitigates expectation values [4,11,30]. As proven in Ref. [11], these quasiprobabilities provide an unbiased estimate for the expectation value ξ of an operator O, with a spectral radius of one, that is diagonal in the computational basis

$$\xi = \sum_{i=0}^{2^N-1} \left[OA^{-1}\vec{p}_{\text{noisy}}\right]_i. \tag{2}$$

Near-term algorithms such as the ubiquitous variational quantum eigensolver (VQE) [21,31] and quantum machine learning [23,28] rely on the computation of expectation values, making the correction of measurement errors in these quantities an important step along the road to quantum advantage.

Current measurement-mitigation techniques utilize the full 2^N-dimensional probability space and thus do not scale beyond a handful of qubits. A truncation scheme

*paul.nation@ibm.com

TABLE I. Representative error rates for Cloud-accessible quantum computing systems.

System	Average assignment error (%)
Alibaba 11Q [32]	7.4
Google 53Q Sycamore [33]	3.2
IBM 27Q Falcon_R5.11 [34]	1.1
IONQ 11Q [35]	0.4
Quantum Inspire 5Q Starmon [36]	4.0
Rigetti 32Q Aspen-9 [35]	6.1

has been developed in Ref. [24]; however, it did so at the cost of a loss of measurement information and it still required explicit construction of the full A-matrix. The creation of a scalable mitigation strategy requires reduction of the dimensionality of the linear system in Eq. (1) without the need for computing A itself. Fortunately, present-day Cloud-accessible quantum computing systems have measurement-error rates of a few percent or less (see Table I), indicating that it is possible to view the measurement-error process as a small correction to the ideal probability distribution; measurement errors redistribute small fractions of probability from a given bit string primarily to those that are a short Hamming distance away (see, e.g., Appendix A). To good approximation, the solution is contained within $\vec{p}_{\text{noisy}}$ and we can mitigate errors in a renormalized subspace defined by these bit strings. In the worst case, this subspace dimension is equal to the number of times the input circuit is sampled. For Cloud-accessible quantum computers, the number of times a circuit can be sampled is limited—typically 8192 times on IBM Quantum systems—and therefore the dimensionality of the corresponding reduced assignment matrix $\tilde{A}$ can be markedly smaller than the full A-matrix for N qubits. In practice, $\tilde{A}$ is often small enough such that the solution is amenable to standard LU factorization, returning a vector of quasiprobabilities for use in a reduced version of Eq. (2).

However, for situations where explicitly forming $\tilde{A}$ is still prohibitive due to large numbers of unique samples, it is possible to use preconditioned matrix-free iterative linear-solution methods. Such methods have also been explored in numerical solutions of large-scale steady-state density matrices [37]. In practice, this gives quick convergence, typically in $\mathcal{O}(1)$ steps, and is competitive with direct solution run times while requiring orders of magnitude less memory. Although the methods introduced here return quasiprobabilities, it is possible to find the nearest probability distribution, in terms of the L^2-norm, in linear time [38].

In this paper, we detail this efficient mitigation method, beginning with Sec. II, which motives the subspace reduction procedure and describes how it is performed. Section III shows how preconditioned matrix-free methods can be utilized for a performant and memory-efficient solution technique. In Sec. IV, we show that one can obtain bounds on the variance of the computed expectation values in a similarly efficient manner with an overhead of $\mathcal{O}(1)$ in terms of additional run time. We demonstrate our technique in Sec. V, showing the validity of our method and mitigating readout errors out to numbers of qubits that would be intractable on even the largest of supercomputers using previous methods. Finally, Sec. VI ends with a brief summary and discusses possible future directions.

II. SUBSPACE REDUCTION

We aim to construct $\tilde{A}$ without necessarily forming the full A-matrix. To this end, we look to compute elements $A_{\texttt{row,col}}$, directly from the bit-string values of the input noisy counts and a small set of calibration-data matrices. For concreteness, we study the case of tensored measurement errors, as IBM Quantum systems are calibrated for high-fidelity quantum nondemolition (QND) measurements, where uncorrelated errors are nominally dominant. Other quantum hardware vendors also report the same [27]. The full-dimensional tensored A-matrix $A^{(T)}$ over N qubits can be constructed from N 2×2 calibration matrices: $A^{(T)} = S_{N-1} \otimes \ldots S_1 \otimes S_0$, where S_k is the calibration matrix for the kth qubit with the form

$$S_k = \begin{bmatrix} P^{(k)}_{0,0} & P^{(k)}_{0,1} \\ P^{(k)}_{1,0} & P^{(k)}_{1,1} \end{bmatrix}, \tag{3}$$

where $P^k_{i,j}$ is the probability of the kth qubit being in state $j \in \{0,1\}$ and measured in state $i \in \{0,1\}$. Here, we use the convention that qubit 0 corresponds to the least-significant bit. For two bit strings, $\texttt{row}, \texttt{col} \in \{0,1\}^N$, the matrix element $A^{(T)}_{\texttt{row,col}}$ can be computed using

$$A^{(T)}_{\texttt{row,col}} = \prod_{k=0}^{N-1} P^{(k)}_{\texttt{row}[N-1-k],\texttt{col}[N-1-k]}. \tag{4}$$

Therefore it is possible to compute individual matrix elements directly from bit-string values and a number of calibration matrices that scales at most linearly with the number of qubits. The accommodation of correlated errors in our method simply amounts to finding an equivalent expression to Eq. (4). We give an example in Appendix B, where we intentionally induce correlated errors into the measurement process. As a corollary of grabbing elements individually, it is possible to select only those elements within a given Hamming distance, $d(\texttt{row}, \texttt{col}) \leq D$, where D is the desired maximum distance. This allows for varying the sparsity of $\tilde{A}$ and examining the effect of low-distance approximations.

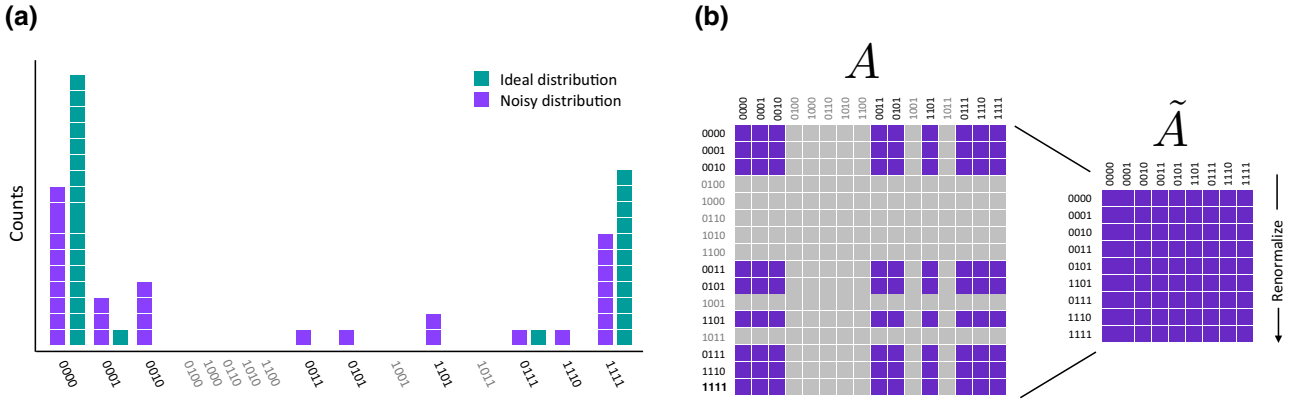

FIG. 1. (a) A simulated discrete probability distribution for 30 samples, showing the ideal distribution for a Greenberger-Horne-Zeilinger (GHZ) state subject to gate errors only as well as noisy data affected by both gate and measurement errors. Observed bit strings are in bold. The same random number seed is used for both simulations, such that the difference between ideal and noisy distributions is solely due to measurement errors. (b) The elements of the assignment matrix A that are used for constructing the reduced assignment matrix $\tilde{A}$ that corresponds to the noisy distribution from (a). The columns of $\tilde{A}$ must be renormalized such that they again sum to one [Eq. (5)].

$\tilde{A}$ is defined to be the assignment matrix over only those bit strings observed in $\vec{p}_{\text{noisy}}$. To understand the validity of this reduction, consider the simulation presented in Fig. 1(a), where we compare the distributions for $\vec{p}_{\text{ideal}}$ and $\vec{p}_{\text{noisy}}$. In effect, measurement errors take blocks of probability, as defined by the finite number of samples, from $\vec{p}_{\text{ideal}}$ and redistribute them amongst other bit strings to obtain $\vec{p}_{\text{noisy}}$. This redistribution is predominantly to those bit strings that are close in Hamming distance. Importantly, as seen in Fig. 1(a), when measurement errors are small and a sufficient number of circuit samples is performed, it is unlikely that a given bit string in $\vec{p}_{\text{ideal}}$ is completely redistributed; the bit strings in $\vec{p}_{\text{ideal}}$ are also contained in $\vec{p}_{\text{noisy}}$. Therefore, the mitigation of measurement errors requires only elements $A_{\texttt{row,col}}$ that correspond to transitions between bit strings in $\vec{p}_{\text{noisy}}$. An example is shown in Fig. 1(b). Because we are grabbing only select elements of the full A-matrix, the columns of $\tilde{A}$ must be renormalized such that they once again sum to one. That is to say, given any two bit strings $\texttt{row}, \texttt{col} \in \vec{p}_{\text{noisy}}$, the reduced matrix element $\tilde{A}_{\texttt{row,col}}$ is given by

$$\tilde{A}_{\texttt{row,col}} = \begin{cases} \dfrac{A_{\texttt{row,col}}}{\sum\limits_{\substack{k\in\vec{p}_{\text{noisy}} \\ d(k,\texttt{col})\le D}} A_{k,\texttt{col}}}, & d(\texttt{row,col})\le D, \\ 0, & d(\texttt{row,col})>D, \end{cases} \tag{5}$$

where D is the desired Hamming distance [39]. Performing a finite number of circuit executions, even when measurement errors are weak, may result in completely redistributing probability away from some small-magnitude elements in $\vec{p}_{\text{ideal}}$ such that those elements are not present in $\vec{p}_{\text{noisy}}$; the solution vector will be missing these elements. However, as we show, for typical numbers of circuit samples, this effect is minimal.

It is important to note that the restriction that $\vec{p}_{\text{ideal}}$ be contained in $\vec{p}_{\text{noisy}}$ is not unique to our method and is also enforced in bounded least-squares. While in our technique this is motivated by physical understanding, in bounded least-squares solutions the positivity constraints prohibit the free reallocation of probabilities amongst bit strings and redistribution of probabilities occurs only amongst elements originally in $\vec{p}_{\text{noisy}}$ [40]. In the limit of only a single bit string in $\vec{p}_{\text{noisy}}$, either from a single-shot or fortuitous sampling, neither our method nor bounded least-squares provides any mitigation. Conversely, when probing very wide distributions, insufficient sampling leads to a $\vec{p}_{\text{noisy}}$ composed of bit strings that are often far apart in Hamming distance, leading to negligible mitigation. Thus both methods require accurate sampling of the distributions over which they are to mitigate. As the same holds true for computing accurate expectation values, this does represent a fundamental limitation.

III. MATRIX-FREE SOLUTION

Although $\tilde{A}$ is much smaller than the original, when sampling circuits with wide probability distributions many times or executing on systems with appreciable error rates, it is possible that $\tilde{A}$ itself may become too costly to explicitly construct. Fortunately, being able to grab elements of A individually [Eq. (4)] allows us to take advantage of matrix-free iterative techniques [41]. The time to solution for iterative methods greatly depends on the properties of $\tilde{A}$. A-matrices obtained from present-day Cloud-accessible platforms nominally have strict diagonal dominance $|A_{\texttt{row,row}}| > \sum_{\texttt{col}\neq\texttt{row}} |A_{\texttt{row,col}}| \forall \texttt{row}$

and are readily solved by simple iterative methods such as Jacobi iteration [41]. However, this condition does not hold in general for systems with large error rates. Moreover, as the number of qubits grows, so does the number of possible error channels (i.e., the number of states at low Hamming distance) and it becomes harder to satisfy this stringent condition. Thus general-purpose methods such as generalized-minimal-residual (GMRES) [42] or biconjugate-gradient-stabilized (Bi-CGSTAB) [43] methods must be used. Importantly, these Krylov-subspace methods require the computation of only the product $\tilde{A}\vec{p}_{\text{noisy}}$ but not $\tilde{A}$ itself [41,44]. However, having strict diagonal dominance, or close to it, suggests that we can increase the rate of convergence by using a simple Jacobi preconditioner P^{-1} to solve

$$P^{-1}\tilde{A}\vec{x} = P^{-1}\vec{p}_{\text{noisy}}, \tag{6}$$

where P^{-1} is a diagonal matrix with $P^{-1}_{i,i} = 1/\tilde{A}_{i,i}$ [41]. In practice, Eq. (6) gives rapid convergence, requiring only $\mathcal{O}(1)$ iterations for an absolute tolerance value of 10^{-5} while simultaneously dramatically reducing the memory requirements for mitigation.

IV. UNCERTAINTY ESTIMATES

The mitigation of measurement errors does not come for free. Rather, it results in an increase in the uncertainty of repeated-measurement outcomes that must be compensated for by increasing the number of times the circuit is sampled. This mitigation overhead $\mathcal{M}$ is determined by the one-norm of the inverse of the A-matrix $\mathcal{M} = ||A^{-1}||_1^2$ [11] and gives an upper bound on the standard deviation of an observable $\sigma_O \leq \sqrt{\mathcal{M}/s}$, where s is the number of samples. Not wanting to construct $\tilde{A}^{-1}$, here we use the iterative Hager-Higham algorithm [45,46] for estimating $||\tilde{A}^{-1}||_1$ using only linear systems of equations involving $\tilde{A}$ and $\tilde{A}^T$. When using direct factorization, the LU decomposition of $\tilde{A}$ can be cached and thus the overall run time is approximately $2\times$ longer than mitigation alone. However, for iterative methods, the overhead is between $4\times$ and $10\times$ longer, depending on how many steps the Hager-Higham routine requires. Although this method gives a lower bound on $\sqrt{\mathcal{M}}$, in practice it is often exact or nearly so (see the related discussion in Ref. [46]). Because our truncation method selects only those rows and columns from $\vec{p}_{\text{noisy}}$, the one-norm of $||\tilde{A}^{-1}||_1$, and thus the mitigation overhead, is dependent on the circuit being executed and the noise properties of the device on which it is run.

V. DEMONSTRATIONS

Our method [47] is implemented with NumPy [48], SciPy [49], and CYTHON [50], and makes use of QISKIT [29] for calibration-circuit construction and execution. All timing data are taken on a quad-core Intel i3-10100 system with 32 Gb of memory, with NumPy and SciPy compiled against OpenBlas [51].

We begin by showcasing the veracity of our method by comparing expectation values for the circuit shown in Fig. 2(a), computed with the full-space tensored method from QISKIT IGNIS [29] along with our method, called M3 [52], varying the number of samples taken per circuit. Here, the circuits are executed on the 27-qubit IBM Quantum Kolkata system, mapping the virtual circuit qubits to physical qubits $[1, 4, 7, 10, 12, 13, 14, 11, 8, 5, 3, 2]$. The calibration-data and raw-input samples are identical for both mitigation methods. In Fig. 2(b), we see that,

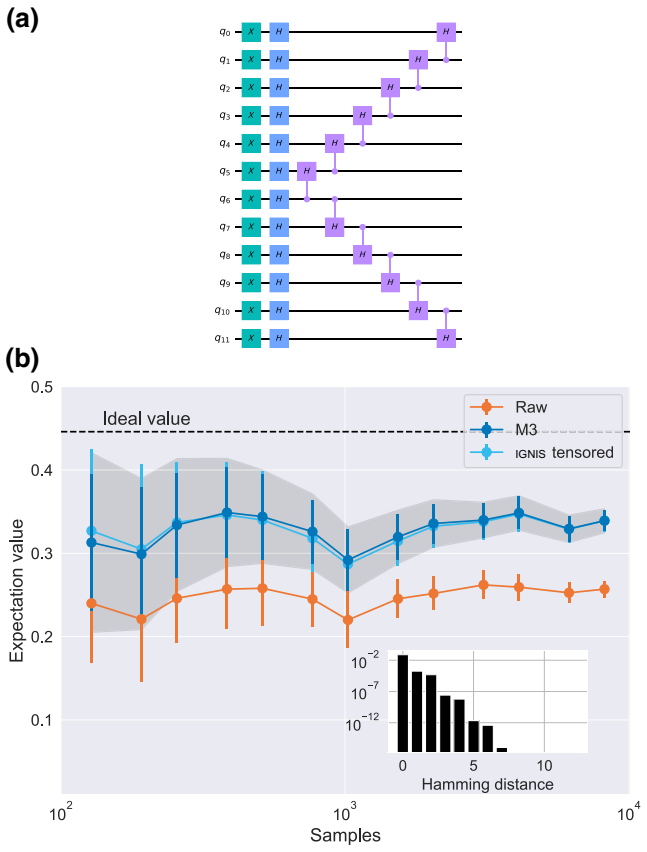

FIG. 2. (a) A 12-qubit circuit with an exact distribution of 43 unique bit strings and an expectation value of approximately 0.446. Measurements are omitted for brevity. (b) The expectation values before and after mitigation of measurement errors for the circuit in (a) executed 100 times on the IBM Quantum Kolkata system while varying the number of samples per circuit. The error bars show one standard deviation, while the shaded region gives the error bounds determined by the average of the computed mitigation overheads and the number of samples. The same calibration data are used for both IGNIS and M3 and are computed using 8192 samples per circuit. The inset shows the absolute error when truncating $\tilde{A}^{(T)}$ to a given Hamming distance for the data collected at 8192 samples.

despite using *at most* 371 bit strings (9% of the full-dimensionality), M3 closely matches the full-dimensional $A^{(T)}$ results over the entire range of samples and agrees remarkably well with the IGNIS results for the numbers of executions typically employed in practice, > 1000. An example $\tilde{A}^{(T)}$ generated by M3 is presented in Appendix 3. The subspace reduction results in a run-time performance improvement as well, with QISKIT IGNIS mitigation taking approximately 3 s per circuit, whereas M3 takes at most approximately 7 ms. Additionally, we see that the uncertainty bound given by $\mathcal{M}$ closely matches the experimental values and verifies the use of this technique in reporting faithful error bounds. The inset of Fig. 2(b) also shows that, for a tolerance of $\leq 10^{-5}$, $\tilde{A}^{(T)}$ is well approximated by keeping only terms out to $D = 3$.

We now demonstrate the scalability of our M3 method by mitigating GHZ states out to 42 qubits on the 65-qubit IBM Quantum Brooklyn system. Details of this experiment are given in Appendix C. In Fig. 3, we compare M3 along with the QISKIT IGNIS tensored and bounded least-squares methods [53]. Only M3 allows for the mitigation of errors beyond 14 qubits due to algorithmic breakdown or extreme run times, for the tensored and least-squares methods, respectively, and shows the importance of performing measurement mitigation for large-scale experiments. The overall expectation values drop as the circuit depth increases, where gate errors and decoherence, effects that measurement mitigation cannot resolve, start to dominate.

The mitigation overhead [see the inset of Fig. 3(a)] shows exponential scaling at small numbers of qubits, after which the overhead begins to plateau. The exponential scaling arises as the diagonal elements of $\tilde{A}$, formed from the product of N probabilities [Eq. (4)] are effectively being inverted when computing $\tilde{A}^{-1}$, with additional contributions coming from elements close in Hamming distance. Provided that the bit strings in $\vec{p}_{\text{noisy}}$ sample sufficient portions of these short-Hamming-distance elements, the renormalization used in obtaining $\tilde{A}$ is small and one recovers the exponential scaling shown for the full-dimensional A-matrix [11]. However, if this is not the case, then renormalization increases the magnitude of the elements in $\tilde{A}$ (in particular, the diagonal elements), suppressing the exponential growth in $\mathcal{M}$.

Figure 3(b) details the timing across the different mitigation methods. We see that M3 greatly improves the computed expectation values while taking at most 1.2 s to compute at 42 qubits. When computing the mitigation overhead, the total time increases to 2.4 and 4.5 s for the direct and iterative solutions at 42 qubits, respectively (not shown); an extraordinary improvement upon the exponential run times observed for the QISKIT methods. As with the example in Fig. 2, a $D = 3$ Hamming approximation well captures the full mitigation process to the desired tolerance and performs best at large numbers of bit strings, where the overhead from computing the Hamming distance between elements starts to become less than the cost of additional floating-point multiplications. For the M3 results, the rate at which the run times increase begins to slow down at larger numbers of qubits, following a similar slow down in the number of unique bit strings observed [see Fig. 3(a)] when additional qubits are added.

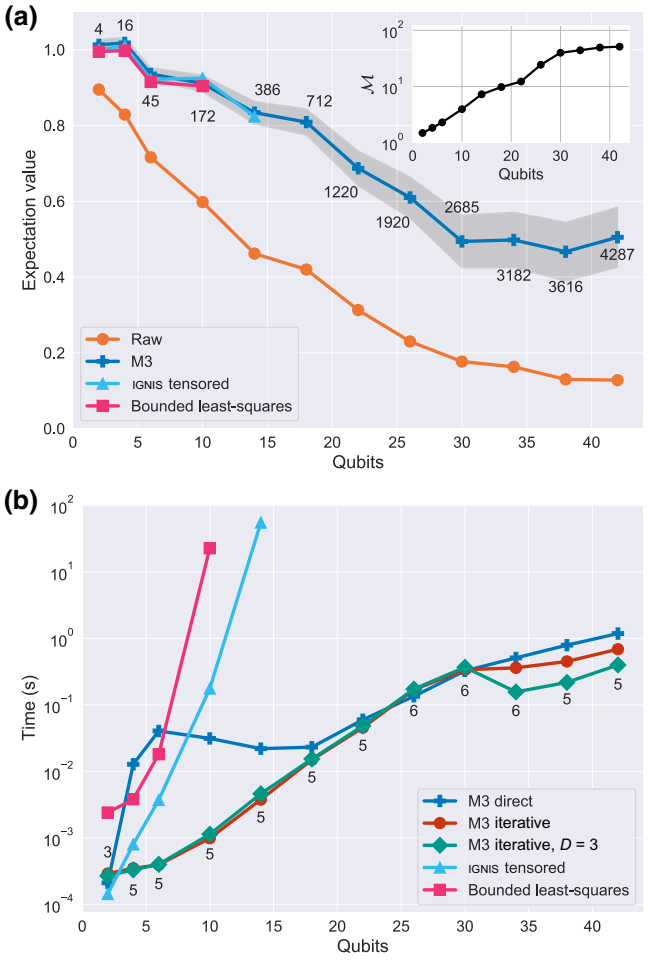

FIG. 3. (a) A comparison of the expectation values for GHZ states out to 42 qubits on the IBM Quantum Brooklyn system using M3 and the QISKIT IGNIS tensored and bounded least-squares methods. The M3 direct and iterative approaches and a $D = 3$ Hamming approximation all yield the same values to a tolerance of approximately 10^{-5} and the associated shaded region shows the error bound from $\mathcal{M}$. The numbers alongside the M3 data show the number of bit strings in $\vec{p}_{\text{noisy}}$ for each number of qubits. The inset shows the computed mitigation overhead $\mathcal{M}$. All calibration-data and GHZ-circuit execution is performed using 8192 shots per circuit. (b) The timing information (best of three runs) for the mitigation methods presented in (a). The annotated numerical values indicate the number of iterations needed for a tolerance $\leq 10^{-5}$. The full and $D = 3$ iterative solutions use the same number of iterations in all cases.

Finally, we note that at 42 qubits, the storage of a full 2^N-vector of single-precision floating-point values for $\vec{p}_{\text{noisy}}$ requires 16 TiB of memory, well beyond the limits of our computer on which the mitigation is implemented but amenable to storage on a supercomputer. Juxtapose that with storing a sparse representation of A using, for example, compressed-sparse-column (CSC) format out to $D = 3$. This requires approximately 580 PiB of memory, which is 120× more than that available in the Fugaku supercomputer [54] (for details, see Appendix 2). In contrast, the M3 iterative method uses approximately 1 MiB of storage, highlighting the benefit of the techniques presented here for mitigating measurements at scales amenable to demonstrations of quantum advantage.

VI. CONCLUSION

We demonstrate a method that circumvents the usual exponential overhead when performing measurement mitigation by working in a subspace defined by the noisy input bit strings. This technique bypasses the construction of the full-dimensional assignment matrix, and its inverse, by computing matrix elements directly from a small set of calibration matrices using bit strings to index the elements. The diagonally dominant structure of the A-matrix, combined with the ability to compute matrix elements individually, allows for utilizing matrix-free preconditioned iterative-solution methods that converge in $\mathcal{O}(1)$ steps. We show the validity of the subspace truncation method and demonstrate the scalability that it offers it terms of both memory and run time. In particular, we demonstrate measurement mitigation out to numbers of qubits that would otherwise be impractical on even the largest of supercomputers, yet remain well within reach of even modest computing resources using our method.

Looking forward, the combination of both measurement and gate-error mitigation at scales approaching quantum advantage is a promising avenue to explore. Indeed, scalable zero-noise extrapolation has recently been demonstrated [10] and the use of measurement mitigation in computing the expectation values needed in this method is one of many possibilities. With respect to the numerical method itself, the small memory footprint of the matrix-free iterative method, combined with the ability to compute matrix-vector products in parallel, suggests that this method can likely see additional run-time improvements by utilizing graphics processing unit (GPU) acceleration.

ACKNOWLEDGMENTS

We thank Doug McClure, David McKay, and Matthew Treinish for helpful discussions. Figures 2–7 are produced using MATPLOTLIB [55].

APPENDIX A: EXAMPLE A-MATRICES

1. Complete A-matrix

An example complete A-matrix computed by running 2^N circuits, one for each computational basis state, on the IBM Quantum Kolkata system is given in Fig. 4. Because of finite sampling, the matrix is nominally sparse and only those elements close in Hamming distance have appreciable transition probabilities. The matrix has strict diagonal dominance and thus is guaranteed to be invertible.

2. Tensored A-matrix

The A-matrix corresponding to tensored measurement errors, $A^{(T)}$, is constructed by taking the tensor product of single-qubit calibration matrices given by Eq. (3) in the main text. Unlike the complete A-matrix, $A^{(T)}$ contains only nonzero elements unless one or more qubits has no reported measurement error for $P_{0,1}^{(k)}$ and/or $P_{1,0}^{(k)}$; $A^{(T)}$ expects every element of $\vec{p}_{\text{noisy}}$ to have a nonzero entry. Like Fig. 4, the matrix in Fig. 5 is strictly diagonally dominant and indicates that transitions between elements close in Hamming distance are more likely. The data for Fig. 5 are taken immediately after those shown in Fig. 4.

3. Example 12-qubit truncated A-matrix

Figure 6 shows one of the 100 truncated $\tilde{A}^{(T)}$ used in the M3 mitigation performed in Fig. 2 at 8192 counts. This matrix is also strictly diagonally dominant. For each circuit execution, the number of elements in $\tilde{A}^{(T)}$ may vary, as can their associated amplitudes.

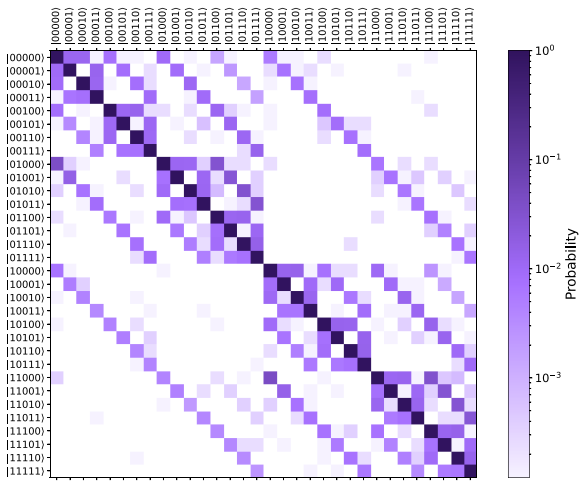

FIG. 4. The full A-matrix for qubits $0 \rightarrow 4$ on the IBM Quantum Kolkata system. A circuit for each of the 32 bit strings is executed 8192 times to fill in the columns.

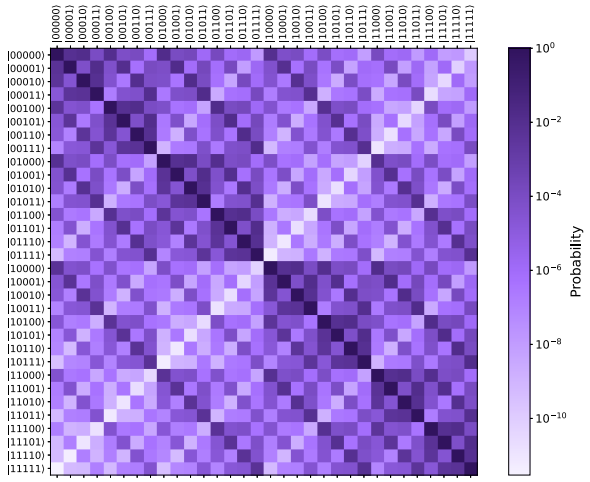

FIG. 5. The tensored A-matrix $A^{(T)}$ for qubits $0 \rightarrow 4$ on IBM Quantum Kolkata.

APPENDIX B: CORRELATED ERRORS

Although we focus on tensored mitigation techniques, our method is equally capable of handling correlated errors provided that matrix elements $\tilde{A}_{\mathtt{row,col}}$ can be obtained by their bit-string values as done in Eq. (4). IBM Quantum systems operating normally are dominated by uncorrelated measurement error and are thus well mitigated by the tensored A-matrix methods presented in the main text. It is possible, however, to purposely induce correlated errors in the readout process and explore correlated mitigation strategies.

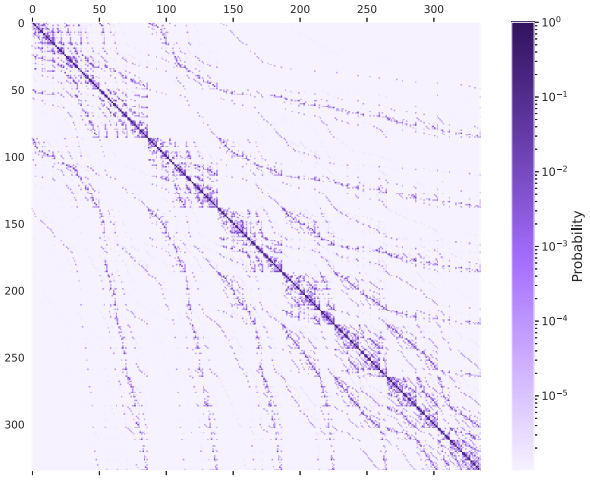

FIG. 6. A sample $\tilde{A}$ from the M3 results in Fig. 2(b), taken at 8192 samples. The matrix contains 341 bit strings out of a possible 4096. The rows and columns are sorted in lexicographical order in terms of their bit-string representation.

To take into account pairwise-correlated errors, we can modify Eq. (4), grabbing elements using

$$A^{(C)}_{\mathtt{row,col}} = \frac{1}{\binom{N}{2}} \sum_{k=1}^{N-1} \sum_{l=0}^{k-1} C_{kl}[a^{kl}, b^{kl}] \prod_{\substack{m=0 \\ m\neq k,l}}^{N-1} S_m[q_m, q'_m], \tag{B1}$$

where $\mathtt{row} = q_{N-1}q_{N-2}..q_0$ and $\mathtt{col} = q'_{N-1}q'_{N-2}..q'_0$. Here, the S_m are defined as in Eq. (3) and the C_{kl} are a $4x4$ stochastic matrix (local noise matrix) between qubits k and l, where $a^{kl} = 2q_k + q_l$, $b^{kl} = 2q'_k + q'_l$, respectively. The elements of C_{kl} are obtained following Sec. V of Ref. [11].

The IBM Quantum Kolkata, a Falcon 5.11 series system, has readout output multiplexing ratios ranging from 3 : 1 to 5 : 1, with readout frequencies in a shared output typically separated by 50–60 MHz. This separation is much larger than the average cavity line width κ (5.6 MHz) and dispersive shift χ (1.6 MHz), which when combined enable short 330-ns readout pulses for all qubits. Using default readout pulse amplitudes, calibrated to optimize fidelity while maintaining QND readout, Fig. 7(a) shows expectation values produced by executing an eight-qubit GHZ circuit 100 times using qubits [8, 5, 3, 2, 1, 4, 7, 6] that span two multiplex readout groupings. Importantly, we perform mitigation using the complete A-matrix, $\tilde{A}^{(C)}$, as well as $\tilde{A}^{(T)}$, with results in agreement, with uncorrelated errors largely dominating the readout process.

An intentional increase in the readout pulse amplitudes by approximately $2\times$ from the optimized values results

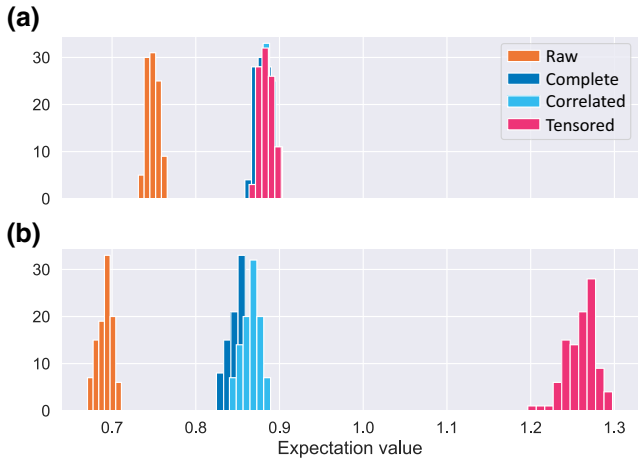

FIG. 7. (a) A histogram of the expectation values collected from executing an eight-qubit GHZ circuit 100 times on the IBM Quantum F5.11 Kolkata system under normal operating conditions and using qubits [8, 5, 3, 2, 1, 4, 7, 6] and 8192 samples per circuit. The raw results are mitigated with the complete (A), correlated M3 ($\tilde{A}^{(C)}$), and tensored M3 ($\tilde{A}^{(T)}$) A-matrices. (b) Repeated experimental results in the presence of correlated readout errors generated by degraded readout.

in correlated readout errors. With these larger readout amplitudes, while there is no appreciable change in the average readout fidelity ($< 0.2\%$) compared to the default setting, there is a substantial uptick in non-QNDness. In Fig. 7(b), we rerun our eight-qubit GHZ experiment from Fig. 7(a) under these new conditions. While the correlated M3 method using Eq. (B1) captures the correlated readout errors well, as is evident by the agreement with the complete A-matrix, the tensored mitigator strongly overcorrects due to the misalignment of probabilities in $\tilde{A}^{(T)}$ with those actually present in the system. Although Eq. (B1) works for both uncorrelated and pairwise-correlated errors, each matrix element requires $\mathcal{O}\left(N^2\right)$ floating-point evaluations, as opposed to N in Eq. (4).

APPENDIX C: 42-QUBIT GHZ DEMONSTRATION

1. Experimental details

The experiments are run on the 65-qubit IBM Quantum Brooklyn system. The GHZ states are prepared starting with a Hadamard gate on qubit 11 and entangling additional qubits as shown in Fig. 8. After entangling the first six qubits, this pattern allows for increasing the GHZ state by four qubits per layer. The average assignment and controlled-NOT (CNOT) error rates across the qubits used are 2.15% and 1.01%, respectively.

2. Memory requirements for storing full 42-qubit A-matrix to $D = 3$

The inclusion of elements up to a Hamming distance of three requires

$$\binom{42}{0} + \binom{42}{1} + \binom{42}{2} + \binom{42}{3} = 12\,384 \qquad \text{(C1)}$$

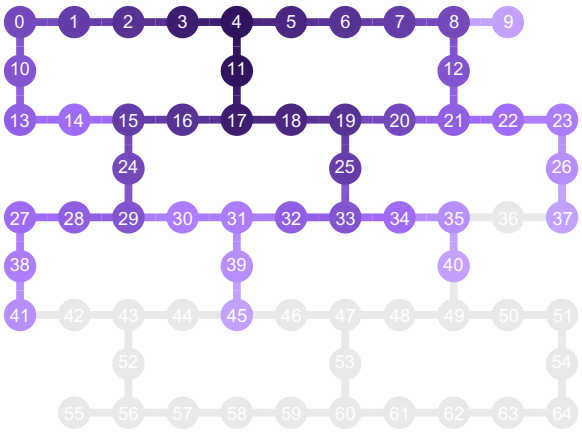

FIG. 8. The qubits used in generating GHZ states on the IBM Quantum Brooklyn system. Gates entangled earlier are color coded darker. The initial two-qubit GHZ state is between qubits 11 and 4, while the final 42-qubit GHZ state is formed by adding qubits 9, 37, 40, and 45 to the previous iteration.

elements in each of the 2^{42} columns. The storage of these values using single-precision floating-point numbers, 4 bytes per entry, requires 193.5 PiB of memory. In addition, we must also specify the row and column indices for these values. In CSC format, we need $2^{42} + 1$ elements, the difference of which specifies the number of nonzero elements in each of the columns. Lastly, we also need the row index for each nonzero matrix element. At 42 qubits, the size of these indices cannot be stored using 32-bit integers and we must use 64 bits per entry. The storage of these values requires an additional 387 PiB of memory. The total memory required is therefore 580.5 PiB and is approximately $120\times$ larger than the 4.85 PiB of memory on the Fugaku supercomputer [54].

[1] S. Bravyi, D. Gosset, and R. König, Quantum advantage with shallow circuits, Science **362**, 308 (2018).

[2] S. Bravyi, D. Gosset, R. König, and M. Tomamichel, Quantum advantage with noisy shallow circuits, Nat. Phys. **16**, 1040 (2020).

[3] D. Maslov, J.-S. Kim, S. Bravyi, T. J. Yoder, and S. Sheldon, Quantum advantage for computations with limited space, Nat. Phys. **17**, 894 (2021).

[4] K. Temme, S. Bravyi, and J. M. Gambetta, Error Mitigation for Short-Depth Quantum Circuits, Phys. Rev. Lett. **119**, 180509 (2017).

[5] S. Endo, S. C. Benjamin, and Y. Li, Practical Quantum Error Mitigation for Near-Future Applications, Phys. Rev. X **8**, 031027 (2018).

[6] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, Extending the computational reach of a noisy superconducting quantum processor, Nature **567**, 491 (2019).

[7] S. McArdle, X. Yuan, and S. C. Benjamin, Error-Mitigated Digital Quantum Simulation, Phys. Rev. Lett. **122**, 180501 (2019).

[8] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Denver, CO, 2020), p. 306.

[9] J. Sun, X. Yuan, T. Tsunoda, V. Vedral, S. C. Benjamin, and S. Endo, Mitigating Realistic Noise in Practical Noisy Intermediate-Scale Quantum Devices, Phys. Rev. Appl. **15**, 034026 (2021).

[10] Y. Kim, C. J. Wood, T. J. Yoder, S. T. Merkel, J. M. Gambetta, K. Temme, and A. Kandala, Scalable error mitigation for noisy quantum circuits produces competitive expectation values, arXiv:2108.09197 (2021).

[11] S. Bravyi, S. Sheldon, A. Kandala, D. C. McKay, and J. M. Gambetta, Mitigating measurement errors in multi-qubit experiments, Phys. Rev. A **103**, 042605 (2021).

[12] M. R. Geller, Rigorous measurement error correction, Quantum Sci. Technol. **5**, 03LT01 (2020).

[13] M. R. Geller and M. Sun, Efficient correction of multiqubit measurement errors, arXiv:2001.09980 (2020).

[14] K. E. Hamilton, T. Kharazi, T. Morris, A. J. McCaskey, R. S. Bennink, and C. Pooser, Raphael, Scalable quantum processor noise characterization, arXiv:2006.01805 (2020).

[15] E. van den Berg, Z. K. Minev, and K. Temme, Model-free readout-error mitigation for quantum expectation values, arXiv:2012.09738 (2020).

[16] F. B. Maciejewski, Z. Zimborás, and M. Oszmaniec, Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography, Quantum **4**, 257 (2020).

[17] B. Nachman, M. Urbanek, W. A. de Jong, and C. W. Bauer, Unfolding quantum computer readout noise, Npj Quantum Inf. **6**, 84 (2020).

[18] R. Hicks, C. W. Bauer, and B. Nachman, Readout rebalancing for near-term quantum computers, Phys. Rev. A **103**, 022407 (2021).

[19] E. Peters, A. C. Y. Li, and G. N. Perdue, Perturbative readout error mitigation for near term quantum computers, arXiv:2105.08161 (2021).

[20] K. Wang, Y.-A. Chen, and X. Wang, Measurement Error Mitigation via Truncated Neumann Series, arXiv:2103.13856 (2021).

[21] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature **549**, 242 (2017).

[22] M. Gong *et al.*, Genuine 12-Qubit Entanglement on a Superconducting Quantum Processor, Phys. Rev. Lett. **122**, 110501 (2019).

[23] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature **567**, 209 (2019).

[24] K. X. Wei, I. Lauer, S. Srinivasan, N. Sundaresan, D. T. McClure, D. Toyli, D. C. McKay, J. M. Gambetta, and S. Sheldon, Verifying multipartite entangled greenberger-horne-Zeilinger states via multiple quantum coherences, Phys. Rev. A **101**, 032343 (2020).

[25] G. J. Mooney, G. A. L. White, C. D. Hill, and L. C. L. Hollenberg, Generation and verification of 27-qubit greenberger-horne-Zeilinger states in a superconducting quantum computer, J. Phys. Commun. **5**, 095004 (2021).

[26] G. J. Mooney, G. A. L. White, C. D. Hill, and L. C. L. Hollenberg, Whole-device entanglement in a 65-qubit superconducting quantum computer, arXiv:2102.11521 (2021).

[27] K. J. Satzinger *et al.*, Realizing topologically ordered states on a quantum processor, arXiv:2104.01180 (2021).

[28] J. R. Glick, T. P. Gujarati, A. D. Córcoles, Y. Kim, A. Kandala, J. M. Gambetta, and K. Temme, Covariant quantum kernels for data with group structure, arXiv:2105.03406 (2021).

[29] QISKIT 0.26, https://qiskit.org.

[30] H. Pashayan, J. J. Wallman, and S. D. Bartlett, Estimating Outcome Probabilities of Quantum Circuits Using Quasiprobabilities, Phys. Rev. Lett. **115**, 070501 (2015).

[31] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, and A. Aspuru-Guzik, A variational eigenvalue solver on a photonic quantum processor., Nat. Commun. **5**, 4213 (2014).

[32] https://quantumcomputer.ac.cn (Accessed August 27, 2020).

[33] https://quantumai.google/hardware (Accessed June 01, 2021).

[34] https://quantum-computing.ibm.com/systems (Accessed August 23, 2021).

[35] https://aws.amazon.com/braket (Accessed June 01, 2021).

[36] https://www.quantum-inspire.com (Accessed August 27, 2020).

[37] P. D. Nation, J. R. Johansson, M. P. Blencowe, and A. J. Rimberg, Iterative solutions to the steady-state density matrix for optomechanical systems, Phys. Rev. E **91**, 013307 (2015).

[38] J. A. Smolin, J. M. Gambetta, and G. Smith, Efficient Method for Computing the Maximum-Likelihood Quantum State from Measurements with Additive Gaussian Noise, Phys. Rev. Lett. **108**, 070502 (2012).

[39] Keeping all elements is equivalent to setting D equal to the number of measured qubits, whereas $D = 0$ yields the identity matrix.

[40] Because least-squares is a numerical method, it is possible that elements outside of $\vec{p}_{\text{noisy}}$ are not strictly zero at the completion of this routine but. rather. zero up to floating-point precision. With this in mind, we consider all elements with absolute value $< 10^{-}15$ to be zero.

[41] Y. Saad, *Iterative Methods for Sparse Linear Systems* (Society for Industrial and Applied Mathematics, 2003), 2nd ed.

[42] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. and Stat. Comput. **7**, 856 (1986).

[43] H. A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. and Stat. Comput. **13**, 631 (1992).

[44] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van er Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* (SIAM, 1994), 2nd ed.

[45] W. W. Hager, Condition estimates, SIAM J. Sci. Star. Comput. **5**, 311 (1984).

[46] N. J. Higham, FORTRAN codes for estimating the one-norm of a real or complex matrix with applications to condition estimation, ACM Trans. Math. Softw. **14**, 381 (1988).

[47] https://github.com/qiskit-partners/mthree.

[48] C. R. Harris *et al.*, Array programming with NumPy, Nature **585**, 357 (2020).

[49] P. Virtanen, R. Gommers, and T. E. Oliphant *et al.*, SciPy 1.0: Fundamental algorithms for scientific computing in PYTHON, Nat. Methods **17**, 261 (2020).

[50] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, CYTHON: The best of both worlds, Comput. Sci. Eng. **13**, 31 (2011).

[51] http://www.openblas.net.

[52] "M3" stands for *m*atrix-free *m*easurement *m*itigation.

[53] We modify the QISKIT least-squares method to use $\vec{p}_{\text{noisy}}$ as the starting vector as opposed to a random vector. This gives a $3\times$ or more improvement in run time.

[54] https://www.r-ccs.riken.jp/en/fugaku/about/ (Accessed July 05, 2021).

[55] J. D. Hunter, MATPLOTLIB: A 2D graphics environment, Comput. Sci. Eng. **9**, 90 (2007).