Quantum error mitigation by layerwise Richardson extrapolation

Vincent Russo[1,*] and Andrea Mari[1,2]

[1] *Unitary Fund*

[2] *Physics Division, School of Science and Technology, Università di Camerino, 62032 Camerino, Italy*

A widely used method for mitigating errors in noisy quantum computers is Richardson extrapolation, a technique in which the overall effect of noise on the estimation of quantum expectation values is captured by a single parameter that, after being scaled to larger values, is eventually extrapolated to the zero-noise limit. We generalize this approach by introducing *layerwise Richardson extrapolation (LRE)*, an error mitigation protocol in which the noise of different individual layers (or larger chunks of the circuit) is amplified and the associated expectation values are linearly combined to estimate the zero-noise limit. The coefficients of the linear combination are analytically obtained from the theory of multivariate Lagrange interpolation. LRE leverages the flexible configurational space of layerwise unitary folding, allowing for a more nuanced mitigation of errors by treating the noise level of each layer of the quantum circuit as an independent variable. We provide numerical simulations demonstrating scenarios where LRE achieves superior performance compared to traditional (single-variable) Richardson extrapolation.

I. INTRODUCTION

In recent years, the field of quantum technologies has witnessed extraordinary progress, especially in the evolution of noisy intermediate-scale quantum (NISQ) devices. Despite their capacity to excel over classical devices in certain tasks [1–7], NISQ devices are notably hindered by substantial noise, adversely affecting their output.

As we await the advent of fault-tolerant devices [8], a significant field of exploration for addressing the prevalent noise issues is quantum error mitigation (QEM) [9–23]. QEM serves as an intermediate approach to fault tolerance that can be realized at present to overcome the hurdle of noisy devices. There are a variety of QEM techniques that are the subject of active research, for example, zero-noise extrapolation (ZNE) [10, 11, 13], probabilistic-error cancellation (PEC) [11, 12, 24, 25], dynamical decoupling [26–29], and Clifford data regression [30, 31].

In this work, we focus on ZNE, a technique that has been used in many quantum computing experiments [13, 16, 18, 32–34] and that has shown strong performance despite the simplicity of its practical implementation. For a given quantum circuit, the primary idea of ZNE contains two steps; intentionally scaling up the noise of the circuit and then extrapolating to the noiseless limit.

For the first step, there are several techniques one can consider to intentionally increase the noise, one of which is *unitary folding* [15, 35]; a process that increases the length of the quantum circuit, and by proxy, the noise. The second step is achieved by fitting a curve to the expectation values measured at different noise levels to extrapolate to the noiseless expectation value. One such method, *Richardson extrapolation* (RE) [11], corresponds to a single-variable polynomial interpolation of the noise-scaled expectation values.

In this work, we generalize Richardson extrapolation to a multivariate framework in which we consider multiple independent noise parameters associated with the different layers (or with different chunks) of the full circuit. We call this new approach *layerwise Richardson extrapolation* (LRE), while we use the acronym RE for the conventional approach based on single-variable Richardson extrapolation. To generalize RE to the multivariate LRE technique, we need to address two sub-problems: (i) A way of scaling up the noise of specific layers, without perturbing the rest of the circuit. (ii) A way of post-processing the information obtained from the (layerwise) noise-scaled circuits to infer the zero-noise limit.

A noise-scaling strategy that can be used to solve the first sub-problem is *layerwise folding* [36, 37]: an approach that considers a quantum circuit as being comprised of several layers and where a variable amount of *folding* [15, 35] can occur at any given layer of the circuit. Layerwise folding has been used in [36, 37] as a circuit debugging technique, for example, to assess what layers in a quantum circuit are particularly susceptible to noise. Instead, in this work, we are not interested in using layerwise folding as an error characterization method, but as an error mitigation tool.

The second sub-problem that we need to solve is how to generalize Richardson extrapolation in a framework in which the expectation value of an observable can be considered as a multivariate function of the noise levels associated with different layers. We address this sub-problem by applying the mathematical theory of multivariate Lagrange interpolation [38, 39], which allows us to express the zero-noise limit as a linear combination of the noise-scaled expectation values, each one weighted with a suitable real coefficient which only depends on the noise scaling factors.

It is interesting to compare the characteristic features of LRE for two similar techniques: PEC and RE. Like PEC, LRE involves a linear combination of many circuits

* vincent@unitary.fund

in which only some specific layers are changed, while the rest of the circuit is kept unmodified. Unlike PEC but similar to RE, LRE does not necessitate full knowledge of the noise model. This is because the generation of modified circuits in LRE is deterministic and solely depends on the choice of the noise scale factors. It is also worth noting that for the case of linear extrapolation, LRE reduces to the noise-scaling variant of the NOX method described in the Appendix of [40]. A further interesting connection is to the NEPEC technique introduced in [41], in which noise scaling has been proposed as a way to build quasi-probability representations of individual gates (or layers) to be used for probabilistic error cancellation. Our technique is also related to [42], in which ZNE has been proposed for mitigating a multi-parameter noise model.

In [42], however, the parameters are associated with different physical errors acting uniformly along the circuit (e.g. the values of T_1 and T_2 for a qubit), noise scaling is obtained by running the same circuit on different qubits, and the final extrapolation is obtained by a numerical best fit. In this work instead, we tune the noise level of different layers by using localized folding operations and without introducing additional qubits. Moreover, instead of using a numerical best fit, we provide an analytic expression for the zero-noise limit based on the theory of Richardson extrapolation.

This article is organized as follows. In Section II, we formally define the LRE technique and describe the noise scaling (Section II A) and extrapolation strategies (Section II B). We also consider how one can apply LRE to a circuit in chunks (Section II C) as well as the sampling overhead of LRE (Section II D). In Section III, we showcase some examples and numerical experiments using LRE, illustrating its practical advantages and limitations. We conclude in Section IV with future directions and potential applications for the LRE technique.

II. LAYERWISE RICHARDSON EXTRAPOLATION (LRE)

In this section we present the layerwise Richardson extrapolation (LRE) technique, for the mitigation of errors acting on quantum circuits. Much like RE, it consists of two major steps; noise scaling and extrapolation. For noise scaling, in Section II A, we evaluate an expectation value at different vectors of scale factors via a layerwise folding approach. For extrapolation and post-processing of these expectation values, covered in Section II B, we make use of the mathematical theory of multivariate Lagrange interpolation.

A. Noise scaling

In RE, one of the mechanisms that is often used to scale the noise is *unitary folding* [15, 35]. A more targeted way in which the noise can be scaled is to apply *layerwise folding*, as proposed in, for instance, [36, 37]. Instead of increasing the depth of the entire circuit considered as a single global entity, layerwise folding acts on specific layers of the circuit (see Figure 1).

An n-qubit quantum circuit C may be represented as a series of ℓ layers. Each layer L_k for $1 \leq k \leq \ell$ contains one or more quantum gates acting concurrently on an n-qubit system

$$C = L_\ell L_{\ell-1} \cdots L_2 L_1. \tag{1}$$

In what follows, we denote each term L_k as a *layer*. However, the full theory of LRE is equally applicable assuming that each L_k represents a multi-layer *chunk* of the full circuit (see Section II C for more details).

Consider a collection of N different scale factor vectors

$$\Lambda = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_N\}, \tag{2}$$

where each $\boldsymbol{\lambda}_i$ is a vector of ℓ scale factors that specifies how the noise is scaled across different layers

$$\boldsymbol{\lambda}_i = \left(\lambda_1^{(i)}, \lambda_2^{(i)}, \ldots, \lambda_\ell^{(i)}\right), \quad \lambda_k^{(i)} \geq 1. \tag{3}$$

For a collection of scale factor vectors defined by Λ, we denote

$$C^\Lambda = \{C^{\boldsymbol{\lambda}_1}, C^{\boldsymbol{\lambda}_2}, \ldots, C^{\boldsymbol{\lambda}_N}\} \tag{4}$$

as the corresponding set of circuits. Each circuit in C^Λ is layerwise noise-scaled according to the corresponding scale factor vector.

While layerwise folding is our chosen method for scaling the noise, it is important to note that this approach is not the only option. In principle, various methods can be employed to selectively scale the noise of specific circuit layers. For example, a promising alternative is given by the pulse-stretching method [11, 13], assuming the possibility of applying different stretchings to different layers.

For layerwise folding, each scale factor $\lambda_k^{(i)}$ corresponds to the k-th layer L_k of the circuit C and is defined as

$$\lambda_k^{(i)} = 1 + 2m_k^{(i)}. \tag{5}$$

Here, $m_k^{(i)}$ is a non-negative integer representing the number of times the k-th layer L_k is to be folded. The folding operation [15, 35] for each layer L_k is expressed as

$$L_k^{\lambda_k^{(i)}} = \left(L_k L_k^\dagger\right)^{m_k^{(i)}} L_k, \tag{6}$$

where $L_k^{\lambda_k^{(i)}}$ is the new k-th layer after the folding operation. If L_k represents a chunk of the circuit that is itself composed of t elementary sub-layers $L_k = G_{k,t} \cdots G_{k,2} G_{k,1}$, unitary folding can be applied in different ways. One option, known as *global folding* [15], corresponds to Equation (6). A common alternative option, known as *local folding* [15], is instead:

$$L_k^{\lambda_k^{(i)}} = (G_{k,t} G_{k,t}^\dagger)^{m_k^{(i)}} G_{k,t} \cdots (G_{k,1} G_{k,1}^\dagger)^{m_k^{(i)}} G_{k,1}. \tag{7}$$

Both methods scale the depth of L_k by $\lambda_k^{(i)} \geq 1$ and are exactly equivalent when $t = 1$. For large t, one can apply unitary folding partially or randomly [15, 35], such that the scale factor $\lambda_k^{(i)}$ is not constrained to take the odd integer values as implied by Equation (5). However, for simplicity, in this work, we always assume odd integer scale factors since they are always implementable for any t.

For a given vector $\boldsymbol{\lambda}_i$ of scale factors, the resulting circuit $C^{\boldsymbol{\lambda}_i}$ is represented as

$$C^{\boldsymbol{\lambda}_i} = L_1^{\lambda_1^{(i)}} L_2^{\lambda_2^{(i)}} \cdots L_{\ell-1}^{\lambda_{\ell-1}^{(i)}} L_\ell^{\lambda_\ell^{(i)}}. \tag{8}$$

In this framework, the vector of scale factors $\boldsymbol{\lambda}_i$ explicitly defines which layers of the circuit are to be folded and the number of times each specified layer is folded. A depiction of the folding operation for an arbitrary circuit is shown in Figure 1.

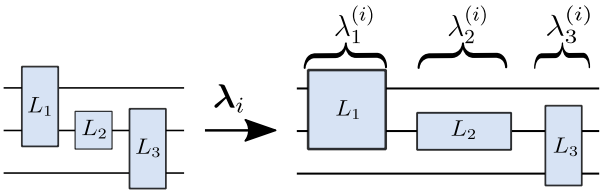

Figure 1. An arbitrary quantum circuit consisting of three layers; L_1, L_2, and L_3. The circuit on the right is constructed according to a vector of noise scale factors $\boldsymbol{\lambda}_i = (\lambda_1^{(i)}, \lambda_2^{(i)}, \lambda_3^{(i)})$ that determines how much the depth of each layer is scaled up by unitary folding (or by any other noise scaling method which can act layerwise). Without noise, the two circuits are equivalent. With noise, the circuit on the right is subject to more errors. Moreover, noise is amplified on some specific layers and less amplified (or unchanged) on other layers.

For each circuit in C^Λ from Equation (4), one may compute the corresponding expectation value of a fixed observable of interest O. Specifically, we denote all the expectation values associated with C^Λ as

$$\mathbf{z} = (\langle O(\boldsymbol{\lambda}_1)\rangle, \langle O(\boldsymbol{\lambda}_2)\rangle, \ldots, \langle O(\boldsymbol{\lambda}_N)\rangle)^\intercal \tag{9}$$

where $\langle O(\boldsymbol{\lambda}_i)\rangle$ is the expectation value of the observable O, estimated from the execution of the circuit $C^{\boldsymbol{\lambda}_i}$.

B. Extrapolation

Once we have scaled the noise via the layerwise folding approach discussed in Section II A and obtained a vector of expectation values evaluated at different vectors of scale factors as in Equation (9), we proceed to post-process this raw data by a multivariate generalization of Richardson extrapolation.

For a vector of scale factors $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_\ell)$, we define the basis of all monomial terms of ℓ variables of maximum degree d as $\mathcal{M}(\boldsymbol{\lambda}, d)$. For instance, for $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ and $d = 2$, we have

$$\mathcal{M}(\boldsymbol{\lambda}, 2) = \{1, \lambda_1, \lambda_2, \lambda_1^2, \lambda_1\lambda_2, \lambda_2^2\}. \tag{10}$$

In general, the number of monomial terms is given by

$$M \equiv |\mathcal{M}(\boldsymbol{\lambda}, d)| = \binom{d+\ell}{d}, \tag{11}$$

and we assume that the monomials are ordered with an increasing total degree. For example, a typical choice is the graded lexicographic order [43]. This implies that the first element of the list of monomials is 1, i.e., the term of zero degree that survives when taking the zero-noise limit $\boldsymbol{\lambda} \to \mathbf{0}$, where $\mathbf{0}$ is the all-zero vector.

For our purposes, typical values of the maximum degree are $d = 1$ or $d = 2$, corresponding to a linear scaling $M = \ell + 1$ and a quadratic scaling $M = (\ell + 1)(\ell + 2)/2$ of the number of terms, respectively. More generally, for a fixed extrapolation order d, the number of monomials M scales polynomially with respect to ℓ since we have

$$M = \frac{1}{d!} \prod_{i=1}^{d} (\ell + i) = \mathcal{O}(\ell^d). \tag{12}$$

We aim to interpolate a multivariate polynomial function that captures the relationship between the vectors of scale factors and the expectation values as defined from Equation (9). Specifically, we model the dependence of the expectation value as a function of the noise scale factors as a generic polynomial of degree d

$$\langle O(\boldsymbol{\lambda})\rangle = \sum_{j=1}^{M} c_j \mathcal{M}_j(\boldsymbol{\lambda}, d), \tag{13}$$

where $\{c_j\}$ are real coefficients. We are particularly interested in extrapolating Equation (13) to the zero-noise limit, that is

$$O_{\mathrm{LRE}} \equiv \langle O(\mathbf{0})\rangle = \sum_{j=1}^{M} c_j \mathcal{M}_j(\mathbf{0}, d) = c_1. \tag{14}$$

Given the collection Λ of scale factor vectors, as defined in Equation (2), we define the *sample matrix*

$$\mathbf{A}(\Lambda, d) = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,M} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,M} \end{bmatrix}, \tag{15}$$

where each entry in the matrix is defined as

$$a_{i,j} = \mathcal{M}_j(\boldsymbol{\lambda}_i, d). \tag{16}$$

As a notational convention, we often write Equation (15) as just $\mathbf{A}$, whenever it is clear what the values of Λ and d are. Each row of $\mathbf{A}$ corresponds to a specific scale

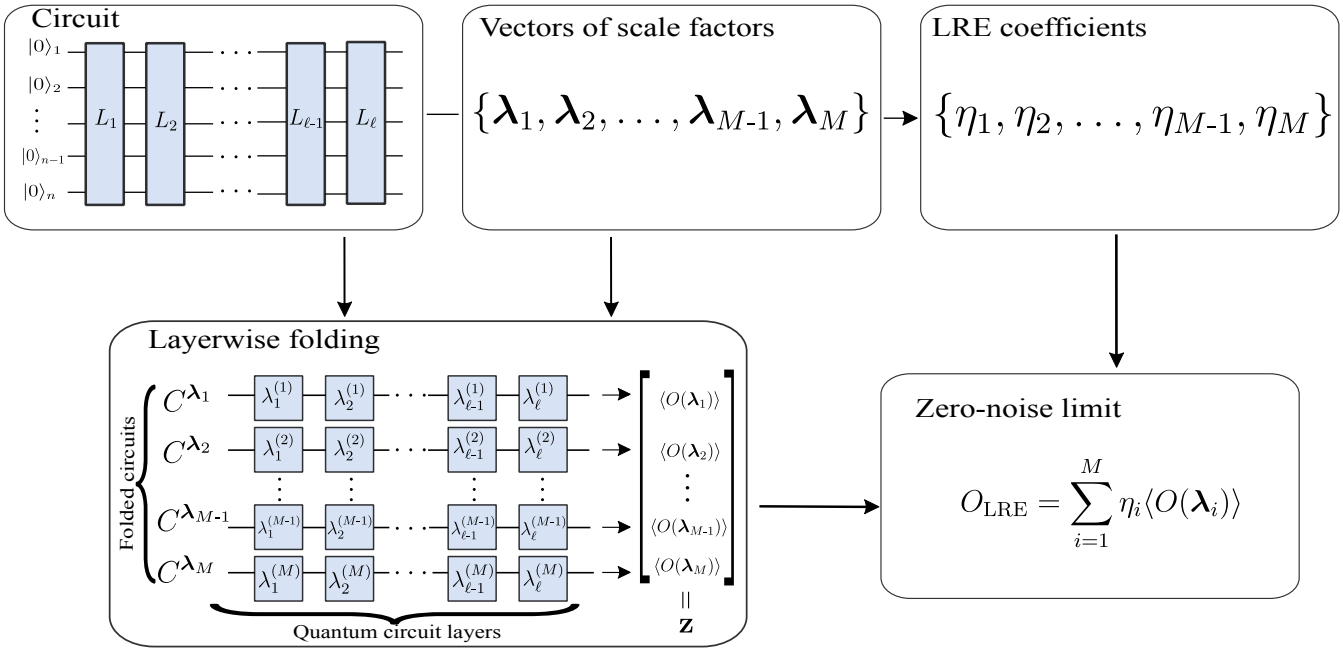

Figure 2. An overview of the LRE experimental workflow. As input, we consider an n-qubit quantum circuit consisting of ℓ layers or, equivalently, l circuit chunks. Given the parameter l and the extrapolation order d, we generate $M = \binom{d+l}{d}$ linearly-independent vectors of scale factors (see Equation (21) for a convenient generation pattern). In practice, for each vector of scale factors, one can set most elements to 1 (no noise scaling) and assign larger values to just a few elements. From this, we perform layerwise folding on the input circuit generating M different circuits, one for each vector of scale factors. Each generated circuit is almost identical to the input one, except for a few layers that are folded to amplify their noise sensitivity. For each resulting circuit (Equation (4)), we experimentally estimate the respective expectation value (Equation (9)). The linear combination coefficients $\{\eta_j\}$ can be computed straightforwardly from the multivariate Lagrange interpolation formula (Equation (20)) and, remarkably, they only depend on the scale factor vectors. By taking a linear combination of the noise-scaled expectation values, we obtain the error-mitigated result.

factor vector, while each column corresponds to a specific monomial. The interpolation problem can be cast as a linear system,

$$\mathbf{A}\mathbf{c} = \mathbf{z}, \tag{17}$$

where $\mathbf{z}$ is the known vector of noise-scaled expectation values as defined in Equation (9) and $\mathbf{c} = (c_1, \ldots, c_M)^\intercal$ is the unknown vector of coefficients defined in Equation (13). In principle, solving for $\mathbf{c}$, one can determine all the coefficients of the interpolating polynomial, which can be used to evaluate new domain points, including the zero-noise limit ($\langle O(\mathbf{0})\rangle = c_1$). However, if we are only interested in the zero-noise limit, it is not necessary to evaluate the full vector of coefficients $\mathbf{c}$. We will use the theory of Lagrange interpolation to obtain a simple formula that directly provides the zero-noise limit.

To have a unique solution for the system of equations, we assume that the sample matrix is square and that its determinant is non-zero. This implies that the number N of different scale factor vectors is not arbitrary but it must be equal to the number of monomials, i.e.,

$$N = M \quad \text{and} \quad \det\left(\mathbf{A}(\Lambda, d)\right) \neq 0. \tag{18}$$

In practice, for a given number of layers ℓ and a given degree d of the interpolating polynomial, the number of different noise scaling configurations and the number of different expectation values that one needs to measure is given by Equation (11). Note that assuming $\det(\mathbf{A}) \neq 0$ is not a strong limitation since, in the case of a zero (or close to zero) determinant, one can always change some of the scale factor vectors in such a way to avoid an ill-conditioned system of equations.

By a direct application of the theory of multivariate Lagrange interpolation (as shown in Appendix V B), we can obtain the zero-noise limit via the following linear combination of the noisy expectation values

$$O_{\mathrm{LRE}} = \sum_{i=1}^{M} \eta_i \langle O(\boldsymbol{\lambda}_i)\rangle, \tag{19}$$

where the coefficients are given by

$$\eta_i = \frac{\det\left(\mathbf{M}_i\right)}{\det\left(\mathbf{A}\right)}, \tag{20}$$

where $\mathbf{M}_i$ is the matrix obtained from $\mathbf{A}$ after replacing the i-th row by the vector $\mathbf{e}_1 = (1, 0, \ldots, 0)$ consisting of a 1 followed by zeros.

C. Applying LRE to chunks of the circuit

As we anticipated in Section II A, if instead of decomposing the circuit into a sequence of elementary layers (of depth one) we split the circuit into chunks of arbitrary depth, the whole theory of LRE is equally applicable. Indeed, in the theoretical derivation developed in the previous sections, we never had to invoke any assumption on the actual depth of each term L_k in Equation (1).

In practice, this means that the total number of chunks l in Equation (1) is an arbitrary hyperparameter of LRE that we are free to choose at our convenience. This flexibility allows us to interpolate from $l = 1$ corresponding to traditional (single-chunk) RE, up to $l = l_{\max}$, where $l_{\max}$ is the maximum number of elementary layers of the circuit.

Operationally, given a circuit C of depth $l_{\max}$ and a target observable O, the implementation of LRE corresponds to the following protocol (see also Figure 2):

1. Choose the hyperparameters: the extrapolation order d, the number of splittings $l \leq l_{\max}$, and the minimum noise scaling gap Δ. By default, use $\Delta = 2$ (minimum gap allowed by unitary folding). See Section II D for more details on hyperparameters.

2. Compute the number M of degrees of freedom using Equation (11).

3. Choose M different vectors of scale factors $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_M$. A simple choice is the following

$$\boldsymbol{\lambda}_i = \mathbf{1} + \boldsymbol{m}_i \Delta, \quad i = 1, 2, \ldots, M, \tag{21}$$

   where $\mathbf{1} = (1, 1, \ldots)$ and $\{\boldsymbol{m}_i\}$ are all the vectors of l non-negative integers with $\|\boldsymbol{m}_i\|_1 \leq d$.

4. Evaluate the corresponding M real coefficients $\eta_1, \eta_2, \ldots, \eta_M$ using Equation (20).

5. Split C into l chunks and apply layerwise folding as defined in Equations (5-8), generating M noise-scaled circuits $C^{\boldsymbol{\lambda}_1}, C^{\boldsymbol{\lambda}_2}, \ldots, C^{\boldsymbol{\lambda}_M}$.

6. Evaluate the corresponding M expectation values $\langle O(\boldsymbol{\lambda}_1)\rangle, \langle O(\boldsymbol{\lambda}_2)\rangle, \ldots, \langle O(\boldsymbol{\lambda}_M)\rangle$ on the quantum computer.

7. Compute the error-mitigated result using $O_{\mathrm{LRE}} = \sum_{i=1}^{M} \eta_i \langle O(\boldsymbol{\lambda}_i)\rangle$.

Note that only Step 6 of the above protocol involves the actual usage of a quantum computer, all the other steps are just classical pre- or post-processing.

D. Sampling overhead of LRE

The error-mitigated expectation value obtained from layerwise Richardson extrapolation is subject to statistical uncertainty. Each noisy expectation value in the right-hand side of Equation (19) must be measured with a finite number of shots and, therefore, each term will be subject to a statistical error (shot noise). After taking the linear combination, the left-hand side of the equation will be subject to statistical uncertainty due to the propagation of the statistical error of each term on the right-hand side.

For a fixed target of statistical error, the total number of shots s_{tot} required to evaluate Equation (19) is larger than the number of shots s_{u} required to directly estimate the unmitigated expectation value $\langle O(\boldsymbol{\lambda})\rangle|_{\boldsymbol{\lambda}=\mathbf{1}}$. The sampling overhead required to apply LRE is captured by the ratio $s_{\mathrm{tot}}/s_{\mathrm{u}}$. Assuming all the noisy expectation values of Equation (19) have equal variance and that they are measured with the same number of shots s_{tot}/M, it is easy to show [9] that:

$$\tilde{c} := \frac{s_{\mathrm{tot}}}{s_{\mathrm{u}}} = M\tilde{\gamma}^2, \quad \tilde{\gamma} := \left(\sum_{i=1}^{M} |\eta_i|^2\right)^{\frac{1}{2}}, \; s_i = \frac{s_{\mathrm{tot}}}{M}. \tag{22}$$

However, the sampling overhead can be reduced by using more shots on the terms that are more "important" in the linear combination of Equation (19). For a fixed total budget of shots s_{tot}, it is more convenient to invest $s_i \propto |\eta_j|$ shots when estimating each noise-scaled expectation value $\langle O(\boldsymbol{\lambda}_i)\rangle$. In this case, we have [9, 11]:

$$c := \frac{s_{\mathrm{tot}}}{s_{\mathrm{u}}} = \gamma^2, \quad \gamma := \sum_{i=1}^{M} |\eta_i|, \quad s_i = \frac{s_{\mathrm{tot}}|\eta_i|}{\gamma}. \tag{23}$$

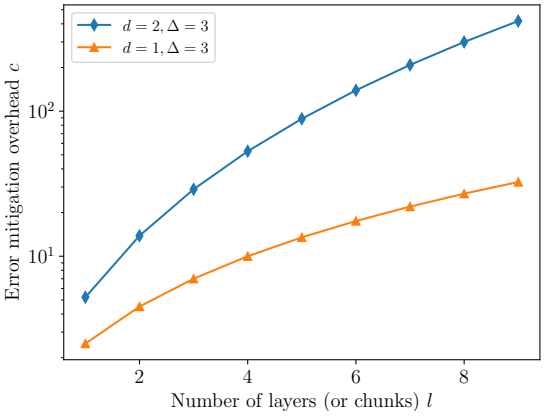

Figure 3. Sampling overhead of layerwise Richardson extrapolation for quadratic ($d = 2$) and linear ($d = 1$) interpolation as a function of the number of layers (or circuit chunks). The overhead is estimated according to Equation (23) assuming the specific choice of scale factors given in Equation (21), with $\Delta = 2$ (the minimum gap achievable via layerwise folding). The noise of each circuit chunk is scaled by *local folding* as defined in Equation (7).

The fact that the one norm γ of the linear combination of coefficients in (19) is related to the error mitigation overhead is well-known in the error mitigation

literature [9, 11, 44], and it is a consequence of Hoeffding's inequality applied within the context of probabilistic Monte-Carlo algorithms. Here we have just confirmed that the same result also holds for deterministic LRE, assuming that each expectation value in (19) is measured with the appropriate number of shots. As a direct consequence of the Cauchy-Schwartz inequality (see e.g. [9]), we have $\tilde{c} > c$, meaning that Equation (23) is the appropriate figure of merit for the optimal sampling cost. On the other hand, in real experiments, it can be more practical to estimate noisy expectation values with the same number of shots s_{tot}/M for each noise-scaled circuit. In this case, $\tilde{c}$ is a more appropriate estimate of the sampling cost.

In Figure 3 we plot c as a function of the number of layers (or circuit chunks) l and for different values of the extrapolation order d. In this figure, we keep fixed the minimum gap between scale factors $\Delta = 2$, corresponding to the minimum gap of noise scaling achievable with folding operations.

1. *Methods for reducing the sampling overhead*

In Figure 4, we fix $l = 10$ and show the dependence of the sampling overhead as a function of the minimum gap between scale factors $\Delta = 2, 4, 6, \ldots$ corresponding to a gap in the number of folding operations equal to $1, 2, 3, \ldots$, respectively (see Equation (5)). We observe that using a large gap between scale factors reduces the sampling cost. On the other hand, high values of noise scaling can increase the bias of the polynomial extrapolation, since the noisy expectation value is sampled further away from the zero-noise limit. Therefore, by altering Δ one can change the variance-bias tradeoff of the error-mitigated result.

Another simple way of reducing the overhead is by splitting the full circuit into a smaller number of chunks l, where each chunk contains multiple elementary layers. From Figure 3, it is clear that using a small value of l is a direct way of reducing the sampling cost.

In practice, even for very deep circuits, we can always keep the overhead of LRE under control by setting an upper bound to the number of splittings l or by increasing Δ, at the cost of increasing the estimation bias (see Sections III A 4 and III A 5 for numerical examples).

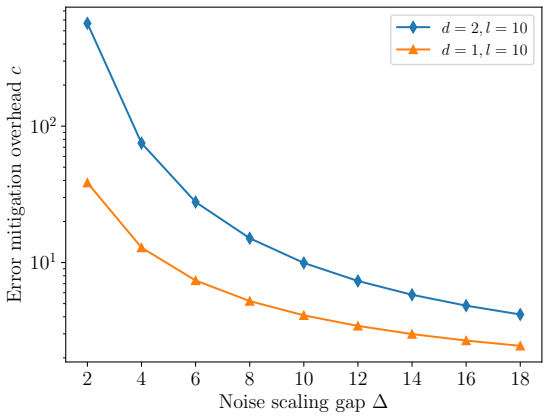

Figure 4. Sampling overhead of layerwise Richardson extrapolation for quadratic ($d = 2$) and linear ($d = 1$) interpolation as a function of the minimum gap between scale factors Δ. For both curves, we assume the same number of layers (or circuit chunks) $l = 10$. The vectors of scale factors are chosen according to Equation (21).

III. NUMERICAL EXPERIMENTS

In the previous section, we presented the theory of layerwise Richardson extrapolation. In this section, we test the technique with several numerical experiments to understand its practical advantages and its limitations. In particular, we focus on a systematic comparison between LRE and traditional single-variable Richardson extrapolation (RE).

A convenient choice of circuits for benchmarking error mitigation strategies are those which, without noise, restore all the qubits to the initial state $|00\ldots\rangle$. In this case, by taking as a target observable the projector on the zero state, i.e. $O = |00\ldots\rangle\langle 00\ldots|$, the ideal expectation value is always equal to 1 for a noiseless quantum computer. For a noisy backend instead, we can quantify the performance of different mitigation strategies by checking how close their associated predictions are to the ideal value of 1. For all of the quantum circuits simulated in this section, we assume a local amplitude damping noise model as described in Appendix V A.

In our analysis, we always fix the same total budget of shots s_{tot} that must be used by each error mitigation strategy (trivial unmitigated, LRE, RE, etc.). This means that if an error mitigation technique requires running M circuits, the total budget of shots is optimally split among the M circuits such that the total sum of circuit executions is kept constant, i.e. $\sum_{i=1}^{M} s_i = s_{\text{tot}}$. For both LRE and RE, we use the optimal splitting s_i defined in Equation (23) (recalling that RE is a special case of LRE with $l = 1$). If not explicitly specified, we fix a total budget of $s_{\text{tot}} = 10^6$ shots.

A. Benchmarking LRE with GHZ-like circuits

The first type of benchmark circuit that we use to test LRE is based on the concatenation of a GHZ circuit followed by its inverse, as shown in Figure 5.

The intermediate states during the execution of a GHZ-like circuit are highly entangled and, therefore, highly sensitive to environmental noise and decoherence. For this reason, they provide a good playground for test-

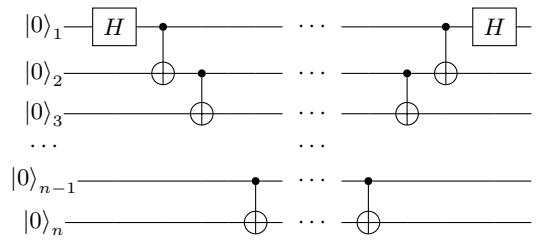

Figure 5. A GHZ-like benchmarking circuit composed of an n-qubit GHZ circuit followed by its inverse. By construction, the expectation value of $O = |00\ldots\rangle\langle 00\ldots|$ evaluated on an ideal noiseless device is equal to 1.

ing the efficacy of LRE on structured, entangling circuits.

1. *Vary over number of layers*

In Figure 6 and Table I, we compare the performance of LRE relative to RE and the unmitigated case as the number of layers l increases (the number of qubits increases as well since $l = 2n$).

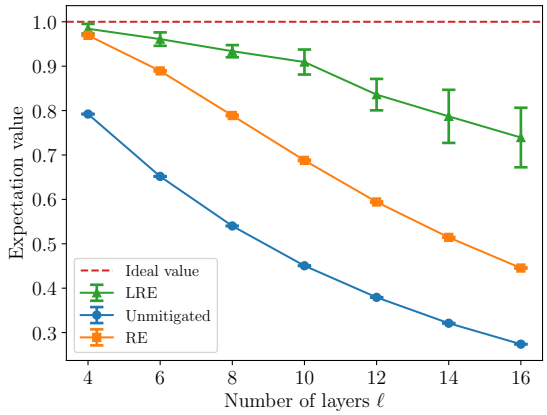

Figure 6. Expectation value of the observable $O = |00\ldots\rangle\langle 00\ldots|$ estimated with different error mitigation strategies for a GHZ-like circuit as defined in Figure 5. Each data point is averaged over 10 trials. For each trial, a total budget of $s_{\text{tot}} = 10^6$ shots is used. Error bars for each data point represent the standard deviation over the 10 independent trials. For all the data points considered in this example, layerwise Richardson extrapolation (LRE) is more accurate than traditional single-variable Richardson extrapolation (RE) and direct unmitigated estimation.

Increasing the size of a GHZ circuit elevates its complexity and susceptibility to errors. As expected, the estimation error increases with l for all the results but, for each l, the expectation value estimated with LRE is closer to the ideal value. Error bars are evaluated by repeating the same experiment for 10 trials and computing the standard deviation of the results. We observe that LRE results are subject to higher statistical uncertainty. This is expected from the overhead analysis presented in Section II D and from Figure 3: the error mitigation cost c of LRE increases with the number of layers (or circuit chunks) and, for a fixed budget of shots $s_{\text{tot}} = 10^6$, this implies a proportional increase of the statistical variance. Note however that, even taking into account error bars, the overall estimation error of LRE is smaller than RE due to a strong reduction of the estimation bias.

Depth	Unmitigated	RE	LRE	Improvement
2	0.2078	0.0306	0.0174	75.41%
3	0.3483	0.1107	0.0390	183.75%
4	0.4599	0.2110	0.0662	218.79%
5	0.5495	0.3121	0.0906	244.34%
6	0.6206	0.4058	0.1640	147.40%
7	0.6789	0.4856	0.2130	127.98%
8	0.7261	0.5546	0.2607	112.76%

Table I. Table of mean absolute estimation errors for each data point reported in Figure 6. The last column provides a percentage of improvement for the performance of layerwise Richardson extrapolation (LRE) over single-variable Richardson extrapolation (RE).

2. *Vary over extrapolation order*

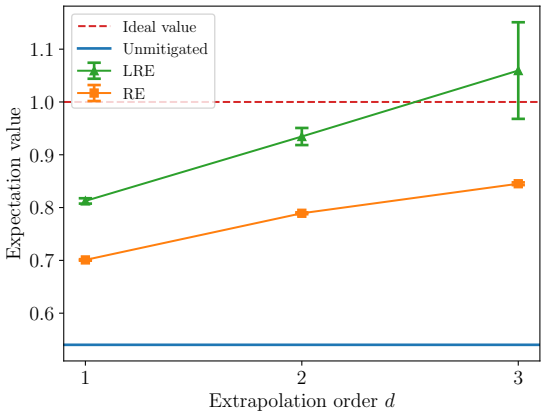

Figure 7. Expectation value estimated with layerwise Richardson extrapolation (LRE) and single-variable Richardson extrapolation (RE) for different values of the extrapolation degree d (linear, quadratic, cubic). As a benchmark circuit we used a 4-qubit GHZ-like circuit having the structure shown in Figure 5 and as observable we used $O = |00\ldots\rangle\langle 00\ldots|$.

In Figure 7, we explore how the performance of LRE and RE varies with the extrapolation order d, i.e., the degree of the interpolating polynomial. Specifically, for both LRE and RE, we compare the results obtained via linear, quadratic, and cubic extrapolation. As expected,

the bias of both LRE and RE decreases with the extrapolation order d. However, statistical noise increases (exponentially) with d. In practice, for real-world use cases, we expect LRE and RE to be useful for $1 \leq d \leq 3$, since large values of d are subject to the instabilities typical of high-order polynomial interpolation.

For applications where high fidelity is paramount, and resource constraints are less stringent, high extrapolation orders (e.g. quadratic or cubic) may be preferable. Conversely, for more resource-constrained environments or where moderate improvements in fidelity are sufficient, low extrapolation orders (e.g. linear) might be more suitable.

3. *Vary over number of shots*

In the previous simulations, for each expectation value estimation, we used a fixed budget of $s_{\text{tot}} = 10^6$ shots (total number of circuit executions). We now analyze what happens if we vary s_{tot}. The results are depicted in Figure 8.

Increasing the number of shots induces a reduction of the statistical variance for any estimation strategy (unmitigated, LRE, RE). However, we observe that LRE is much more sensitive to statistical noise and, as a consequence, to the number of shots. For a small number of shots, the statistical variance of LRE is too large to produce a reliable estimation. In this regime, one could try to reduce the overhead of LRE by reducing the number of chunks l or by increasing the gap Δ between scale factors. As the number of shots increases, the performance of LRE stabilizes, yielding more consistent and reliable results.

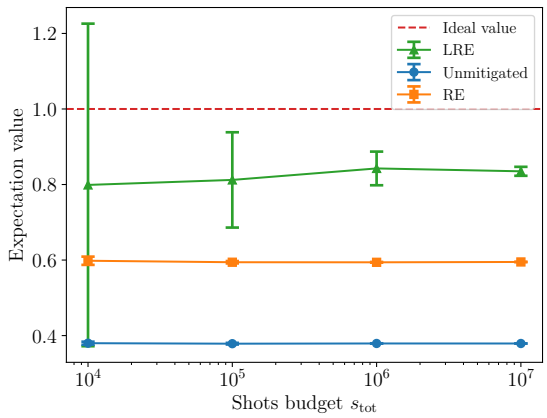

Figure 8. Expectation value of the observable $O = |00\ldots\rangle\langle 00\ldots|$ estimated with different error mitigation strategies for a 6-qubit GHZ-like circuit as defined in Figure 5. Each data point is averaged over 10 trials. For each trial, we use a budget of shots as reported in the horizontal axis. The error bars in each data point illustrate the standard deviation over the different trials.

4. *Vary over the gap between scale factors*

This section delves into the impact of increasing the minimum gap Δ between noise scale factors, as a way of reducing statistical noise at the cost of increasing the estimation bias. The results are presented in Figure 9.

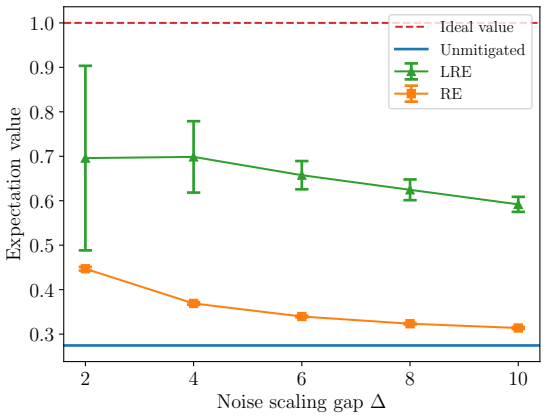

Figure 9. Expectation value estimated with increasing gap Δ between scale factors, for an 8-qubit GHZ-like circuit. We assume a fixed and limited number of shots $s_{\text{tot}} = 10^5$ for any estimation strategy. The error bars are obtained by calculating the standard deviation over the 10 trials.

From our previous analysis of the sampling overhead of LRE, we know that increasing the gap between noise scale factors reduces the sampling cost (see Figure 4). Equivalently, for a fixed number of shots s_{tot}, we expect a reduction of statistical noise for larger values of Δ. This is indeed what we observe for LRE in Figure 9, where error bars get smaller for increasing Δ. A similar reduction is also present for single-variable RE but, error bars are too small to be visible in the plot. In Figure 9 we also see the drawback of using a large gap between noise scale factors: the bias of the associated extrapolation increases due to stronger noise amplification. For practical scenarios, we expect that the net effect of increasing Δ is typically not a convenient strategy when using traditional RE, but it can help when using LRE due to its larger sampling cost.

5. *Vary over the number of circuit chunks*

Finally, we explore the influence of varying the number of circuit chunks l on the estimation accuracy of LRE. As discussed in Section II C, we are not forced to apply LRE to depth-1 layers, but we can apply it to multi-layer chunks of the input circuit. This implies that we are free to split the circuit into an arbitrary number $l = 1, 2, \ldots, l_{\max}$ of chunks, where the upper limit $l_{\max}$ is the total number of depth-1 layers.

In Figure 10, we apply LRE to an 8-qubit GHZ-like

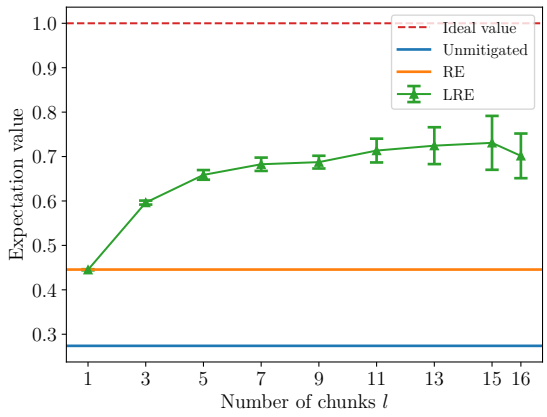

Figure 10. Expectation value estimation via layerwise Richardson extrapolation (LRE) as a function of the number of chunks into which the input circuit is split (as proposed in Section II C). As a benchmark circuit, we use an 8-qubit GHZ-like circuit. The blue line represents the unmitigated expectation value, the orange line depicts the result of applying single-variable RE, and the green triangles show the results after applying LRE. As expected, LRE reduces to RE for $l = 1$. Error bars report the standard deviation over 10 trials.

circuit assuming a splitting of the circuit into a different number of chunks. In practice, for each l, we split the circuit into l chunks of approximately equal depth (up to a rounding error of at most a single layer). Afterward, we apply LRE in the same way as in the previous examples but, instead of associating a noise scale factor to each depth-1 layer, we associate a scale factor to each chunk. For each circuit chunk, we use *local folding* as defined in Equation (7) (also employed for RE).

By construction, LRE reduces to RE for $l = 1$. For larger values of l, we observe a significant reduction of the bias for LRE. We also observe an increase in statistical noise for large l, as expected. The overall interpretation of Figure 10 is that, if we can afford the sampling overhead, it is always convenient to increase l. However, we also expect that for deeper circuits (e.g. $l_{\max} > 100$) the complexity and the sampling cost of applying LRE at the level of single layers may become too large such that applying LRE on a smaller number of multi-layer chunks is a more pragmatic solution.

B. Benchmarking LRE with randomized circuits

In this subsection, we use a different benchmark circuit to test the error mitigation performance of LRE. Instead of the GHZ-like circuit used in the previous examples, we use randomized circuits having the following structure:

$$C = C_{\text{rand}}^{-1} C_{\text{rand}}, \tag{24}$$

where C_{rand} is a random circuit obtained via a randomized application of single-qubit gates (H, X, Y, Z, S, T) and CNOT gates. An instance of C_{rand} is shown in Figure 11. To increase the amount of entanglement during the circuit execution, we assign a high probabilistic weighting to CNOT gates ($p_{\text{CNOT}} = 0.9$), thus ensuring a high density of CNOT gates in the benchmark circuits.

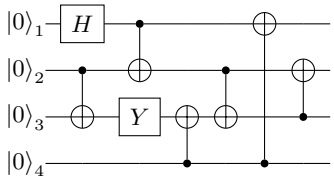

Figure 11. An example of a randomly generated 4-qubit circuit C_{rand}, with high CNOT density. Not that the actual circuit used to benchmark LRE is $C = C_{\text{rand}}^{-1} C_{\text{rand}}$.

Unlike GHZ-like circuits in which the depth is implied by the number of qubits, for the randomized circuits considered in this subsection, we are free to independently vary the number of qubits and the number of layers. This freedom allows us to explore the performance of LRE when varying the number of qubits (at constant depth).

Figure 12 presents a comparative analysis of the error mitigation performance when varying the number of qubits. The plot aggregates results obtained across 10 randomly generated circuits. The results are qualitatively similar to the GHZ-like case reported in Figure 6, in the sense that LRE outperforms both single-variable RE and the trivial unmitigated estimation. However, we also notice an important difference: error bars do not

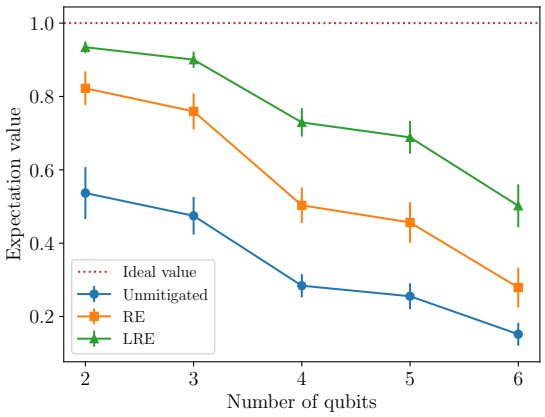

Figure 12. Expectation value of the observable $O = |00\ldots\rangle\langle 00\ldots|$ estimated with different error mitigation strategies for a randomized circuit as defined in Equation (24) having total depth $l_{\max} = 4$. Each data point is averaged over 10 different random instances of the benchmark circuit. Error bars for each data point represent the standard deviation over the 10 random instances. For each circuit, a total budget of $s_{\text{tot}} = 10^6$ shots is used.

grow when increasing the number of qubits. This is a characteristic feature of Richardson extrapolation (both LRE and RE), for which the sampling overhead only depends on the choice of noise scale factors. This implies that the overhead of LRE depends only on the depth l (or number of circuit chunks) but not on the number of qubits. Even if the statistical variance is constant concerning the number of qubits, the bias of all estimation strategies (LRE, RE, unmitigated) gets larger for wider circuits.

IV. DISCUSSION

We introduced *layerwise Richardson extrapolation* (LRE), an error mitigation technique inspired by conventional (single-variable) Richardson extrapolation (RE) [11–13] but generalized to a framework in which the errors acting on different layers of a circuit can be amplified independently. We then presented several numerical experiments in which we compared LRE against conventional RE and direct unmitigated estimation.

Our findings suggest that LRE can be a convenient technique for practical applications since it presents several advantages (low bias, flexible sampling cost, noise-model agnostic). The main limitations of LRE are its statistical uncertainty (higher than RE) and the requirement of running a significant number of different circuits (similar to PEC [11, 12]). We also explored different ways of reducing the sampling cost, such as increasing the gap between scale factors or reducing the number of circuit splittings, that can be useful for controlling the balance between error mitigation bias and sampling cost in large-scale experiments. From a theoretical perspective, LRE also provides a general multivariate formalism in which previous techniques are recovered as special limits. For example, LRE reduces to conventional RE for $l = 1$ and to the noise-scaling version of the NOX protocol [40] for $d = 1$.

The new technique proposed in this work opens up avenues for further research. In our examples, we considered numerical experiments based on a simple amplitude-damping noise model. It would be interesting to numerically investigate other noise models or, even better, test LRE on real hardware. An aspect worth exploring is the design of suitable calibration experiments [25, 36, 37, 40, 45] to estimate the noise levels of different layers or to determine the optimal hyperparameters of the LRE protocol, given a specific backend. For example, one could run calibration experiments to optimize the noise scaling gap Δ, the number of circuit splittings l, and the extrapolation order d.

An interesting analysis would be an experimental comparison between LRE and PEC [11, 12]. In theory, PEC can provide a more tailored error mitigation, since it is a noise-aware technique while LRE is noise-agnostic. In practice, however, it is not obvious what technique is more competitive in a real-world scenario [32]. PEC requires many noise characterization experiments [25, 40] that are known to be complex, costly, and subject to imperfections which have a strong impact on the quality of the final result. LRE is instead simpler and perhaps more robust to imperfections since, by construction, the executed circuits are generated according to a noise-agnostic and deterministic protocol.

Inspired by the PEC protocol, a future direction worth exploring is the probabilistic implementation of LRE. Rather than executing all the M circuits necessary for computing the sum in Equation (19), a Monte Carlo method employing importance sampling could be utilized. This approach would selectively and probabilistically evaluate only a subset of the terms in the full sum and could potentially extend the applicability of LRE to more layers l and to higher orders d.

CODE AVAILABILITY

Software that implements the LRE method along with the code that is used to generate the data and plots in this work is available in [46].

ACKNOWLEDGMENTS

VR acknowledges Nate T. Stemen and Nathan Shammah for insightful discussions as well as William J. Zeng for suggesting the idea of applying layerwise folding as a tool for error mitigation. This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing under Award Numbers DE-SC0020266 and DE-SC0020316 as well as by IBM under Sponsored Research Agreement No. W1975810.

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[2] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, *et al.*, "Quantum computational advantage using photons," *Science*, vol. 370, no. 6523, pp. 1460–1463, 2020.

[3] C. Neill, P. Roushan, K. Kechedzhi, S. Boixo, S. V. Isakov, V. Smelyanskiy, A. Megrant, B. Chiaro, A. Dunsworth, K. Arya, *et al.*, "A blueprint for demonstrating quantum supremacy with superconducting qubits," *Science*, vol. 360, no. 6385, pp. 195–199,

2018.

[4] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins, *et al.*, "Quantum computational advantage with a programmable photonic processor," *Nature*, vol. 606, no. 7912, pp. 75–81, 2022.

[5] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, *et al.*, "Strong quantum computational advantage using a superconducting quantum processor," *Physical review letters*, vol. 127, no. 18, p. 180501, 2021.

[6] S. Ebadi, T. T. Wang, H. Levine, A. Keesling, G. Semeghini, A. Omran, D. Bluvstein, R. Samajdar, H. Pichler, W. W. Ho, *et al.*, "Quantum phases of matter on a 256-atom programmable quantum simulator," *Nature*, vol. 595, no. 7866, pp. 227–232, 2021.

[7] A. J. Daley, I. Bloch, C. Kokail, S. Flannigan, N. Pearson, M. Troyer, and P. Zoller, "Practical quantum advantage in quantum simulation," *Nature*, vol. 607, no. 7920, pp. 667–676, 2022.

[8] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[9] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, "Quantum error mitigation," *arXiv preprint arXiv:2210.00921*, 2022.

[10] Y. Li and S. C. Benjamin, "Efficient variational quantum simulator incorporating active error minimization," *Physical Review X*, vol. 7, no. 2, p. 021050, 2017.

[11] K. Temme, S. Bravyi, and J. M. Gambetta, "Error mitigation for short-depth quantum circuits," *Physical review letters*, vol. 119, no. 18, p. 180509, 2017.

[12] S. Endo, S. C. Benjamin, and Y. Li, "Practical quantum error mitigation for near-future applications," *Physical Review X*, vol. 8, no. 3, p. 031027, 2018.

[13] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, "Error mitigation extends the computational reach of a noisy quantum processor," *Nature*, vol. 567, no. 7749, pp. 491–495, 2019.

[14] A. Strikis, D. Qin, Y. Chen, S. C. Benjamin, and Y. Li, "Learning-based quantum error mitigation," *PRX Quantum*, vol. 2, no. 4, p. 040330, 2021.

[15] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng, "Digital zero noise extrapolation for quantum error mitigation," in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 306–316, IEEE, 2020.

[16] R. LaRose, A. Mari, S. Kaiser, P. J. Karalekas, A. A. Alves, P. Czarnik, M. El Mandouh, M. H. Gordon, Y. Hindy, A. Robertson, *et al.*, "Mitiq: A software package for error mitigation on noisy quantum computers," *Quantum*, vol. 6, p. 774, 2022.

[17] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, "Hybrid quantum-classical algorithms and quantum error mitigation," *Journal of the Physical Society of Japan*, vol. 90, no. 3, p. 032001, 2021.

[18] 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," *Nature Physics*, pp. 1–8, 2023.

[19] B. Koczor, "Exponential error suppression for near-term quantum devices," *Physical Review X*, vol. 11, no. 3, p. 031057, 2021.

[20] A. He, B. Nachman, W. A. de Jong, and C. W. Bauer, "Zero-noise extrapolation for quantum-gate error mitigation with identity insertions," *Physical Review A*, vol. 102, no. 1, p. 012426, 2020.

[21] W. J. Huggins, S. McArdle, T. E. O'Brien, J. Lee, N. C. Rubin, S. Boixo, K. B. Whaley, R. Babbush, and J. R. McClean, "Virtual distillation for quantum error mitigation," *Physical Review X*, vol. 11, no. 4, p. 041036, 2021.

[22] V. R. Pascuzzi, A. He, C. W. Bauer, W. A. De Jong, and B. Nachman, "Computationally efficient zero-noise extrapolation for quantum-gate-error mitigation," *Physical Review A*, vol. 105, no. 4, p. 042406, 2022.

[23] C. Song, J. Cui, H. Wang, J. Hao, H. Feng, and Y. Li, "Quantum computation with universal error mitigation on a superconducting quantum processor," *Science advances*, vol. 5, no. 9, p. eaaw5686, 2019.

[24] S. Zhang, Y. Lu, K. Zhang, W. Chen, Y. Li, J.-N. Zhang, and K. Kim, "Error-mitigated quantum gates exceeding physical fidelities in a trapped-ion system," *Nature communications*, vol. 11, no. 1, p. 587, 2020.

[25] E. Van Den Berg, Z. K. Minev, A. Kandala, and K. Temme, "Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors," *Nature Physics*, pp. 1–6, 2023.

[26] L. F. Santos and L. Viola, "Dynamical control of qubit coherence: Random versus deterministic schemes," *Physical Review A*, vol. 72, no. 6, p. 062303, 2005.

[27] L. Viola and E. Knill, "Random decoupling schemes for quantum dynamical control and error suppression," *Physical review letters*, vol. 94, no. 6, p. 060502, 2005.

[28] B. Pokharel, N. Anand, B. Fortman, and D. A. Lidar, "Demonstration of fidelity improvement using dynamical decoupling with superconducting qubits," *Physical review letters*, vol. 121, no. 22, p. 220502, 2018.

[29] P. Sekatski, M. Skotiniotis, and W. Dür, "Dynamical decoupling leads to improved scaling in noisy quantum metrology," *New Journal of Physics*, vol. 18, no. 7, p. 073034, 2016.

[30] P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, "Error mitigation with Clifford quantum-circuit data," *Quantum*, vol. 5, p. 592, 2021.

[31] A. Lowe, M. H. Gordon, P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, "Unified approach to data-driven quantum error mitigation," *Physical Review Research*, vol. 3, no. 3, p. 033098, 2021.

[32] V. Russo, A. Mari, N. Shammah, R. LaRose, and W. J. Zeng, "Testing platform-independent quantum error mitigation on noisy quantum computers," *IEEE Transactions on Quantum Engineering*, 2023.

[33] C. Cirstoiu, S. Dilkes, D. Mills, S. Sivarajah, and R. Duncan, "Volumetric benchmarking of error mitigation with Qermit," *Quantum*, vol. 7, p. 1059, 2023.

[34] R. LaRose, A. Mari, V. Russo, D. Strano, and W. J. Zeng, "Error mitigation increases the effective quantum volume of quantum computers," *arXiv preprint arXiv:2203.05489*, 2022.

[35] A. He, B. Nachman, W. A. de Jong, and C. W. Bauer, "Resource efficient zero noise extrapolation with identity insertions," *arXiv preprint arXiv:2003.04941*, 2020.

[36] F. A. Calderon-Vargas, T. Proctor, K. Rudinger, and M. Sarovar, "Quantum circuit debugging and sensitivity analysis via local inversions," *Quantum*, vol. 7, p. 921, 2023.

[37] T. Patel, D. Silver, and D. Tiwari, "Charter: Identifying the most-critical gate operations in quantum circuits via amplified gate reversibility," in *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16, IEEE, 2022.

[38] K. Saniee, "A simple expression for multivariate Lagrange interpolation," *SIAM undergraduate research online*, vol. 1, no. 1, pp. 1–9, 2008.

[39] P. J. Olver, "On multivariate interpolation," *Studies in Applied Mathematics*, vol. 116, no. 2, pp. 201–240, 2006.

[40] S. Ferracin, A. Hashim, J.-L. Ville, R. Naik, A. Carignan-Dugas, H. Qassim, A. Morvan, D. I. Santiago, I. Siddiqi, and J. J. Wallman, "Efficiently improving the performance of noisy quantum computers," *arXiv preprint arXiv:2201.10672*, 2022.

[41] A. Mari, N. Shammah, and W. J. Zeng, "Extending quantum probabilistic error cancellation by noise scaling," *Physical Review A*, vol. 104, no. 5, p. 052607, 2021.

[42] M. Otten and S. K. Gray, "Recovering noise-free quantum observables," *Physical Review A*, vol. 99, no. 1, p. 012338, 2019.

[43] D. Cox, J. Little, and D. OShea, *Ideals, varieties, and algorithms: An introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

[44] R. Takagi, S. Endo, S. Minagawa, and M. Gu, "Fundamental limits of quantum error mitigation," *npj Quantum Information*, vol. 8, no. 1, p. 114, 2022.

[45] L. Hour, S. Heng, M. Go, and Y. Han, "Improving zero-noise extrapolation for quantum-gate error mitigation using a noise-aware folding method," *arXiv preprint arXiv:2401.12495*, 2024.

[46] UnitaryFund, "UnitaryFund Research." https://github.com/unitaryfund/research/, Feb. 2024.

[47] Qiskit contributors, "Qiskit: An open-source framework for quantum computing," 2023.

[48] M. Gasca and T. Sauer, "Polynomial interpolation in several variables," *Advances in Computational Mathematics*, vol. 12, pp. 377–410, 2000.

V. APPENDIX

A. Noise model for experiments

For the experiments in Section III, we consider a noise model characterized by amplitude damping errors. Let the probability of amplitude damping error for a single qubit and two-qubit gate be denoted as p_1 and p_2 respectively, with $p_1 = 0.04$ and $p_2 = 0.08$. The single-qubit amplitude damping channel is represented as

$$\mathcal{E}_1(\rho) = E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger \tag{25}$$

where,

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-p_1} \end{bmatrix} \quad \text{and} \quad E_1 = \begin{bmatrix} 0 & \sqrt{p_1} \\ 0 & 0 \end{bmatrix}. \tag{26}$$

This channel is added to all single-qubit gates. For the two-qubit CNOT gate, we apply the tensor product of two single-qubit channels,

$$\mathcal{E}_2(\rho) = \sum_{i \in \{0,1\}} \sum_{j \in \{0,1\}} E_i \otimes E_j \, \rho \, E_i^\dagger \otimes E_j^\dagger, \tag{27}$$

where we replace p_1 in Equation (26) with $p_2 = 0.08$. We use the Qiskit Aer simulator [47] to simulate circuits with the above noise model.

B. Multivariate Lagrange interpolation in LRE

Single-variable Lagrange interpolation constructs a single-variable polynomial to fit a set of N points in $\mathbb{R}^2$ [48]. In the case of *multivariate Lagrange interpolation*, this approach is extended to handle the multivariate polynomial interpolation of points in higher-dimensional spaces. Here, for a set of N points of a polynomial with l variables, the interpolation is conducted in $\mathbb{R}^{l+1}$ [38, 39]. In this appendix we adapt the mathematical formalism of Lagrange interpolation of [38] to the specific notation of the LRE framework introduced in Section II B.

We aim to find the interpolating l-variable polynomial passing through a set of N points representing the noise-scaled expectation values of an observable. Each of these points corresponds to a circuit execution under a specific noise scaling, captured by a vector $\boldsymbol{\lambda}$ containing l real scale factors, corresponding to the amount of noise scaling applied to the l-th layer of the circuit. Given the N measured points, we define the set of scale factor vectors as $\Lambda = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_N\}$ and the array of the associated expectation values as

$$\mathbf{z} = (\langle O(\boldsymbol{\lambda}_1)\rangle, \langle O(\boldsymbol{\lambda}_2)\rangle, \ldots, \langle O(\boldsymbol{\lambda}_N)\rangle)^\intercal. \tag{28}$$

The most general l-variable polynomial of degree d can be written as

$$P(\boldsymbol{\lambda}) = \sum_{j=1}^{M} c_j \mathcal{M}_j(\boldsymbol{\lambda}, d), \tag{29}$$

where $\{c_j\}$ are real coefficients and $\{\mathcal{M}_j(\boldsymbol{\lambda}, d) : j = 1, 2, \ldots, M\}$ is the set of all l-variable monomials of degree at most d. The number of monomials is given by $M = \binom{d+l}{d}$ and is therefore fixed by l and d. The interpolation problem corresponds to determining the M unknown coefficients $\{c_j\}$ such that the polynomial passes through the measured points, i.e.:

$$P(\boldsymbol{\lambda}_i) = \langle O(\boldsymbol{\lambda}_i)\rangle = \mathbf{z}_i, \quad \forall \boldsymbol{\lambda}_i \in \Lambda. \tag{30}$$

Define the following *sample matrix* which contains the values of all monomials evaluated at each scale factor vector in Λ:

$$\mathbf{A}(\Lambda, d) = \begin{bmatrix} \mathcal{M}_1(\boldsymbol{\lambda}_1, d) & \mathcal{M}_2(\boldsymbol{\lambda}_1, d) & \cdots & \mathcal{M}_M(\boldsymbol{\lambda}_1, d) \\ \mathcal{M}_1(\boldsymbol{\lambda}_2, d) & \mathcal{M}_2(\boldsymbol{\lambda}_2, d) & \cdots & \mathcal{M}_M(\boldsymbol{\lambda}_2, d) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{M}_1(\boldsymbol{\lambda}_N, d) & \mathcal{M}_2(\boldsymbol{\lambda}_N, d) & \cdots & \mathcal{M}_M(\boldsymbol{\lambda}_N, d) \end{bmatrix}. \tag{31}$$

If we cast the coefficients of the polynomial in a vector $\mathbf{c} = (c_1, c_2, \ldots, c_M)^\intercal$, the interpolation problem can be expressed as the following linear system:

$$\mathbf{A}\mathbf{c} = \mathbf{z}. \tag{32}$$

To have a unique solution, we assume $N = M$ and $\det(\mathbf{A}) \neq 0$. In practice, given l and d, this is a constraint on the number and the values of the scale factor vectors in the set Λ that is straightforward to check and satisfy.

One way of determining the interpolating polynomial would be to solve for $\mathbf{c}$ and to replace the solution into Equation (29). There is however an alternative way, which does not require the explicit computation of $\mathbf{c}$ and is given by the following Lagrange interpolation formula [38]:

$$P(\boldsymbol{\lambda}) = \sum_{i=1}^{M} \langle O(\boldsymbol{\lambda}_i)\rangle \frac{\det\left(\mathbf{M}_i(\boldsymbol{\lambda})\right)}{\det\left(\mathbf{A}\right)}, \tag{33}$$

where $\mathbf{M}_i(\boldsymbol{\lambda})$ is the matrix obtained by substituting the i-th row of the sample matrix $\mathbf{A}$ with the same row of monomials but evaluated on the generic polynomial variable $\boldsymbol{\lambda}$ (instead of $\boldsymbol{\lambda}_i \in \Lambda$), for example:

$$\mathbf{M}_2(\boldsymbol{\lambda}) = \begin{bmatrix} \mathcal{M}_1(\boldsymbol{\lambda}_1, d) & \mathcal{M}_2(\boldsymbol{\lambda}_1, d) & \cdots & \mathcal{M}_M(\boldsymbol{\lambda}_1, d) \\ \mathcal{M}_1(\boldsymbol{\lambda}, d) & \mathcal{M}_2(\boldsymbol{\lambda}, d) & \cdots & \mathcal{M}_M(\boldsymbol{\lambda}, d) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{M}_1(\boldsymbol{\lambda}_N, d) & \mathcal{M}_2(\boldsymbol{\lambda}_N, d) & \cdots & \mathcal{M}_M(\boldsymbol{\lambda}_N, d) \end{bmatrix}. \tag{34}$$

By construction, the right-hand side of Equation (33) is a polynomial in the variable $\boldsymbol{\lambda}$ of degree at most d. Moreover, it is easy to check that it also interpolates all points since, if we evaluate the expression at a specific $\boldsymbol{\lambda}_j \in \Lambda$, we have

$$P(\boldsymbol{\lambda}_j) = \sum_{i=1}^{M} \langle O(\boldsymbol{\lambda}_i)\rangle \frac{\det\left(\mathbf{M}_i(\boldsymbol{\lambda}_j)\right)}{\det\left(\mathbf{A}\right)} = \langle O(\boldsymbol{\lambda}_j)\rangle \frac{\det\left(\mathbf{M}_j(\boldsymbol{\lambda}_j)\right)}{\det\left(\mathbf{A}\right)} = \langle O(\boldsymbol{\lambda}_j)\rangle \frac{\det\left(\mathbf{A}\right)}{\det\left(\mathbf{A}\right)} = \langle O(\boldsymbol{\lambda}_j)\rangle, \tag{35}$$

where we used that, for $i \neq j$, $\det\left(\mathbf{M}_i(\boldsymbol{\lambda}_j)\right) = 0$ since the i-th row and the j-th row are equal.

Evaluating Equation (33) at the zero-noise limit (denoted as $\boldsymbol{\lambda} = \mathbf{0}$), we get:

$$O_{\mathrm{LRE}} = P(\mathbf{0}) = \sum_{i=1}^{M} \langle O(\boldsymbol{\lambda}_i)\rangle \frac{\det\left(\mathbf{M}_i(\mathbf{0})\right)}{\det\left(\mathbf{A}\right)}. \tag{36}$$

The matrix $\mathbf{M}_i(\mathbf{0})$ can be obtained from the sample matrix $\mathbf{A}$ after replacing the i-th row by the array $\mathbf{e}_1 = (1, 0, \ldots, 0)^\intercal$, since all monomials are zero at $\boldsymbol{\lambda} = \mathbf{0}$, with the exception of the constant one $\mathcal{M}_1(\mathbf{0}, d) = \mathcal{M}_1(\boldsymbol{\lambda}, d) = 1$. Here, we implicitly assumed that monomials are ordered with increasing degree. Otherwise, the element 1 in the vector $\mathbf{e}_1$ should be shifted to the position associated with the zero-order monomial. Equation (36) corresponds to Equations (19) and (20) of the main text.