Perturbative readout-error mitigation for near-term quantum computers

Evan Peters [1,2,3,*] Andy C. Y. Li [1] and Gabriel N. Perdue [1]
[1]*Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA*
[2]*Institute for Quantum Computing, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*
[3]*Department of Physics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*

(Received 9 May 2023; accepted 21 June 2023; published 30 June 2023)

Readout errors on near-term quantum computers can introduce significant error to the empirical probability distribution sampled from the output of a quantum circuit. These errors can be mitigated by classical postprocessing given the access of an experimental response matrix that describes the error associated with the measurement of each computational basis state. However, the resources required to characterize a complete response matrix and to compute the corrected probability distribution scale exponentially with the number of qubits, n. In this work, we modify standard matrix inversion techniques using perturbative approximations with significantly reduced complexity and bounded error when the likelihood of high-order bit-flip events is strongly suppressed. Given a characteristic error rate q, we discuss a method to recover the probability of the all-zeros bit string p_0 by sampling only a small subspace of the response matrix before inverting readout error, resulting in a relative speedup of poly$[2^n/\binom{n}{w}]$, which we motivate using a simplified error model for which the approximation incurs only $O(q^w)$ error for some integer w. We then provide a generalized technique to efficiently recover full output distributions with $O(q^w)$ error in the perturbative limit. These approximate techniques for readout-error correction may greatly accelerate near-term quantum computing applications.

DOI: 10.1103/PhysRevA.107.062426

I. INTRODUCTION

While quantum computing will potentially provide an exponential speedup in solving certain problems, noisy intermediate-scale quantum (NISQ) [1] devices are subject to high error rates that must be mitigated in order to extract useful information from the quantum processors. Readout error is unique among the standard sources of decoherence since it is well modeled by a *classical* stochastic process and is therefore entirely reversible by classical postprocessing. In the simplest approach, the effects of readout error can be reversed by inverting a *response matrix* R that relates premeasurement computational basis states to bit strings sampled by the measurement, provided that R is nonsingular and accurately characterizes the readout-error dynamics.

Previous works in readout-error mitigation typically involve some variation of inverting a Markovian process [2–8]. The goal of these postprocessing techniques is to recover the entire probability mass function $p(x)$ over bit strings $x \in \{0, 1\}^n$. However, these techniques are generally not scalable as they require experimental characterization of a response matrix followed by an (approximate) matrix inversion step, requiring device time and computing resources that grow exponentially in n. Bayesian iterative unfolding [9,10] avoids the latter hurdle by approximating the matrix inversion and readout rebalancing [11] improves on the accuracy of readout-error correction for recovering high-weight bit strings by biasing measurements based on some prior knowledge of the support of $p(x)$ on R^{2^n}. However, unless error mitigation is applied to recover a specific observable [12–14], both techniques still generally require an exponentially large device time to characterize the response matrix. Furthermore, with limited exceptions (e.g., Ref. [15]), few of these techniques have been specialized for the case where only a single bit-string probability is desired, which requires significantly fewer resources to mitigate readout error.

In this work, we present a perturbative technique for approximately correcting readout error on near-term quantum computers. Intuitively, the technique relies on an assumption that the likelihood of a readout-error event involving many simultaneous bit flips (for instance, observing the bit string 1111 after a computational basis measurement of the state $|0000\rangle$) is strongly suppressed in the number of simultaneous bit flips. This includes scenarios for which the bit flips are weakly correlated between different qubits and the individual bit-flip rates are small, which is often the case on existing devices [16].

We introduce our technique by considering the task of recovering the probability of the all-zeros bit string p_0 and show that this may be accomplished using only a small submatrix of R, and we provide numerical and theoretical evidence justifying this approximation. By tailoring the experimental determination of R towards recovering a specific bit string even in the presence of correlated readout errors, this approach offers a potential performance advantage over existing techniques designed to recover full distributions.

We then present the general technique to approximately recover the full empirical bit-string probability distribution by perturbatively expanding R in terms of a characteristic readout-error rate q. This technique represents a middle ground between full matrix inversion of R and sparsity-based

*e6peters@uwaterloo.ca

techniques. It is well suited for mitigating readout error when both the strength of the correlations between readout errors is known and the distribution $p(x)$ has nontrivial support on a large number of bit strings (e.g., superpolynomial in n), such that the probability of observing each bit string is influenced by the underlying probabilities for many other bit strings that are close in Hamming distance. Both variants of our technique allow for probabilities to be *approximately* corrected with the benefit of greatly reduced error correction overhead, and are therefore especially well suited for experiments in which readout error is not the limiting factor in the accuracy of the sampled probabilities.

II. INVERTING READOUT ERROR

Given an n-qubit state represented by its density matrix ρ, the error in a projective measurement $\{|i\rangle\langle i|\}$ for $i = 0, \ldots, 2^n - 1$ over the computational basis states can be modeled as a classical Markovian process [17], which is described by the equation

$$p' = \mathsf{R}\, p. \tag{1}$$

Here, R is a $2^n \times 2^n$ matrix with non-negative entries whose columns sum to one (i.e., a left stochastic matrix), $p = \mathrm{diag}(\rho)$ is a length-2^n normalized array of probabilities measured in the computational basis without measurement noise, and p' is the length-2^n array of observed (erroneous) bit-string probabilities. The response matrix R may be defined elementwise in terms of transition likelihoods,

$$\mathsf{R}_{ij} \equiv p(i|j) = p(i_1 \ldots i_n | j_1 \ldots j_n), \tag{2}$$

where $i, j \in \{0, 1\}^n$ are length-n bit strings, and the notation i_k is understood to refer to the kth bit of i. If we are provided with an invertible R, a basic prescription for correcting readout error is to compute

$$p = \mathsf{R}^{-1} p'. \tag{3}$$

In practice, R may be singular and a least-squares approximation to the linear equation (1) may be used.

Even when R is invertible, computing Eq. (3) in a general setting requires two distinct, resource-intensive steps: (i) measuring the complete response matrix of bit-string transition probabilities using a diagnostic experiment to determine R, with time complexity $O(2^n)$, and (ii) performing matrix inversion on R, which can be as costly as $O(2^{3n})$ [18]. Notably, recent works have explored more efficient sparsity-based techniques for mitigating readout error, for example, inverting the response matrix in the subspace spanned by nonzero components of p' [19,20].

A small infidelity in the readout-error mitigation can usually be tolerated as a trade-off for an improved complexity scaling in many cases, for example, when the readout error is less significant compared to other sources of error such as decoherence. We now introduce heuristic techniques for reducing the resource requirements of both of these steps while incurring some small, controllable error that may be inferred under some mild assumptions about the structure of R.

III. RECOVERING THE PROBABILITY OF AN ALL-ZEROS BIT STRING

We first study a special case of our technique for complexity reduction when one only desires to determine the probability of the all-zeros bit string $p_0 = \mathrm{Tr}(|0^n\rangle\langle 0^n|\rho)$. This scenario is relevant for near-term algorithms such as quantum kernel methods [7,21], dual-state purification [22], qubit assignment on hardware [23], and quantum circuit learning [24,25]. In this context, Eq. (3) can be cast in the form of a dot product,

$$p_0 = r \cdot p', \tag{4}$$

where the vector $r \in \mathbb{R}^{2^n}$ is defined elementwise as $r_i = (\mathsf{R}^{-1})_{0i}$. One would expect that a simplified readout-error mitigation can be performed to recover the observable p_0 with both tight error bounds and greater efficiency than for recovering the full distribution p, since only the subspace of R^{-1} that describes likely transitions into and out of 0^n is relevant. Assuming that the probability of a bit-string transition falls monotonically in the number of individual bits flipped, this subspace corresponds to the set of probabilities $(\mathsf{R})_{ij}$ for which i and j are low-weight strings.

Following this intuition, our proposed technique works by correcting p_0 using the inverse of a projection of R onto the subspace of low-weight basis vectors. The weight of a binary bit string $x = x_1 x_2 \ldots x_n \in \{0, 1\}^n$ is defined as

$$w(x) = \sum_{i=1}^{n} x_i, \tag{5}$$

which is the number of 1's appearing in x. We denote the set of all bit strings with weight less than w as

$$S_w = \{x : x \in \{0, 1\}^n,\ w(x) \leqslant w\} \subseteq \{0, 1\}^n. \tag{6}$$

We then define the weight projection operator $P_w : \mathbb{R}^d \to \mathbb{R}^{|S_w|}$ that projects vectors onto the subspace spanned by basis vectors whose binary index is in S_w. This is equivalent to the action

$$P_w \hat{e}_j = \begin{cases} \hat{e}_j & \text{if } j \in S_w \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where $\hat{e}_j$ is the jth unit vector. Then, a $d \times d$ matrix A may be projected onto the same subspace by the operation $P_w A P_w$. Defining $\mathsf{R}_T = P_w \mathsf{R} P_w$ and the first row of its inverse as $(r_T)_i = (\mathsf{R}_T^{-1})_{0i}$ by analogy with Eq. (4), our goal is to demonstrate that for some choice of $w < n$ and mild assumptions about the structure of R, we can compute

$$\tilde{p}_0 = r_T \cdot p'_T \tag{8}$$

as a close approximation to p_0, where $p'_T = P_w p'$. Equation (8) simply uses the first row r_T of the inverse of a truncated response matrix in place of r applied to a truncated observed probability vector p'_T. Applying Eq. (8) consumes a significantly smaller response matrix R_T which may be understood conceptually as the top-left submatrix of R with rows and columns re-sorted by index weight. R_T has dimensions

$t_w \times t_w$ given by the sum over binomial coefficients,

$$t_w = \sum_{j=0}^{w} \binom{n}{j}, \tag{9}$$

and so the readout-error correction for $\tilde{p}_0$ may be carried out by sampling R_T from a quantum processor with complexity $O(t_w)$ and then computing R_T^{-1} with complexity $O(t_w^3)$. In this work, we will neglect the effects of sampling error in both R and p', which introduces a constant overhead for readout-error correction as a function of the strength of noise on the device [13]. In the absence of statistical effects [such that Eq. (4) is satisfied], the error of our method is given as

$$|r_T \cdot p'_T - p_0| = |r_T \cdot p'_T - r \cdot p'|. \tag{10}$$

This approach therefore introduces error from two different sources: the first kind of error results from computing p after discarding bit-flip events involving more than w simultaneous relaxations and excitations, while the second kind of error is due to the truncation approximation R_T^{-1} - $P_w \mathsf{R}^{-1} P_w$. To motivate our technique, we proceed with study situations for which this difference vanishes and provide the resulting bounds on $|r_T \cdot p'_T - r \cdot p'|$ for progressively looser restrictions on the structure of R.

A. Exact bounds for a relaxation-only model

It is convenient to use the convention that vectors and matrices be sorted according to the weight of the binary representation of the index, with indices of equal weight sorted arbitrarily. For example, with $n = 3$, this has the effect of rearranging the vector of readout probabilities such that

$$p = (p_{000}, p_{001}, p_{010}, p_{100}, p_{110}, \cdots)^T. \tag{11}$$

We now consider an instructive toy model for readout error for which an analytical upper bound on the error $|p_0 - r \cdot p'|$ may be derived exactly. In this model, R is both overly simplified and trivially invertible, but the analysis will provide insight into approximations for situations where R has a more complex structure. The model for readout error that we study analytically is an extreme example of asymmetric readout error described by a response matrix of the form

$$\mathsf{R} = \bigotimes_{k=1}^{n} Q_k, \tag{12}$$

$$Q_k = \begin{pmatrix} 1 & q \\ 0 & (1-q) \end{pmatrix}, \tag{13}$$

for $0 \leqslant q < 0.5$. We can make very strong arguments about readout error arising from this model.

Proposition 1. Let R be defined as in Eq. (12). For any fixed projector P_w satisfying $P_w^2 = P_w$, define $\mathsf{R}_T = P_w \mathsf{R} P_w$. Then,

$$P_w(\mathsf{R}^{-1})P_w = (\mathsf{R}_T)^{-1}. \tag{14}$$

In other words, for this definition of R, the inverse of the projected response matrix R_T is equal to a projection of R^{-1}. This is a straightforward property of the kinds of upper triangular matrices that we are interested, but an intuitive proof is provided in Appendix A. The following theorem applies Proposition 1 to show that we can compute only a very small subspace of R and invert that subspace to apply readout-error correction to recover p_0 with error that is exponentially suppressed in our choice of truncation weight w.

Theorem 1. Let R be defined as in Eq. (12). Then the error introduced by correcting readout error using a truncated response matrix is bounded by

$$|r_T \cdot p'_T - r \cdot p'| \leqslant (2q)^{w+1}, \tag{15}$$

where r_T and r are defined elementwise as $(r_T)_i = (\mathsf{R}_T^{-1})_{0i}$ and $r_i = (\mathsf{R}^{-1})_{0i}$.

The proof is given in Appendix B. We remark that this is the tightest possible bound given the structure assumed of R that does not incorporate additional information about the readout probability distribution over the truncated subspace. Theorem 1 makes a simple but powerful observation that given the restricted noise model we have considered, one can apply readout error using the inverse of a projected matrix R_T such that the introduced truncation error is exponentially suppressed in w. This bound becomes quite weak in the limit that $q \to 0.5$ since the dimension of the truncated matrix itself grows combinatorially in w. Conversely, for $q \ll 1$ such that $q^2 \to 0$, this result guarantees that a matrix projected onto the $w = 1$ weight subspace with size *linear in n* can recover the probability of 0 with an accuracy almost as good as using the exponentially large R. Since this result is based only on the structure of R and not the value of elements contained therein, we can immediately lift some of the restrictions on constructing R.

Corollary 1. Let R have a tensor structure composed of distinct individual qubit response matrices of the following form:

$$\mathsf{R} = \bigotimes_{k=1}^{n} \begin{pmatrix} 1 & q_k \\ 0 & (1-q_k) \end{pmatrix}, \tag{16}$$

for $0 \leqslant q_k < 0.5$. Then,

$$|r_T \cdot p'_T - r \cdot p'| \leqslant (2q_{\max})^{w+1}, \tag{17}$$

where $q_{\max} = \max_k \{q_k\}$.

This is shown in Appendix B. Corollary 1 expands on the intuition of Theorem 1: If kth-order simultaneous bit-flip events are suppressed exponentially in k, then we need only sample a submatrix of R to perform good readout correction. We can further extend this line of reasoning to its practical limit in a somewhat less rigorous way. Suppose R is *any* response matrix that allows only for "relaxation" events, that is, R may be defined elementwise as

$$\mathsf{R}_{ij} = \begin{cases} p(i|j) & \text{for } w(i) < w(j) \text{ or } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

If we assume that the probability of a relaxation event is suppressed exponentially in the number of simultaneous bit flips, i.e., $p(i|j) \leqslant O(q^{w(j)-w(i)})$ for some characteristic rate q, then the above bounds still hold in the approximate sense,

$$|r_T \cdot p'_T - r \cdot p'| \lesssim O(2q^{w+1}). \tag{19}$$

This follows directly from Theorem 1; each entry with magnitude exactly $(1-q)^{w(x)} q^{w(y)}$ in the strictly upper triangular part of R can be replaced with an approximate term with order $O[(1-q)^{w(x)} q^{w(y)}]$. As this modification does not affect the

structure of R, similar nilpotency and series-expansion arguments that lead to Theorem 1 may be applied by substituting $q \to O(q)$. In this situation, R can no longer be decomposed and therefore R^{-1} can no longer be efficiently computed as $\bigotimes_k Q_k^{-1}$ using individual qubit response matrices $\{Q_k\}_{k=1}^n$. This extension also marks a departure from Corollary 1 by relaxing the assumption that R has a tensor structure, and therefore accommodates weakly correlated readout errors. Despite this structural change, the projected R_T^{-1} constructed from events with order no greater than w still serves as a useful surrogate for R^{-1} if only elements in the first row of R^{-1} are desired, and provides some justification for extending the reasoning of Eq. (8) to the more general case.

IV. PERTURBATIVE MITIGATION FOR RECOVERING THE FULL DISTRIBUTION

In the previous section, projecting R onto a subspace of low-weight indices was motivated by a model for readout error that penalizes the transition of high-weight bit strings into 0^n. This reasoning can be generalized to recovering the full bit-string distribution p more efficiently, assuming an error model that penalizes transitions between any two bit strings that differ by a large hamming weight. This is a practical model even when there is correlated readout error between different qubits, provided the correlation strength is not comparable to the characteristic rate. This model is further motivated by the observation that correlations in readout error are likely to be strong only among qubits that are physically adjacent on a device, for example, nearest neighbors on a two-dimensional grid of superconducting qubits [26].

To proceed, we assume there is a characteristic rate q that describes the probability of any given single bit-flip event. For each $j = 1, \ldots, n$, we define a sparse $2^n \times 2^n$ off-diagonal matrix R_j whose entries are of magnitude $O(1)$. Then, without loss of generality, we define R with respect to a series structure such that

$$\mathsf{R} = \mathsf{R}_0 + \sum_{j=1}^{2^n} q^j \mathsf{R}_j, \tag{20}$$

where each R_j contains a subset of elements of R according to some pairwise comparison function s,

$$(\mathsf{R}_j)_{nm} = \begin{cases} (\mathsf{R})_{nm} & \text{if } s(n,m) = j \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

In this work, we will focus on the specific choice

$$s(n,m) = w(n \oplus m), \tag{22}$$

where w is the weight function of Eq. (5) and $\oplus$ denotes the bitwise modulo-2 sum. Then, R_0 consists of the diagonal matrix elements of R of all orders of q and R_j consists of the off-diagonal terms of the order q^j describing all likelihoods involving bit flips whose weights differ by j. Note that under this definition, R_0 will describe all bit-flip events of even order in which the observed bit string is identical to the prior bit string, and similarly for $j = 1, \ldots, n$, so that R_j does not necessarily characterize events involving exactly j bit flips. Even if R is not well modeled by this choice of decomposition, such as in cases involving strongly correlated readout errors, it still may be possible to define R_j to include all events with probability on order q^j. This will require significant prior knowledge about the scale of readout errors on the device and is out of the scope for this work.

Algorithm 1 Perturbative mitigation for the full distribution

Input : $p', \{\mathcal{R}_j\}_{j=0,1,\cdots,w}$
$\mathsf{S} \leftarrow -\sum_{j=1}^{w} \mathsf{R}_0^{-1}\mathcal{R}_j$;
$v \leftarrow \mathsf{R}_0^{-1}p'$;
$\tilde{p} \leftarrow v$;
for $k \leftarrow 1$ **to** w **do**
  $v \leftarrow \mathsf{S}v$;
  $\tilde{p} \leftarrow \tilde{p} + v$;
end

The form of Eq. (20) suggests that the R_j corresponding to small j will dominate the effects of dynamics error. Applying this intuition, we expand the inverse of Eq. (20) as a series,

$$\mathsf{R}^{-1} = \mathsf{R}_0^{-1} + \sum_{k=1}^{\infty} \left(-\sum_{j=1}^{2^n} q^j \mathsf{R}_0^{-1}\mathsf{R}_j \right)^k \mathsf{R}_0^{-1}. \tag{23}$$

Truncating both series to the order of q^w results in

$$\mathsf{R}^{-1} = \left[1 + \sum_{k=1}^{w} \left(-\sum_{j=1}^{w} q^j \mathsf{R}_0^{-1}\mathsf{R}_j \right)^k \right] \mathsf{R}_0^{-1} + O(q^{w+1}). \tag{24}$$

Note that this expression contains some terms of order higher than q^w, but the error introduced by the truncation remains bounded by $O(q^{w+1})$. Applying this expression to p', we arrive at an approximate probability distribution $\tilde{p}$ given by

$$\tilde{p} = \left[1 + \sum_{k=1}^{w} \left(-\sum_{j=1}^{w} \mathsf{R}_0^{-1}\mathcal{R}_j \right)^k \right] \mathsf{R}_0^{-1} p', \tag{25}$$

where we have introduced $\mathcal{R}_j \equiv q^j \mathsf{R}_j$ as the matrix of elements describing order-j transitions sampled directly from an experimental response matrix (for which knowledge of the rate q is not strictly necessary). This result can be viewed as a generalization of the specialized task described in Eq. (8), which we discuss in Appendix C.

The implementation of the perturbative readout-error mitigation to recover an empirical distribution over bit strings sampled from a quantum computer subject to measurement error is described by the following pseudocode.

If R^{-1} exists, then the Neumann series introduced in Eq. (23) converges only if $\| \sum_{j=1}^{w} \mathsf{R}_0^{-1}\mathcal{R}_j \| < 1$, which determines whether Algorithm 1 can be applied in its given form. If this condition is met, then the error $\epsilon = \|p - \tilde{p}\|_2$ introduced by Algorithm 1 is concentrated in the $(w+1)$th-order terms, resulting in an approximate error given by $\epsilon \lesssim 2q^{w+1} + O(q^{w+2})$, where we have applied the slightly stronger assumption that $\|\mathsf{R}_0^{-1}\mathsf{R}_j\| \leqslant 1$. To reach an accuracy with an error ϵ,

we need to implement the algorithm with an order of at least

$$w \geqslant \left\lceil \frac{\ln \epsilon^{-1} + \ln 2}{\ln q^{-1}} \right\rceil - 1. \tag{26}$$

The complexity of this technique is dominated by matrix products involving S in Algorithm 1. If S is sparse, with s being roughly the fraction of elements that are nonzero, the algorithm requires w matrix product operations resulting in approximate time complexity given by

$$O(s_w w M^3). \tag{27}$$

We now compute the sparsity of $\mathcal{R}_w$ to determine the relative speedup of this technique over standard matrix inversion. The nonzero elements of $\mathcal{R}_j$ occur at all index pairs (x, y) satisfying $x \oplus y = j$, where $x, y \in B$ and $B = \{0, 1\}^n$. The number of pairs (x, y) that satisfy this condition is equivalent to the number of strings x satisfying $x = z \oplus y$, where $z \in B_j$ and $B_j = \{s : s \in B, w(s) = j\}$ is the set of all weight-j bit strings. This number is given by $|B| \times |B_j|$, and so the sparsity factor s_w describing the number of nonzero terms in $\mathcal{R}_w$ may be computed directly as $2^{-2n} \times |B| \times |B_j|$ or

$$s_w = 2^{-n} \binom{n}{w}. \tag{28}$$

This sparsity is computed using the union of Hamming balls around each weight-k bit string, i.e., our construction implicitly assumes readout errors that are only weakly correlated such that transitions between bit strings with large Hamming distance are suppressed.

In the context of Eq. (27), s_w has the effect of replacing one term proportional to 2^n with a term proportional to $w\binom{n}{w}$. The core speedup therefore comes from generating $\tilde{p}$ in Algorithm 1 by a series of sparse matrix-vector products using matrices with, at most, s_w nonzero terms. If convergence of the Neumann series of Eq. (23) is not guaranteed, a perturbative technique may still be useful. In this case, an experimentalist would still measure the set $\{\mathcal{R}_j\}$ to a desired truncation point w, and then directly invert the resulting approximation to R to recover

$$\tilde{p} = \left(\sum_{j=1}^{w} \mathcal{R}_j \right)^{-1} p'. \tag{29}$$

This may result in significant speedup compared to sampling the full R, but incurs additional computational cost to compute a standard matrix inverse, and we explore this tradeoff in Appendix D. This approach might therefore be compatible with other techniques that avoid directly computing a matrix inverse (for instance, Bayesian iterative unfolding [9–11]) or with techniques for efficient inversion of sparse, banded matrices [27,28].

This approach allows us to safely ignore the small response matrix elements corresponding to higher-order terms. In experiments, each column of R may be estimated by preparing the state corresponding to that column and then computing the output bit-string distribution for a computational basis measurement, resulting in an estimate for the matrix elements of the corresponding column of R. By discarding the higher-order terms, this distribution can be reliably determined with a number of shots such as $n_{\text{meas}} \sim 1/q^{2w}$. If a sufficient truncation order w is known either from previous experiments or knowledge of the hardware design (e.g., Ref. [16]), our technique saves resources by allowing the experimentalist to omit measurement of higher-order terms. Moreover, the resource requirement may be reduced further if only a few bit strings of the output distribution are required. This includes, for example, the all-zeros bit-string case discussed in Sec. III and cases where only a specific qubit excitation sector is of interest due to symmetry. In these cases, we need only estimate columns of R by computing elements corresponding to the desired bit-string population (see Appendix C). This allows us to compute R using fewer experiments on quantum hardware.

V. NUMERICAL EXPERIMENTS

We implemented the technique introduced in the preceding sections for a variety of prior distributions and readout-error strengths. To avoid complications due to statistical uncertainty, we restrict ourselves to using p and R that were simulated to floating point precision without introducing any sampling error. We generated each R randomly in the following manner: We constructed a tensor product of the form

$$\mathsf{R} = \bigotimes_{k=1}^{n} \begin{pmatrix} 1 - \eta_k & \epsilon_k \\ \eta_k & 1 - \epsilon_k \end{pmatrix} \tag{30}$$

with $\epsilon_k, \eta_k \sim \text{Uniform}(0, q)$, and then we randomly permuted each weight-k subspace of R for $k = 1, \ldots, n-1$. The resulting matrix is not separable and therefore cannot be trivially inverted by inverting each term in the Kronecker product of Eq. (30). We applied the resulting linear map to a prior distribution p to generate p'. Figure 1 demonstrates the performance of Eq. (8) to correct for p_0 for different prior distributions. The error in the method is suppressed exponentially in the truncation order w, which is consistent with behavior that was analytically derived for a more restricted error model in Sec. III. A similar exponential suppression of the error can be observed in experiments using response matrices measured on IBM QPUs (Appendix E 1). In Appendix E 2, we provide a preliminary comparison of our method to the M3 technique of Ref. [19].

For Algorithm 1, we are interested in assessing the performance of the readout correction for recovering the full distribution p' compared to p. To compare the two distributions, we compute the trace distance (or L1 norm),

$$d(p, q) = \sum_{j \in \{0,1\}^n} |p_j - q_j|. \tag{31}$$

This has a useful interpretation in terms of computing expectation values of Hermitian operators. Let O be an operator with 2^n real entries on the diagonal bounded by $O_j \in [-1, 1]$. The expected value $\langle O \rangle = \text{Tr}(O\rho)$ corresponding to the corrected distribution $\tilde{p}$ is estimated using the quantity

$$E_O = \sum_{j \in \{0,1\}^n} O_j \tilde{p}_j. \tag{32}$$

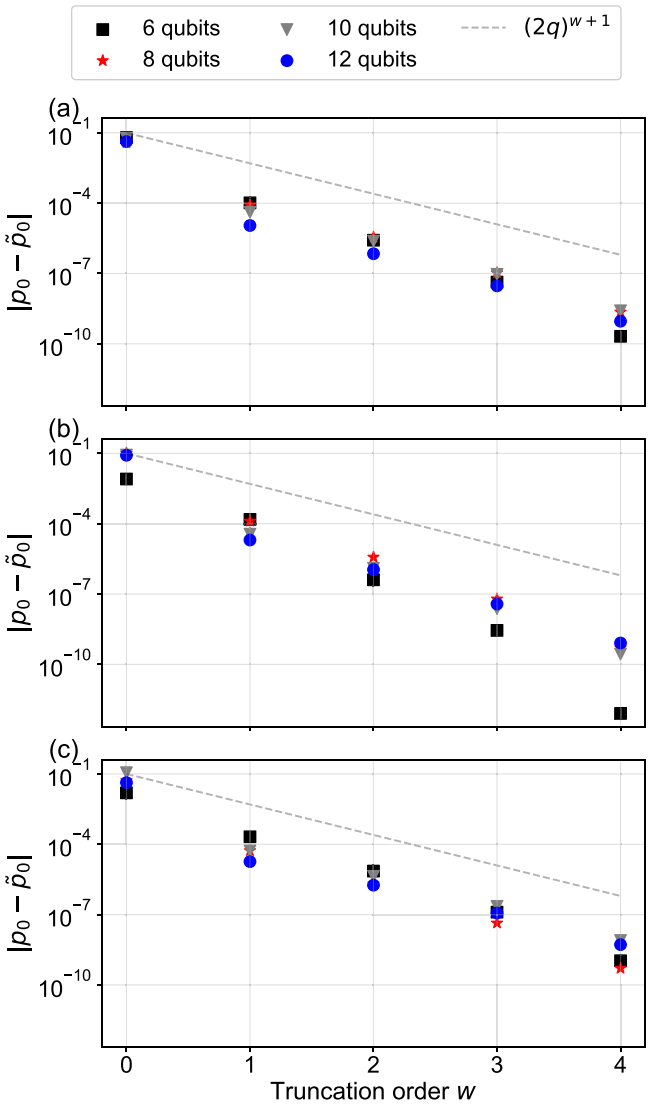

FIG. 1. Performance of all-zeros readout-error mitigation given by Eq. (8) compared for different prior distributions. (a) The Gaussian prior is centered at 0 with n-bit overflow for all bit strings with value less than 2^{n-1}, i.e., $p_j \propto \exp[(x_j - 0.5)^2/\sigma^2]$, where $x_j \equiv 2^{-n}[(j + 2^{n-1}) \bmod 2^n]$ and $\sigma = 0.25$. This distribution is adversarial to recovering p_0 as it has significant support on the high-weight subspace. (b) The truncated Gaussian is given by the same distribution without overflow ($x_j = j \times 2^{-n}, \sigma = 0.25$) and renormalized, and (c) the uniform distribution is $p_j = 2^{-n}$. In all plots, $w = 0$ is defined to correspond to the uncorrected probability p'_0. The dashed line indicates the bound of Eq. (19) derived for a relaxation-only model, which we observed was not violated even for the more general error model of Eq. (30).

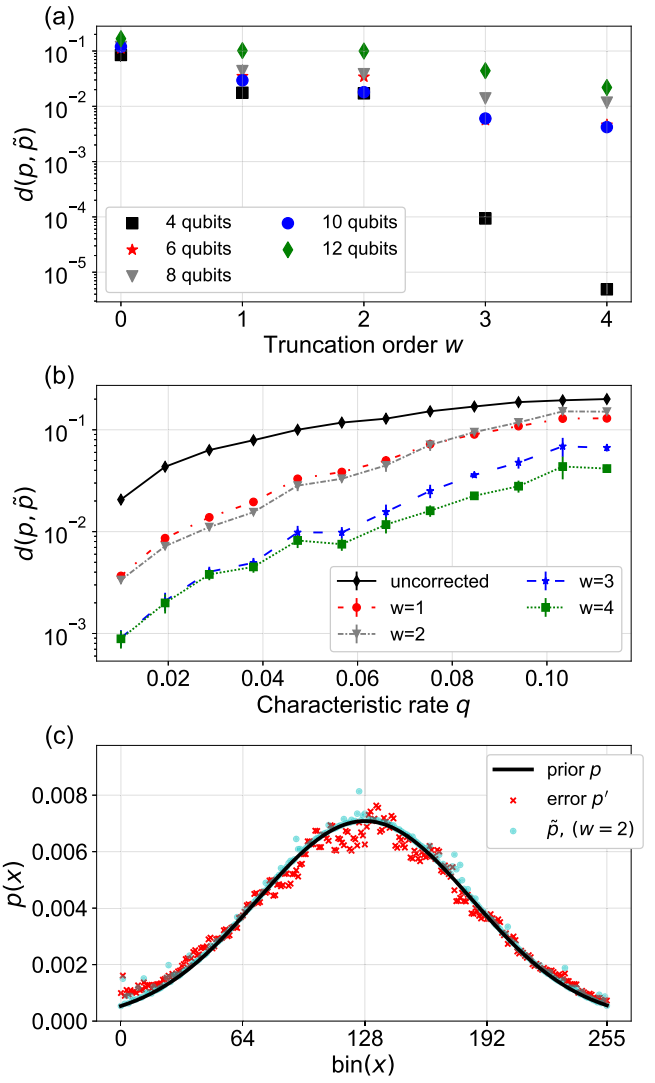

FIG. 2. (a) Performance of Algorithm 1 as a function of truncation order w, for fixed $q = 0.05$. The elements of the initial distribution p were each drawn from Uniform$(0, 1)$ and then normalized. We define $w = 0$ to represent the uncorrected case corresponding to $d(p, p')$. (b) The performance of the algorithm ($n = 8$) diminishes with increasing q, which corresponds to the series approximation for R^{-1} diverging. We discuss the resulting limitations and workarounds for this behavior in Appendix D. (c) Visualization of applying correction to a Gaussian distribution for $n = 8, w = 2$ with a characteristic rate $q = 0.6$.

We can then show that $d(p, \tilde{p})$ bounds the error incurred in E_O,

$$|\mathrm{Tr}(O\rho) - E_O| \leqslant d(p, \tilde{p}), \tag{33}$$

and so d serves as a natural comparison between output distributions that will be postprocessed to compute observables. Figure 2 shows the performance of Algorithm 1 for varying truncation orders, numbers n of qubits, and increasing noise rates q. For modest q, the effects of readout error can be strongly suppressed using only a fraction of the resources required for inverting all of R. As n or q increases, the performance of the algorithm rapidly drops off as the series for truncated inverse requires significantly more terms to converge. Figure 2 also provides a visual example of Algorithm 1 applied using a second-order truncation, which will generally consume $O(n^2)$ resources for sampling R_2 to recover $\tilde{p}$.

VI. CONCLUSION

We have proposed a technique for approximately correcting readout error in quantum computers requiring significantly less overhead than traditional matrix inversion

techniques, while still capturing enough of the readout-error behavior to correct distributions with support on a large number of bit strings that might present a challenge for sparsity-based techniques. Such approximations are beneficial when the error in the correction scheme becomes negligible compared to other sources of device noise, and so this technique may be useful for running quantum algorithms on near-term devices. We have justified the technique in the perturbative regime and also provided numerical evidence suggesting that this technique can be useful if either the characteristic error rate q or the number of qubits, n, remains small (e.g., nq does not grow too large; see Appendix D). Future work may further generalize the bounds we have derived and elucidate the nonperturbative regimes for which the errors in our methods remain well bounded.

Note added. Recently, Ref. [29] appeared. The authors of that paper also employed series approximations for computing R^{-1} (as discussed in Sec. IV), but the implementation is otherwise unrelated to the technique described here.

ACKNOWLEDGMENTS

We thank Achim Kempf for reviewing this manuscript. E.P. is partially supported through a Kempf's Google Faculty Award. E.P. and G.P. are partially supported by the U.S. Department of Energy/HEP QuantISED program grant HEP Machine Learning and Optimization Go Quantum, Identification No. 0000240323. A.C.Y.L. is supported by the U.S. DOE HEP QuantISED Grant No. KA2401032. This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. This research used resources of the Oak Ridge National Laboratory, which is a U.S. DOE Office of Science User Facility supported under Contract No. DE-AC05-00OR22725.

APPENDIX A: PROOF OF PROPOSITION 1

The construction for R given in Eq. (12) is a tensor product of identical single-qubit response matrices, each of which prescribes a fixed probability for a relaxation event $p(0|1) = q$ and disallows excitation [$p(1|0) = 0$]. The outline of the proof is that disallowing excitations results in an R with a block structure such that projection operations commute with the matrix product for strictly upper triangular submatrices of R acting over indices with weight less than w. The tensor structure then allows direct computation of r, and therefore also of r_T.

Q is upper triangular and therefore R is also upper triangular. Given the tensor structure of R, we have that for indices $i, j \in \{0, 1\}^n$ in the upper triangular set, the elements of R are given elementwise by

$$\mathsf{R}_{ij} = \begin{cases} q^{w(j)-w(i)}(1-q)^{w(i)} & \text{for } w(i) < w(j) \\ (1-q)^{w(i)} & \text{for } i = j \\ 0 & \text{else,} \end{cases} \tag{A1}$$

where the term $q^{w(j)-w(i)}(1-q)^{w(i)}$ represents the probability of $|w(j) - w(i)|$ simultaneous relaxations times the probability of the remaining $w(j) - |w(j) - w(i)| = w(i)$ bits *not* relaxing. We split R into a diagonal component R_0 and a strictly upper triangular component

$$\mathsf{R}_u = \sum_{k=0}^{n} \sum_{\ell=k+1}^{n} |k\rangle\langle\ell| \otimes B_{k\ell}, \tag{A2}$$

which represents a block matrix partition of R into $\{B_{k\ell}\}$, each of which contains all elements $(\mathsf{R})_{ij}$ with $w(i) = k$ and $w(j) = \ell$ and has dimensions

$$\dim(B_{k\ell}) = \binom{n}{k} \times \binom{n}{\ell}. \tag{A3}$$

R_u is an $(n+1) \times (n+1)$ block matrix and is strictly upper triangular with respect to this block structure. For any strictly upper triangular $m \times m$ matrix A, we have that $A^m = 0$ [30] and, by similar reasoning, $\mathsf{R}_u^{n+1} = 0$. Therefore, the Neumann series expansion for R^{-1} converges in $n+1$ terms and we obtain

$$\mathsf{R}^{-1} = \left[\sum_{k=0}^{n} \left(-\mathsf{R}_0^{-1}\mathsf{R}_u\right)^k\right]\mathsf{R}_0^{-1}. \tag{A4}$$

In computing $(\mathsf{R}_u)^k$, every column space over basis vectors of weight w depends only on contributions of column spaces over basis vectors of weight less than w (i.e., the columns to the left of the weight-w subspace). Therefore, we further partition R_u into column spaces $\{\sum_{k+\ell \geqslant w} B_{k\ell}\}_{w=1}^{n}$ with basis vectors less than or equal to w and ignore the complementary column space for computing the truncated part of R^{-1}. That is, defining the projector $\pi \equiv P_{0,w}$ and letting $\pi_\perp = I - \pi$ be the projector onto the complementary subspace of π, we simplify our representation of R_u as

$$\mathsf{R}_u \equiv \begin{pmatrix} \pi \\ \pi_\perp \end{pmatrix} \mathsf{R}_u (\pi \quad \pi_\perp) \to \begin{pmatrix} \pi\mathsf{R}_u\pi & * \\ 0 & * \end{pmatrix}. \tag{A5}$$

The sum in Eq. (A4) may then be computed ignoring the column space of weights greater than w,

$$\left(-\mathsf{R}_0^{-1}\mathsf{R}_u\right)^k \mathsf{R}_0^{-1} \tag{A6}$$

$$= \begin{pmatrix} \left(-\pi\mathsf{R}_0^{-1}\mathsf{R}_u\pi\right)^k & * \\ 0 & * \end{pmatrix} \begin{pmatrix} \pi\mathsf{R}_0^{-1}\pi & * \\ 0 & * \end{pmatrix} \tag{A7}$$

$$= \begin{pmatrix} \left(-\pi\mathsf{R}_0^{-1}\mathsf{R}_u\pi\right)^k \pi\mathsf{R}_0^{-1}\pi & * \\ 0 & * \end{pmatrix} \tag{A8}$$

since $\pi^2 = \pi$ by definition. Therefore,

$$\pi(R^{-1})\pi = \pi\left[\sum_{k=0}^{n} \left(-\mathsf{R}_0^{-1}\mathsf{R}_u\right)^k \mathsf{R}_0^{-1}\right]\pi \tag{A9}$$

$$= \sum_{k=0}^{w} \left(-\pi\mathsf{R}_0^{-1}\mathsf{R}_u\pi\right)^k \left(\pi\mathsf{R}_0^{-1}\pi\right) \tag{A10}$$

$$= \sum_{k=0}^{w} \left(-\pi\mathsf{R}_0^{-1}\pi\right)\left(\pi\mathsf{R}_u\pi\right)^k \left(\pi\mathsf{R}_0^{-1}\pi\right) \tag{A11}$$

$$\equiv (\pi R \pi)^{-1}, \tag{A12}$$

where the series now terminates at w due to the nilpotency of $\pi \mathsf{R}_u \pi$ in Eq. (A7). Line (A11) is simply the series expansion for the inverse of $\pi R \pi = \pi \mathsf{R}_0 \pi + \pi \mathsf{R}_u \pi$, which concludes the proof.

APPENDIX B: PROOF OF THEOREM 1 AND COROLLARY 1

The size of the projected subspace that includes all strings of weight less than or equal to w is the binomial sum

$$t(w) = \sum_{k=0}^{w} \binom{n}{k}. \tag{B1}$$

We can compute the error in the projected readout-error correction method directly,

$$\begin{aligned}
|r_T \cdot p'_T - r \cdot p'| &= |[(\pi R \pi)^{-1} p'_T]_0 - (\mathsf{R}^{-1} p')_0| \\
&= |[\pi (\mathsf{R}^{-1}) \pi p'_T]_0 - (\mathsf{R}^{-1} p')_0| \\
&= \left| \sum_{j=0}^{t(w)} (\mathsf{R}^{-1})_{0j} p'_j - \sum_{j=0}^{2^n-1} (\mathsf{R}^{-1})_{0j} p'_j \right| \\
&= \left| \sum_{j=t(w)+1}^{2^n-1} (\mathsf{R}^{-1})_{0j} p'_j \right| \\
&= \left| \sum_{k=w+1}^{n} \left(\frac{q}{1-q} \right)^k \sum_{\ell \in B(w)} p'_\ell \right| && \text{(B2)} \\
&\leqslant \left(\frac{q}{1-q} \right)^{w+1} && \text{(B3)} \\
&\leqslant (2q)^{w+1}, && \text{(B4)}
\end{aligned}$$

where $B(w)$ denotes the set of bit strings of weight w, and $\pi \equiv P_{0,w}$ (as defined in Appendix A). In line (B2), the additional factor of $(1-q)^k$ is found by explicitly computing the first row of R^{-1}: For a binary string $j = j_1 j_2 \ldots j_n$, the jth entry of r is

$$\begin{aligned}
|r_j| &= \left| (Q^{-1})_{0j_1} (Q^{-1})_{0j_2} \ldots (Q^{-1})_{0j_n} \right| && \text{(B5)} \\
&= \left(\frac{q}{(1-q)} \right)^{w(j)}. && \text{(B6)}
\end{aligned}$$

Then, with the requirement that $q < 0.5$, we have $(1-q)^{-k} < 2^{-k}$, which completes the proof. To prove Corollary 1, we again explicitly compute r_j, taking advantage of the tensor structure of R^{-1},

$$\begin{aligned}
|r_j| &= \left| \left(Q_1^{-1}\right)_{0j_1} \left(Q_2^{-1}\right)_{0j_2} \ldots \left(Q_k^{-1}\right)_{0j_n} \right| && \text{(B7)} \\
&= \prod_{j_k} \left(\frac{q_k}{(1-q_k)} \right)^{j_k} && \text{(B8)} \\
&\leqslant \left(\max_k \frac{q_k}{(1-q_k)} \right)^{w(j)}, && \text{(B9)}
\end{aligned}$$

which upper bounds the magnitude of any element in the subspace of R^{-1} excluded by the truncation of all bit strings j with $w(j) > w$. The proof of Corollary 1 proceeds identically as with Theorem 1. Note that the bound given is quite loose and so the exact expression given in Eq. (B8) may be freely substituted if $q_{\max}$ is expected to be significantly larger than a typical q_k.

APPENDIX C: REDUCTION OF ALGORITHM 1 TO SINGLE BIT STRINGS

Algorithm 1 can be further simplified if we are only interested in a specific element $\ell \in \{0,1\}^n$ from the distribution. To illustrate the reduction, we construct a set $S_{\ell,w}$ consisting of all $M_{\ell,w} \equiv |S_{\ell,w}|$ basis states with, at most, a distance w away from ℓ,

$$S_{\ell,w} = \{ m \mid s(m,\ell) \leqslant w \}. \tag{C1}$$

We can define a projection operator $\mathsf{P}_{\ell,w}$ given by

$$\mathsf{P}_{\ell,w} \hat{e}_n = \begin{cases} \hat{e}_n & \text{if } n \in S_{\ell,w} \\ 0 & \text{otherwise.} \end{cases} \tag{C2}$$

It follows from Eq. (24) that the prior probability p_ℓ for measuring a computational basis state $|\ell\rangle$ is given by

$$\begin{aligned}
p_\ell = \sum_m \left\{ \delta_{m,\ell} + \sum_{k=1}^{w} \left[\left(-\sum_{j=1}^{w} q^j \mathsf{R}_0^{-1} \mathsf{R}_j \right)^k \right]_{\ell,m} \right\} \\
\times (\mathsf{R}_0)^{-1}_{m,m} p'_m + O(q^{w+1}). && \text{(C3)}
\end{aligned}$$

It is straightforward to show that any matrix elements $[(-\sum_{j=1}^{w} q^j \mathsf{R}_0^{-1} \mathsf{R}_j)^k]_{\ell,m}$ with $s(m,\ell) > w$ are of the order beyond q^w. We can thus use the projection operator $\mathsf{P}_{\ell,w}$ to write

$$\begin{aligned}
&\left[\left(-\sum_{j=1}^{w} q^j \mathsf{R}_0^{-1} \mathsf{R}_j \right)^k \right]_{\ell,m} \\
&= \left[\left(-\sum_{j=1}^{w} q^j \mathsf{P}_{\ell,w} \mathsf{R}_0^{-1} \mathsf{P}_{\ell,w} \mathsf{R}_j \mathsf{P}_{\ell,w} \right)^k \right]_{\ell,m} + O(q^{w+1}).
\end{aligned}$$

Defining the truncated operators to be $\mathsf{R}_j^{(\ell,w)} = \mathsf{P}_{\ell,w} \mathsf{R}_j \mathsf{P}_{\ell,w}$, we get the approximated probability to be

$$\begin{aligned}
\tilde{p}_\ell = \sum_m \left\{ \delta_{m,\ell} + \sum_{k=1}^{w} \left[-\sum_{j=1}^{w} q^j \left(\mathsf{R}_0^{(\ell,w)}\right)^{-1} \mathsf{R}_j^{(\ell,w)} \right]^k \right\}_{\ell,m} \\
\times \left(\mathsf{R}_0^{(\ell,w)}\right)^{-1}_{m,m} p'_m. && \text{(C4)}
\end{aligned}$$

The algorithm to determine $\tilde{p}_\ell$ is similar to that for the full distribution, but with R_j being truncated to dimensions $M_{\ell,w} \times M_{\ell,w}$. The time complexity is thus given by $O(M_{\ell,w}^3)$, which is consistent with the all-zeros bit-string case discussed in Sec. III.

Similarly to the case in Sec. IV, the strategy of decomposing R into a set of sparse components $\{\mathcal{R}_j\}$ can be combined

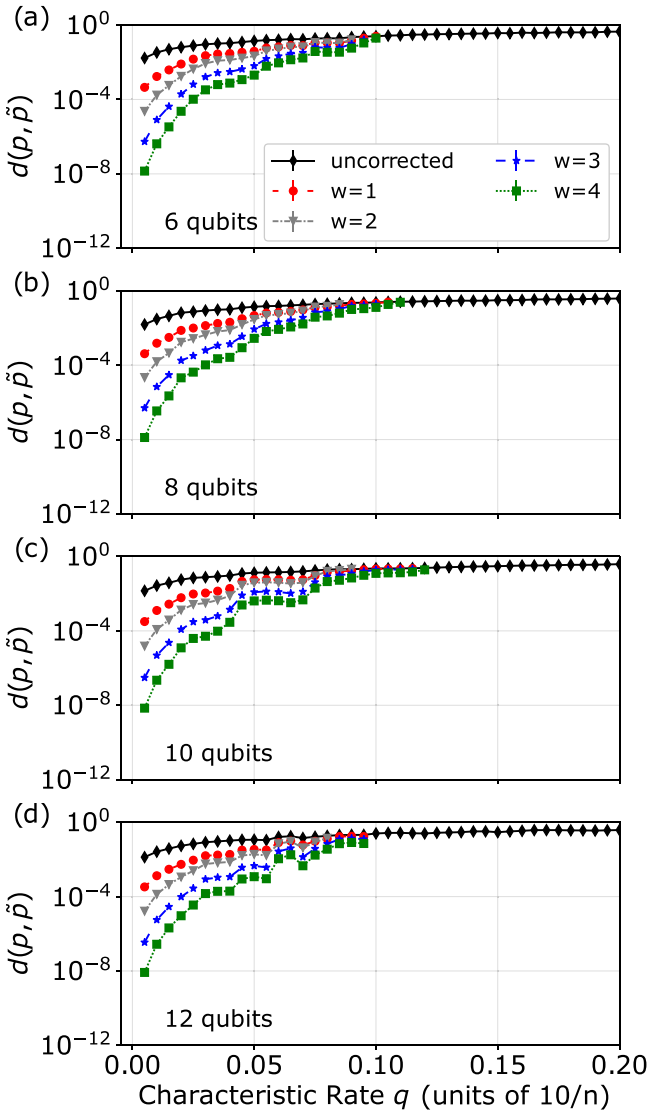

FIG. 3. Failure points for variable number of qubits occur around the same threshold $q_{\max}$ in units of $10\frac{q}{n}$. We empirically observe that this failure threshold scales inversely with n, signifying that our technique loses effectiveness when the characteristic error rate q cannot be suppressed as additional qubits are added to the system.

with alternatives to standard matrix inversion or series approximations to matrix inversion for recovering a specific bit string ℓ. For instance, the technique of [31] for the ϵ-close approximation for specific elements of the solution to $Ax = b$ could be applied to recover p_ℓ with exponential speedup over recovering the entire distribution p provided additional conditions on R. However, the procedure to correct the readout probability for a specific bit string $\ell \neq 0^n$ will generally require more resources than recovering the all-zeros bit string. The subspace $S_{\ell,w}$ can be significantly larger than the subspace $S_{0,w}$, given that the majority of elements in $\{0,1\}^n$ have weight close to $\frac{n}{2}$. For example, to implement the algorithm of Sec. III to recover p_ℓ, R must be projected onto a subspace $\{x : w(\ell) - w_{\min} \leqslant w(x) \leqslant w(\ell) + w_{\max}\}$ consisting of strings with weight in $[w_{\min}, w_{\max}]$. This differs from the $\ell = 0^n$ case since the population p'_0 cannot be increased due to the excitation of other bit strings $x \neq 0$.

From the perspective of readout-error mitigation, computing p_0 [or $p(1^n)$, if necessary] is ideal, as it is the string with the fewest neighbors separated by a low-weight error event. Consequently, if one desires to compute the probability of a fixed bit string $\ell = \ell_1\ell_2\ldots\ell_n$ at the output of a quantum circuit U, from the perspective of mitigating readout error, it is preferable to perform readout rebalancing [11] by appending a single layer of gates to construct $U' = (\sigma_x^{\ell_1} \otimes \sigma_x^{\ell_2} \otimes \cdots \otimes \sigma_x^{\ell_n})U$, where σ_x is the Pauli-X gate. Then the corrected probability $\tilde{p}_0$ sampled from the output of U' is a maximally efficient approximation to p_ℓ sampled from U.

APPENDIX D: MODIFIED PERTURBATIVE TECHNIQUE

As mentioned in the main text, our technique relies on the assumption that the Neumann series for R^{-1} converges, namely,

$$\left\| \sum_{j=1}^{w} \mathsf{R}_0^{-1}\mathcal{R}_j \right\| < 1. \tag{D1}$$

In general, increasing the number of qubits, n, will uniformly increase the left-hand side of Eq. (D1), as the magnitude of the elements of each matrix R_j is constant with respect to n but the size of the matrix grows exponentially in n. Unless the characteristic error rate q is reduced simultaneously as n is increased (for example, by bounding the product nq), the perturbative approximation of Algorithm 1 will no longer hold. Figure 3 shows the effect of increasing q on the performance of Algorithm 1. The failure point of each experiment occurs when the error in $\tilde{p}$ is comparable to the error in p', which we observed to typically coincide with reaching a characteristic rate $q_{\max}$ for which Eq. (D1) was no longer satisfied.

Our technique may be modified slightly so that it still performs well even when the requirement of Eq. (D1) is no longer satisfied, since $\sum_{j=1}^{w} \mathcal{R}_j$ may still be invertible even when the Neumann series for its inverse does not converge. Figure 4 shows the performance of this modified algorithm, and we highlight the fact that this performance improves steadily with increasing w. This improvement comes at the cost of a larger classical computation overhead, but this scenario may still be preferable over completely characterizing R with an exponentially large diagnostic experiment.

APPENDIX E: ADDITIONAL NUMERICS

1. Response matrices measured on IBM QPUs

Our numerical experiments in the main text used response matrices generated with a tensor-structure assumption. In reality, the response matrix does not have a tensor structure in general, though the tensor structure could be a good approximation in many cases. We now implement our technique using response matrices measured experimentally on IBM QPUs and demonstrate the efficacy of the technique for realistic readout errors on NISQ devices.

We measured the response matrices on three different 27-qubit IBM QPUs, namely, "ibm_cairo," "ibm_hanoi," and "ibmq_toronto." The response matrices were estimated by

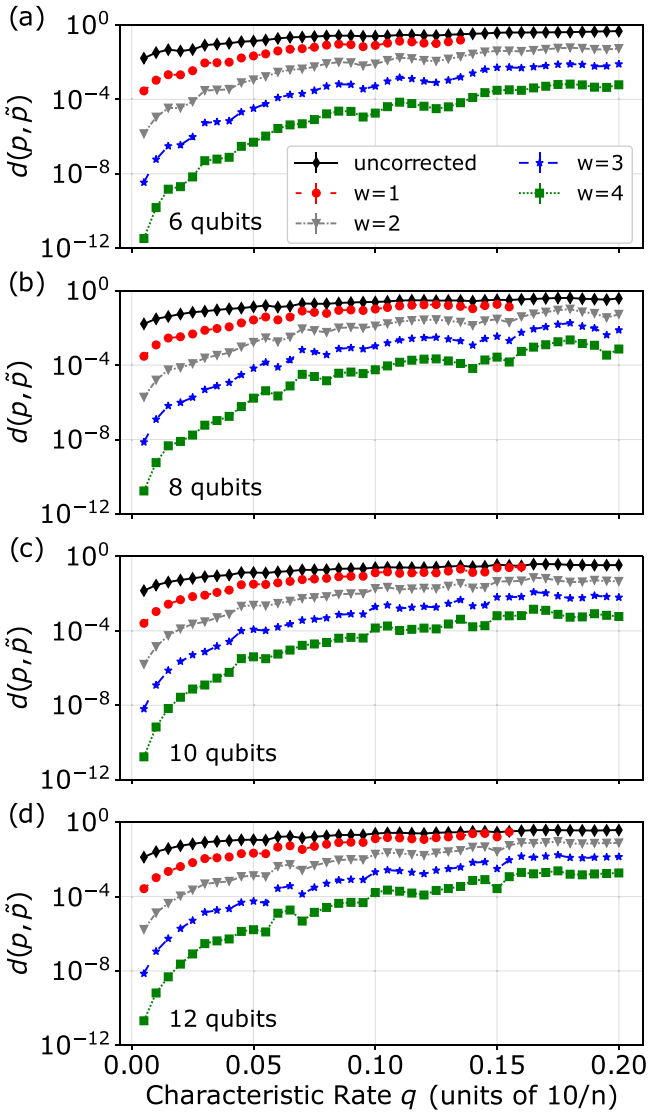

FIG. 4. Our technique can be modified to overcome the limitations shown in Fig. 3 by exactly inverting $\sum_{j=1}^{w} \mathcal{R}_j$. By doing so, the failure threshold $q_{\max}$ approaches 0.5, indicating that the technique will work for arbitrary R constructed according to the model we have employed.

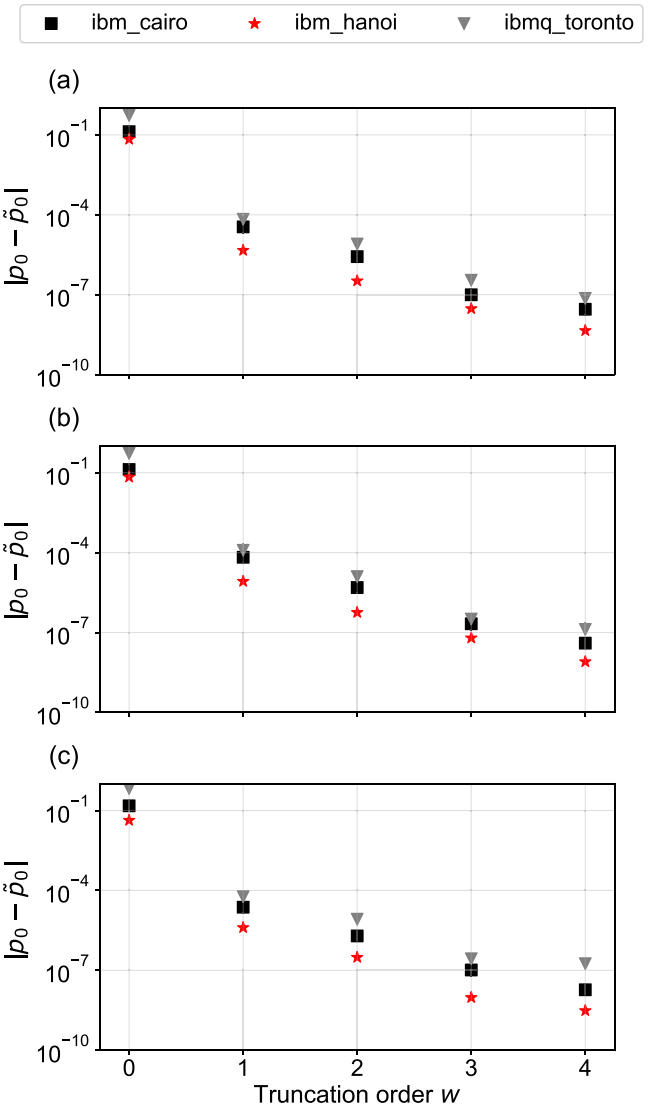

FIG. 5. Performance of all-zeros readout-error mitigation with response matrices experimentally determined on IBM QPUs. We demonstrate the performance using three prior distributions, namely, (a) Gaussian distribution, (b) truncated Gaussian distribution, and (c) uniform distribution. The details of the distributions are explained in the caption of Fig. 1.

preparing computational basis states $|j_1 \dots j_n\rangle$ using a sequence of X gates, i.e., $X_1^{j_1} \dots X_n^{j_n}$, and then determining the output distributions via parallel qubit readout. Each of these measurements requires 2^n different circuit executions, with each execution to be repeated several times to generate a distribution. To avoid exponential resource overhead, we estimated R for 12 out of 27 qubits using 10 000 shots per matrix element. In particular, we picked qubits 1, 2, 3, 5, 8, 11, 14, 16, 19, 22, 25, and 26 for our experiment. Note that all three QPUs have the same connectivity map.

Figure 5 shows the performance of our technique for recovering the all-zeros bit strings using the response matrices measured on IBM hardware. Similar to the result using a tensor-structure assumption shown in Fig. 1, the error is exponentially suppressed in the truncation order w.

2. Sampling error and comparison to "M3"

In this section, we provide additional numerical experiments comparing the performance of our method to existing methods. We provide preliminary evidence that our technique for estimating the all-zeros bit string provides comparable accuracy as the "M3" technique of Ref. [19] for instances tested on eight qubits. M3 performs readout-error correction by operating in a subspace corresponding to bit strings that were sampled in an experiment with finite repetitions. Thus, M3 corrects an empirical distribution $h' \in \mathbb{R}^{2^n}$ sampled according to the observed bit-string probability vector p' and takes as input a response matrix R sampled from calibration circuits on hardware. To introduce similar sampling error into our technique, we prepared independent qubit bit-flip probabilities $\epsilon_k, \eta_k \sim \text{Uniform}(0, q)$ and then prepared

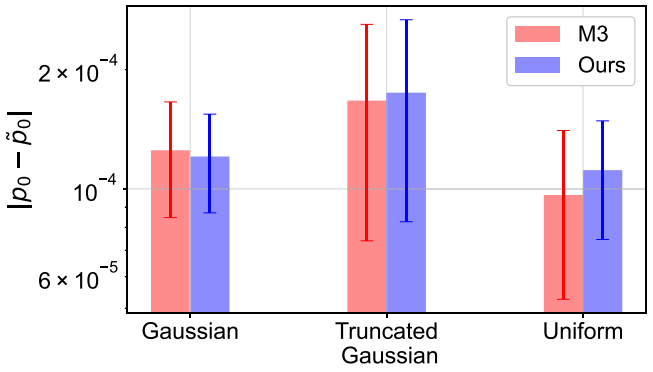

FIG. 6. The accuracy of our technique for recovering the all-zeros bit string is statistically indistinguishable to that of M3 for the distributions considered in Fig. 1 using $n = 8$ qubits, with $M = N = 10^6$ circuit repetitions used for both the calibration experiment and for sampling p' (see main text). The response matrix R has the same characteristic error rate q as in previous numerics. Error bars denote standard deviation.

estimates $\tilde{\epsilon}_k := B(N, \epsilon_k)/N$, $\tilde{\eta}_k := B(N, \eta_k)/N$ (where B is the binomial distribution) to simulate a series of independent calibration experiments using N circuit repetitions for each of qubits $k = 1, \ldots, n$. We then used a sampled response matrix,

$$\tilde{\mathsf{R}} = \bigotimes_{k=1}^{n} \begin{pmatrix} 1 - \tilde{\eta}_k & \tilde{\epsilon}_k \\ \tilde{\epsilon}_k & 1 - \tilde{\epsilon}_k \end{pmatrix}, \tag{E1}$$

in place of R for all parts of our algorithm. Similarly, we substituted an empirical estimate h' with components $h'_i = B(M, p'_i)/M$ for the probability vector over observed bit strings p' to simulate sampling a circuit run for M repetitions. In Fig. 6, we numerically simulate recovering the all-zeros bit string for the eight-qubit case using our technique and the M3 technique. The simulation shows that the two techniques give results with a similar accuracy.

[1] J. Preskill, Quantum **2**, 79 (2018).
[2] M. Neeley, R. C. Bialczak, M. Lenander, E. Lucero, M. Mariantoni, A. D. O'Connell, D. Sank, H. Wang, M. Weides, J. Wenner *et al.*, Nature (London) **467**, 570 (2010).
[3] D. Willsch, M. Willsch, F. Jin, H. De Raedt, and K. Michielsen, Phys. Rev. A **98**, 052348 (2018).
[4] A. Dewes, F. R. Ong, V. Schmitt, R. Lauro, N. Boulant, P. Bertet, D. Vion, and D. Esteve, Phys. Rev. Lett. **108**, 057002 (2012).
[5] Y. Chen, M. Farahzad, S. Yoo, and T.-C. Wei, Phys. Rev. A **100**, 052315 (2019).
[6] M. Gong, M.-C. Chen, Y. Zheng, S. Wang, C. Zha, H. Deng, Z. Yan, H. Rong, Y. Wu, S. Li *et al.*, Phys. Rev. Lett. **122**, 110501 (2019).
[7] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Nature (London) **567**, 209 (2019).
[8] K. X. Wei, I. Lauer, S. Srinivasan, N. Sundaresan, D. T. McClure, D. Toyli, D. C. McKay, J. M. Gambetta, and S. Sheldon, Phys. Rev. A **101**, 032343 (2020).
[9] M. Urbanek, B. Nachman, and W. A. de Jong, Phys. Rev. A **102**, 022427 (2020).
[10] B. Nachman, M. Urbanek, W. A. de Jong, and C. W. Bauer, npj Quantum Inf. **6**, 84 (2020).
[11] R. Hicks, C. W. Bauer, and B. Nachman, Phys. Rev. A **103**, 022407 (2021).
[12] L. Funcke, T. Hartung, K. Jansen, S. Kühn, P. Stornati, and X. Wang, Phys. Rev. A **105**, 062404 (2022).
[13] S. Bravyi, S. Sheldon, A. Kandala, D. C. Mckay, and J. M. Gambetta, Phys. Rev. A **103**, 042605 (2021).
[14] E. van den Berg, Z. K. Minev, and K. Temme, Phys. Rev. A **105**, 032620 (2022).
[15] E. Peters, J. Caldeira, A. Ho, S. Leichenauer, M. Mohseni, H. Neven, P. Spentzouris, D. Strain, and G. N. Perdue, npj Quantum Inf. **7**, 161 (2021).
[16] B. Nachman and M. R. Geller, arXiv:2104.04607.
[17] M. R. Geller, Quantum Sci. Technol. **5**, 03LT01 (2020).
[18] For simplicity, we assume that both inversion and multiplication of generic $M \times M$ matrices have complexity $O(M^3)$. Optimized algorithms such as Strassens' algorithm reduce this complexity, but these considerations will not affect the *relative* speedups that we present in this work.
[19] P. D. Nation, H. Kang, N. Sundaresan, and J. M. Gambetta, PRX Quantum **2**, 040326 (2021).
[20] B. Yang, R. Raymond, and S. Uno, Phys. Rev. A **106**, 012423 (2022).
[21] M. Schuld and N. Killoran, Phys. Rev. Lett. **122**, 040504 (2019).
[22] M. Huo and Y. Li, Phys. Rev. A **105**, 022427 (2022).
[23] E. Peters, P. Shyamsundar, A. C. Y. Li, and G. Perdue, PRX Quantum **3**, 040333 (2022).
[24] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Phys. Rev. A **98**, 032309 (2018).
[25] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, Quantum **3**, 140 (2019).
[26] M. R. Geller and M. Sun, Quantum Sci. Technol. **6**, 025009 (2021).
[27] R. J. Lipton, D. J. Rose, and R. E. Tarjan, SIAM J. Numer. Anal. **16**, 346 (1979).
[28] E. Kılıç and P. Stanica, J. Comput. Appl. Math. **237**, 126 (2013).
[29] K. Wang, Y.-A. Cheng, and X. Wang, arXiv:2103.13856.
[30] H. Lutkepohl, in *Handbook of Matrices* (John Wiley & Sons, Hoboken, New Jersey, 1997), Vol. 2, p. 167.
[31] A. Ozdaglar, D. Shah, and C. L. Yu, arXiv:1411.2647.