Digital zero noise extrapolation for quantum error mitigation

Tudor Giurgica-Tiron*, Yousef Hindy*, Ryan LaRose†§, Andrea Mari†¶, William J. Zeng*†‡
*Stanford University, Palo Alto, CA
†Unitary Fund
‡Goldman, Sachs & Co, New York, NY
§Michigan State University, East Lansing, MI
¶Xanadu, Toronto, Ontario, Canada

***Abstract*—Zero-noise extrapolation (ZNE) is an increasingly popular technique for mitigating errors in noisy quantum computations without using additional quantum resources. We review the fundamentals of ZNE and propose several improvements to noise scaling and extrapolation, the two key components in the technique. We introduce unitary folding and parameterized noise scaling. These are digital noise scaling frameworks, i.e. one can apply them using only gate-level access common to most quantum instruction sets. We also study different extrapolation methods, including a new adaptive protocol that uses a statistical inference framework. Benchmarks of our techniques show error reductions of 18X to 24X over non-mitigated circuits and demonstrate ZNE's effectiveness at larger qubit numbers than have been tested previously. In addition to presenting new results, this work is a self-contained introduction to the practical use of ZNE by quantum programmers.**

***Index Terms*—quantum computing**

I. Introduction

As quantum hardware becomes available in the noisy intermediate-scale quantum computing (NISQ) era [1], it is inevitable that today's quantum programmer must deal with errors. In the long run, fault-tolerance and quantum error-correction have the potential to arbitrarily reduce logical errors [2]–[4]. However, a scalable logical qubit has yet to be demonstrated. Thus the savvy quantum programmer should make use of *error-mitigating* techniques that give practical benefits, even if they do not arbitrarily suppress errors in the asymptotic limit. In the NISQ era, every constant factor counts.

There are many examples of error-mitigating techniques, including probabilistic error cancellation [5], [6], randomized compiling [7], Pauli-frame randomization [8], dynamical decoupling [9]–[12], quantum optimal control [13], [14], etc. In this work, we focus on the specific error-mitigating technique known as *zero-noise extrapolation*.

Zero-noise extrapolation (ZNE) was introduced concurrently in [5] and [15]. In ZNE, a quantum program is altered to run at different effective levels of processor noise. The result of the computation is then extrapolated to an estimated value at a noiseless level. More formally, one can parameterize the noise-level of a quantum system with a dimensionless scale factor λ.

U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing under Award Number DE-SC0020266

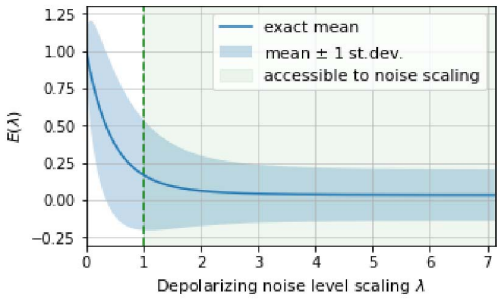

Fig. 1: An example of the change of an expectation value, $E(\lambda)$, with the underlying scaling λ of the depolarizing noise level. Here the simulated base noise value is 5% (marked by the green dashed vertical line). ZNE increases that noise and back extrapolates to the $\lambda = 0$ expectation value. In this example, an accurate extrapolation should be non-linear and take advantage of a known asymptotic behavior.

For $\lambda = 0$ the noise is removed, while for $\lambda = 1$ the true noise-level of the physical hardware is matched. For example, λ could be a multiplicative factor that scales the dissipative terms of a master equation [5]. More generally, λ could represent a re-scaling of any physical quantity which introduces some noise in the quantum computation: the calibration uncertainty of variational parameters, the temperature of the quantum processor, etc.

For a given quantum program, we can measure an arbitrary expectation value $E(\lambda)$. By construction, $E(1)$ represents the expectation value evaluated with the natural noise of the hardware, whereas $E(0)$ denotes the noiseless observable which, despite being not directly measurable, we would like to estimate.

To implement ZNE, one needs a direct or indirect way to scale the quantum computation's noise level to values of λ larger than one. With such a method, ZNE can be implemented in two main steps:

1) **Noise-scaling:** Measure $E(\lambda)$ at m different values of $\lambda \geq 1$.
2) **Extrapolation:** Infer $E(0)$ from the m expectation values measured in previous step.

978-1-7281-8969-7/20/$31.00 ©2020 IEEE
DOI 10.1109/QCE49297.2020.00045

Figure 1 shows an example *noise curve* given by scaling depolarizing noise for a randomized benchmarking circuit.

In this work, we introduce improvements to both noise-scaling and extrapolation methods for quantum error mitigation. In Section II-A we introduce unitary folding, a framework for digital noise scaling of generic gate noise. In Section II-B we introduce a general method for ZNE tailored for a specific kind of errors: calibration noise. We then move to the extrapolation step of ZNE, which we characterize as an inference problem. We study non-adaptive (Section III) extrapolation methods and introduce adaptive (Section IV) extrapolation to improve performance and reduce resource overhead for ZNE.

II. Noise Scaling Methods

In [5] and [16] a time-scaling approach implements the scaling of effective noise on the back-end quantum processor. Control pulses for each gate are re-calibrated to execute the same unitary evolution but applied over a longer amount of time. This effectively scales up the noise. While successfully used to suppress errors in single and two-qubit quantum programs on a superconducting quantum processor [16], time-scaling has some disadvantages:

- It requires programmer access to low-level physical-control parameters. This level of access is not available on all quantum hardware and breaks the gate model abstraction.
- Control pulses must be re-calibrated for each time duration and error-scaling. This calibration can be resource intensive.

Instead, we study alternative approaches that require only a gate-level access to the system. Rather than increasing the time duration of each gate, we increase the total number of gates or, similarly, the circuit depth. This procedure is similar to what is usually done by a *quantum compiler* but with the opposite goal: instead of optimizing a circuit to reduce its depth or its gate count, we are interested in "de-optimizing" to increase the effect of noise and decoherence. We use the term *digital* to describe noise-scaling techniques that manipulate just the quantum program at the instruction set layer. Their advantage is that they can be used with the gate model access that is common to most quantum assembly languages [17]–[19]. Low level access to pulse shaping and detailed physical knowledge of quantum processor physics is no longer required. Our digital framework incorporates and generalizes some recent related work [20], [21].

A. Unitary Folding

We describe two methods–circuit folding and gate folding–for scaling the effective noise of a quantum computation based on *unitary folding*, i.e., replacing a unitary circuit (or gate) U by:

$$U \to U(U^\dagger U)^n, \tag{1}$$

where n is a positive integer. In an ideal circuit, since $U^\dagger U$ is equal to the identity, this folding operation has no logical effect. However, on a real quantum computer, we expect that the noise increases since the number of physical operations scales by a factor of $1 + 2n$. This effect is clearly visible in the quantum computing experiment reported in Figure 6.

A similar trick was used in Ref. [20], [21], where noise was artificially increased by inserting pairs of CNOT gates into quantum circuits. In our framework, U can represent the full input circuit or, alternately, some local gates which are inserted with different strategies.

1) Circuit folding

Assume that the circuit is composed of d unitary layers:

$$U = L_d ... L_2 L_1, \tag{2}$$

where d represents the *depth* of the circuit and each block L_j can either represent a single layer of operations or just a single gate.

In circuit folding, the substitution rule in Eq. (1) is applied globally, i.e., to the entire circuit. This scales the effective depth by odd integers. In order to have a more fine-grained resolution of the scaling factor, we can also allow for a final folding applied to a subset of the circuit corresponding to its last s layers. The general *circuit folding* replacement rule is therefore:

$$U \to U(U^\dagger U)^n L_d^\dagger L_{d-1}^\dagger \dots L_s^\dagger L_s \dots L_{d-1} L_d. \tag{3}$$

The total number of layers of the new circuit is $d(2n+1)+2s$. This means that we can stretch the depth of a circuit up to a scale resolution of $2/d$, i.e., we can apply the scaling $d \to \lambda d$, where:

$$\lambda = 1 + \frac{2k}{d}, \quad k = 1, 2, 3, \dots. \tag{4}$$

Conversely, for every real λ, one can apply the following procedure:

1) Determine the closest integer k to the real quantity $d(\lambda - 1)/2$.
2) Perform an integer division of k by d. The quotient corresponds to n, while the reminder to s.
3) Apply n integer foldings and a final partial folding as described in Eq. (3).

From a physical point of view, the circuit folding method corresponds to repeatedly driving the Hamiltonian of the qubits forwards and backwards in time, such that the ideal unitary part of the dynamics is not changed while the non-unitary effect of the noise is amplified.

2) Gate (or Layer) folding

Instead of globally folding a quantum circuit, appending the folds at the end, one could fold a subset of individual gates (or layers) in place. Let us consider the circuit decomposition of Eq. (2) where we can assume that each unitary operator L_j represents just a single gate applied to one or two qubits of the system or, alternatively, each L_j could be a layer of several gates.

If we apply the replacement rule given in Eq. (1) to each gate (or layer) L_j of the circuit, it is clear that the initial number of gates (layers) d is scaled by an odd integer $1 + 2n$. Similarly to the case of circuit folding, we can add a

TABLE I: Different methods for implementing gate (or layer) folding

Method	Subset of indices to fold
From left	$S = \{1, 2, \ldots, s\}$
From right	$S = \{d, d-1, \ldots, d-s+1\}$
At random	$S = s$ different indices randomly sampled without replacement from $\{1, 2, \ldots, d\}$.

final partial folding operation to get a scaling factor which is more fine grained. In order to achieve such "partial" folding, let us define an arbitrary subset S of the full set of indices $\{1, 2, \ldots d\}$, such that its number of elements is a given integer $s = |S|$. In this setting, we can define the following *gate (layer) folding* rule:

$$\forall j \in \{1, 2, \ldots d\}, \quad L_j \rightarrow \begin{cases} L_j (L_j^\dagger L_j)^n & \text{if} j \notin S, \\ \\ L_j (L_j^\dagger L_j)^{n+1} & \text{if} j \in S. \end{cases} \tag{5}$$

Depending on how we chose the elements of the subset S, different noise channels will be added at different positions along the circuit and so we can have different results. The optimal choice may depend on the particular circuit and noise model. We focus on three different ways of selecting the subset of gates (layers) to be folded: *from left*, *from right* and *at random*. Depending on the method, the prescription for selecting the subset S of indices is reported in Table I.

It is easy to check that the number of gates (or layers), obtained after the application of the gate folding rule given in Eq. (5) is $d(2n+1) + 2s$. This is exactly the same number obtained after the application of the global circuit-folding rule given in Eq. (3). As a consequence, the number of gates (layers) is still stretched by a factor λ, i.e., $d \rightarrow \lambda d$, where λ can take the specific values reported in Eq. (4). Moreover, if we are given an arbitrary λ and we want to determine the values of n and s, we can simply apply the same procedure that was given in the case of circuit-folding.

While preparing this manuscript we became aware of [20] whose technique is similar to our gate folding (at random). The main difference is that [20] focuses mainly on CNOT gates and uses random sampling with replacement, in our case any gate (or layer) can be folded and the sampling is performed without replacement. The rationale of this choice is to sample in a more uniform way the input circuit, and to converge smoothly to the odd integer values of $\lambda = 1 + 2n$ where all the input gates are folded exactly n times.

3) Advantages and limitations of unitary folding

The main advantage of the unitary folding approach is that is is digital, i.e., noise is scaled using a high level of abstraction from the physical hardware. Moreover, it can be applied without knowing the details of the underlying noise-model. It is natural to ask: how justified is this approach physically? Does unitary folding actually correspond to an effective scaling of the physical noise of the hardware?

For example, unitary folding may fail to amplify systematic and coherent errors since applying the inverse of a gate will usually *undo* such errors instead of increasing them. It is also clear that unitary folding is not appropriate to scale state preparation and measurement (SPAM) noise, since this noise is independent of the circuit depth. Instead, we expect that unitary folding can be used for scaling incoherent noise models which are associated both to the application of individual gates and/or to the time-length of the overall computation. The more we increase the depth of the circuit, the more such kinds of noise are usually amplified. In this work this intuition is confirmed by numerical and experimental examples in which unitary folding is successfully used for implementing ZNE (see Figures 2, 3, 4 and 6).

The effect of unitary folding can be analytically derived when the noise-model for each gate L_j is a global depolarizing channel with a gate-dependent parameter $p_j \in [0, 1]$, acting as:

$$\rho \xrightarrow{\text{noisy gate}} p_j L_j \rho L_j^\dagger + (1 - p_j)\mathbb{I}/D, \tag{6}$$

where D is the dimension of the Hilbert space associated to all the qubits of the circuit. Since the depolarizing channel commutes with unitary operations, we can postpone the noise channels of all the gates until the end of the full circuit U, resulting into a single final depolarizing channel:

$$\rho \xrightarrow{\text{noisy circuit}} pU\rho U^\dagger + (1 - p)\mathbb{I}/D, \tag{7}$$

where $p = \Pi_j p_j$ is the product of all the gate-dependent noise parameters p_j. This simple commutation property does not hold for local depolarizing noise, unless we are dealing with singe-qubit circuits.

Consider what happens if we apply unitary folding with a scale factor $\lambda = 1 + 2n$ (odd positive integer). For both the circuit folding and the gate folding methods, defined in Eq. (3) and (5) respectively, the final result is exactly equivalent to an exponential scaling of all the depolarizing parameters of each gate $p_j \rightarrow p_j^\lambda$ or, equivalently, to the global operation:

$$\rho \xrightarrow{\text{noise + unitary folding}} p^\lambda U\rho U^\dagger + (1 - p^\lambda)\mathbb{I}/D. \tag{8}$$

This implies that unitary folding is equivalent to an exponential parameterization of the noise level p, and so any expectation value is also scaled according to an exponential ansatz:

$$E(\lambda) = a + bp^\lambda, \tag{9}$$

which we can fit and extrapolate according to the methods discussed in the Sections III and IV.

Equations (8) and (9) are valid only for depolarizing noise and for odd scaling factors λ. For gate-independent depolarizing noise, the global parameter p is a function of the total number of gates only. This means that all the folding methods (circuit, from left, from right and at random) become equivalent, and induce the exponential scalings of Eqs. (8) and (9)) for all values of λ.

4) Numerical Results

We executed density matrix simulations using unitary folding for zero-noise extrapolation. Broadly these results show that unitary folding is effective in a variety of situations. Furthermore, we benchmark on both random circuits and a variational algorithm at 6 and more qubits. This extends

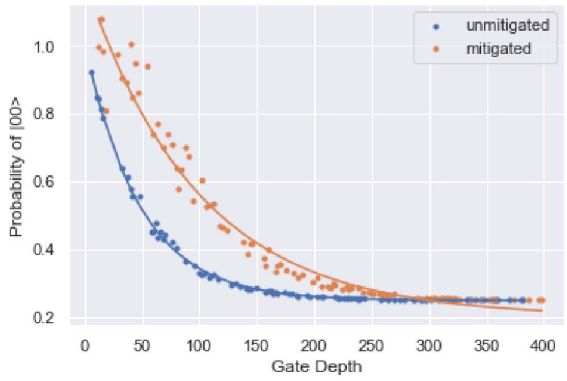

Fig. 2: A comparison of two qubit randomized benchmarking with and without error-mitigation. Data is taken by density matrix simulation with a 1% depolarizing noise model. The unmitigated simulation results in a randomized benchmarking decay of 97.9%. Mitigation is applied using circuit folding and an order-2 polynomial extrapolation at $\lambda = 1, 1.5, 2.0$. With mitigation the randomized benchmarking decay improves to 99.0%. Since we do not impose any constraint on the domain of the extrapolated results, some of the mitigated expectation values are slightly beyond the physical upper limit of 1. This is an expected effect of the noise introduced by the extrapolation fit. If necessary, one could enforce the result to be physical by using a more advanced Bayesian estimator.

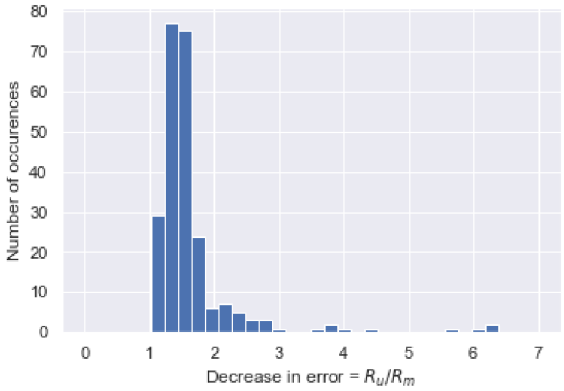

Fig. 3: A comparison of improvements from ZNE (using quadratic extrapolation with folding from left) averaged across all output bitstrings from 250 random six-qubit circuits. Results are from exact density matrix simulations with a base of 1% depolarizing noise. The horizontal axis shows a ratio of L_2 distances from the noiseless probability distribution and the vertical axis shows the frequency of obtaining this result. ZNE improves on the noisy result by factors of 1-7X. The average mitigated error is 0.075 ± 0.035, while the unmitigated errors average 0.114 ± 0.050. Each circuit has 40 moments with single-qubit gates sampled randomly from $\{H, X, Y, Z, S, T\}$ and two-qubit gates sampled randomly from $\{\text{iSWAP}, \text{CZ}\}$ with arbitrary connectivity.

previous work that focuses on the single and two qubit cases [5], [6], [15], [16]. Figure 2 shows a simulated two qubit randomized benchmarking experiment under 1% depolarizing noise with and without error-mitigation. Noise was scaled using circuit folding as described in Section II-A1.

Figure 3 shows the distribution of noise reduction by ZNE with circuit folding on randomly generated six qubit circuits. Let E_m be the mitigated expectation value of a circuit after zero-noise extrapolation. Then $R_m = |E_m - E(0)|$ is the absolute value of the error in the mitigated expectation and $R_u = |E(1) - E(0)|$ is the absolute value of the error of the unmitigated circuit. The improvement from ZNE is quantified as R_u/R_m.

Table II (see Section III) provides a comparison different combinations of folding and extrapolation techniques on a set of randomized benchmarking circuits.

Figure 4 shows the performance of unitary folding ZNE on a variational algorithm. Using exact density matrix simulation we study the percentage closer to optimal achieved by the quantum approximation optimization algorithm [22] on random instances of MAXCUT.

B. Parameter Noise Scaling

While unitary folding applies to general classes of noise models, it is reasonable to ask if we can exploit the specific structure of particular noise models. We give an example of this approach by mitigating errors in the stochastic calibration of parametric quantum gates. The error model that we consider is a generalized form of the "pulse-area" error, which describes what happens when the physical pulses that generate a particular gate in a quantum processor are slightly miscalibrated [23]–[27]. This noise could be due to fluctuations in control electronics, or uncertainty about underlying physical parameters (such as qubit frequencies) in available hardware. Furthermore, our model applies also to variational quantum circuits, in which the parametric dependence of the gates is critically accessed by quantum programmers.

In order to apply ZNE in this setting, we need a method for scaling this particular kind of noise source. Instead of changing the structure of the quantum circuit, as in the unitary folding method, we directly inject classical noise into the control parameters. This artificial noise can increase the native noise of the hardware to larger levels, such that ZNE becomes applicable. We call this approach *parameter noise scaling*.

1) Parameter Noise Scaling Theory

We assume that a quantum gate is parameterized by l classical control parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots \theta_l)$, such that

$$G(\boldsymbol{\theta}) = \exp\left(-i \sum_{j=1}^{l} \theta_j H_j\right), \tag{10}$$

where $H_1, H_2, \ldots H_l$ are Hermitian operators. In practice, the parameters $\boldsymbol{\theta}$ can represent the classical controls that the quantum processor needs to tune in order to implement a particular

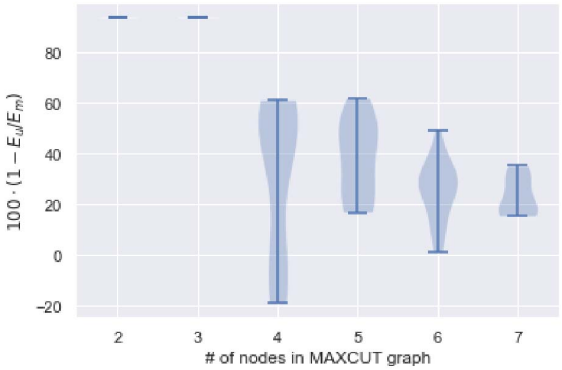

Fig. 4: Percent closer to optimal on random MAXCUT executions. 14 Erdos-Renyi random graphs were generated at each number n. Each random graph has n nodes and n edges. QAOA was then run (with $p = 2$ QAOA steps) and optimized using Nelder-Mead with and without error mitigation. Results are from exact density matrix simulations with a base of 2% depolarizing noise. For the mitigated case, we used zero noise extrapolation with global unitary folding for scaling and linear extrapolation at noise scalings of 1, 1.5 and 2. The y axis shows the percent closer to the optimal solution that was gained by ZNE. Here E_u is the absolute error in the unmitigated expectation and E_m is the absolute error in the mitigated expectation. The violin plot shows the distribution of percentage improvements over the 14 sampled instances. Variance is zero for 2 and 3 nodes graphs as there is only a single valid graph with n nodes and edges for $n = 2, 3$.

gate. Alternately, $G(\boldsymbol{\theta})$ could also model a variational gate which can be programmed by the user.

In both cases, it is reasonable to assume that the control parameters can be applied only up to some finite precision, i.e., that what is actually implemented on the physical system is the gate $G(\boldsymbol{\theta'})$, where

$$\theta'_j = \theta_j + \hat{\epsilon}_j, \tag{11}$$

and $\hat{\epsilon}_j$ is a random variable with zero-mean and variance σ_j^2, which represents a stochastic calibration error (note that this is not a constant systematic error). Going forward, we will assume that $\hat{\epsilon}$ is Gaussian distributed, however the analysis could be generalized to other cases.

Consider the case in which the variances σ_j^2 associated to all control parameters are known. These variances could be estimated by performing tomography on repeated applications of the same gate and inferring the distributions of the control parameters. With σ_j^2 in hand, noise scaling can be directly applied by shifting the control parameters with some additional classical noise $\hat{\delta}_j$:

$$\theta'_j \xrightarrow{\text{parameter noise scaling}} \theta'_j + \hat{\delta}_j \tag{12}$$

where the $\hat{\delta}_j$ is sampled from a zero-mean Gaussian distribution with variance $(\lambda - 1)\sigma_j^2$, such that the variance of the overall noise is scaled by a factor of $\lambda \geq 1$. Equivalently, the effect of parameter noise scaling is that of transforming Eq. (11) into

$$\theta'_j = \theta_j + \sqrt{\lambda}\hat{\epsilon}_j. \tag{13}$$

This gives a simple noise scaling procedure for ZNE that can be done without knowing the particular structure of the Hermitian operators H_j and also without knowing the Kraus operators of the corresponding error channel.

However, if we are interested in a density matrix simulation of the quantum circuit, it may still be useful to derive the analytical Kraus operators corresponding to the noise model of Eq. (11). Since in general the operators H_j do not commute with each other, this is a subtle task. For simplicity, here we derive analytically the noise channel in the case of a single-parameter *rotation-like* gate, i.e., such that it can be expressed as:

$$G(\theta) = \exp(-i\theta H/2) = \cos(\theta/2)\mathbb{I} - i\sin(\theta/2)H. \tag{14}$$

This property holds whenever $H^2 = \mathbb{I}$, including the important cases of Pauli or controlled-Pauli rotations. Moreover, since we are dealing with a single parameter, we can easily factorize the noisy operation as the ideal gate followed by a purely random rotation:

$$G(\theta') = G(\hat{\epsilon})G(\theta). \tag{15}$$

We are interested in the effect of the final noisy gate $G(\hat{\epsilon})$ on the density matrix of the system. This is given by averaging over the Gaussian probability distribution $p(\epsilon)$ associated to the random variable $\hat{\epsilon}$:

$$\begin{aligned}\mathcal{E}(\rho) &= \int_{-\infty}^{\infty} p(\epsilon)G(\epsilon)\rho G^{\dagger}(\epsilon)d\epsilon \\ &= \int_{-\infty}^{\infty} p(\epsilon)\left[\cos^2(\epsilon)\rho + i\sin(2\epsilon)[\rho, H] + \sin^2(\epsilon)H\rho H\right]d\epsilon \\ &= (1-Q)\rho + QH\rho H,\end{aligned} \tag{16}$$

where $Q = \frac{1}{2}(1 - e^{-2\sigma^2})$ is a simple function of the noise variance σ^2. For a single parameter and for rotation-like gates, the effect of the noise-model defined in Eq. (11) is a quantum channel with only two Kraus operators. The channel is probabilistic mixture of the identity operation (with probability $1 - Q$) and the unitary H (with probability Q). In the limit of a small noise variance σ^2, even if the gate does not obey the rotation-like property (14), the quantum channel is still correctly approximated by Eq. (16) up to $O[(\sigma^2)^2]$ corrections. The same derivation can be applied also for different probability distributions of the noise.

Figure 5 uses exact density matrix simulation to estimate the performance of calibration noise scaling. Here we plot the absolute value of observable error ($|E_m - E(0)|$) for randomly generated six qubit circuits. We see that calibration mitigation

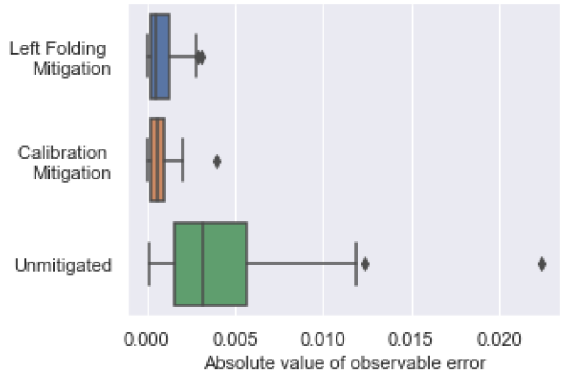

Fig. 5: Errors without mitigation, with parameter noise mitigation, and with unitary folding mitigation. Each distribution is over 50 random six-qubit circuits with random computational basis observables mitigated using gate folding from left and parameter noise scaling. The underlying noise is an angle noise channel at $\sigma^2 = 0.001$. We used a linear extrapolation with noise scale factors $\lambda = \{1, 2, 3\}$. Results were obtained with exact density matrix simulations and are presented in box plots of the distribution across the random circuits. The diamond points are outliers.

performs as well as unitary folding mitigation but it has the advantage of not adding new gates to the circuit. Thus it is likely to be less sensitive to other sources of noise such as decoherence.

III. Non-adaptive extrapolation methods: Zero noise extrapolation as statistical inference

In Section II, we discussed several methods to scale noise. In this section we study, from an estimation theory perspective, the second component of ZNE: extrapolating the measured data to the zero-nose limit.

We assume that the output of the quantum computation is a single expectation value $E(\lambda)$, where λ is the noise scale factor. This expectation could be the result of a single quantum circuit or some combinations of quantum circuits with classical post-processing. The expectation value $E(\lambda)$ is a real number which, in principle, can only be estimated in the limit of infinite measurement samples. In a real situation with N samples, only a statistical estimation of the expectation value is actually possible:

$$\hat{E}(\lambda) = E(\lambda) + \hat{\delta}, \tag{17}$$

where $\hat{\delta}$ is a random variable with zero mean and variance $\sigma^2 = \mathbb{E}(\hat{\delta}^2) = \sigma_0^2/N$, with σ_0^2 corresponding to the single-shot variance. In other words, we can sample a real prediction y from the probability distribution:

$$P(\hat{E}(\lambda) = y) = \mathcal{N}(E(\lambda) - y, \sigma^2), \tag{18}$$

where $\mathcal{N}(\mu, \sigma^2)$ is a generic distribution (typically Gaussian), with mean μ and variance $\sigma^2 = \sigma_0^2/N$.

Algorithm 1: Generic non-adaptive extrapolation

Data: A set of increasing noise scale factors $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, \dots \lambda_m\}$, with $\lambda_j \geq 1$ and fixed number of samples N for each λ_j.
Result: A mitigated expectation value
$\boldsymbol{y} \longleftarrow \emptyset$;
begin
  for $\lambda_j \in \boldsymbol{\lambda}$ **do**
    $y_j \longleftarrow ComputeExpectation(\lambda_j, N)$;
    $Append\,(\boldsymbol{y}, y_j)$;
  `/* Abitrary best fit algorithm (e.g., least squares) */`
  $\boldsymbol{\Gamma}^* \longleftarrow BestFit(E_{\text{model}}(\lambda; \boldsymbol{\Gamma}), (\boldsymbol{\lambda}, \boldsymbol{y}))$;
  return $E_{\text{model}}(0; \boldsymbol{\Gamma}^*)$;

Given a set of m scaling parameters $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, \dots \lambda_m\}$, with $\lambda_j \geq 1$, and the corresponding results

$$\mathbf{y} = \{y_1, y_2, \dots\ y_m\}, \tag{19}$$

the ZNE problem is to build a good estimator $\hat{E}(0)$ for $E(\lambda = 0)$, such that its bias

$$\text{Bias}(\hat{E}(0)) = \mathbb{E}(\hat{E}(0) - E(0)), \tag{20}$$

and its variance

$$\text{Var}(\hat{E}(0)) = \mathbb{E}(\hat{E}(0)^2) - \mathbb{E}(\hat{E}(0))^2, \tag{21}$$

are both reasonably small. More precisely, a typical figure of merit for the quality the estimator is its mean squared error with respect to the true unknown parameter:

$$\text{MSE}(\hat{E}(0)) = \mathbb{E}(\hat{E}(0) - E(0))^2 \tag{22}$$

$$= \text{Var}(\hat{E}(0)) + \text{Bias}(\hat{E}(0))^2. \tag{23}$$

If the expectation value $E(\lambda)$ can be an arbitrary function of λ without any regularity assumption, then zero-noise extrapolation is impossible. Indeed its value at $\lambda = 0$ would be arbitrary and unrelated to its values at $\lambda \geq 1$. However from physical considerations, it is reasonable to have a model for $E(\lambda)$, e.g., we can assume a linear, a polynomial or an exponential dependence with respect to λ. For example, for a depolarizing noise model, one can use the exponential ansatz given in Eq. (9).

If we chose a generic model $E_{\text{model}}(\lambda; \boldsymbol{\Gamma})$ for the quantum expectation value, where $\boldsymbol{\Gamma}$ represents the model parameters, then the zero-noise-extrapolation problem reduces to a regression problem. Algorithm 1 is the general form for a non-adaptive ZNE. Alternatively, the scale factors λ_j and the associated numbers of samples N_j can be chosen in an adaptive way, depending on the results of intermediate steps. This adaptive extrapolation method is studied in more details in Section IV.

We focus on two main non-adaptive models, the polynomial ansatz and the poly-exponential ansatz. These two general models, give rise to a large variety of specific extrapolation

algorithms. Some well known methods, such as Richardson's extrapolation, are particular cases. Some other methods have, to our knowledge, not been applied before for quantum error mitigation.

A. Polynomial extrapolation

The polynomial extrapolation method is based on the following polynomial model of degree d:

$$E^{(d)}_{\rm poly}(\lambda) = c_0 + c_1\lambda + \dots c_d\lambda^d, \tag{24}$$

where $c_0, c_1, \dots c_d$ are $d+1$ unknown real parameters. This essentially corresponds to a Taylor series approximation and is physically justified in the weak noise regime.

In general, the problem is well defined only if the number of data points m is at least equal to the number of free parameters $d+1$. As opposed to Richardson's extrapolation [5], a useful feature of this method is that we can keep the extrapolation order d small but still use a large number of data points m. This avoids an over-fitting effect: if we increase the order d by too much, then the model is forced to follow the random statistical fluctuations of our data at the price of a large generalization error for the zero-noise extrapolation. In terms of the inference error given in Eq. (22), if we increase d by too much, then the bias is reduced but the variance can grow so much that the total mean squared error is actually increased.

B. Linear extrapolation

Linear extrapolation is perhaps the simplest method and is a particular case of polynomial extrapolation. It corresponds to the model:

$$E_{\rm linear}(\lambda) = E^{(d=1)}_{\rm poly}(\lambda) = c_0 + c_1\lambda. \tag{25}$$

In this case a simple analytic solution exists, corresponding to the ordinary least squared estimator of the intercept parameter:

$$\hat{E}_{\rm linear}(0) = \bar{y} - \frac{S_{\lambda y}}{S_{\lambda\lambda}}\bar{x}, \tag{26}$$

where

$$\bar{\lambda} = \frac{1}{m}\sum_j \lambda_j, \qquad \bar{y} = \frac{1}{m}\sum_j y_j,$$
$$S_{\lambda y} = \sum_j (\lambda_j - \bar{\lambda})(y_j - \bar{y}), \quad S_{\lambda\lambda} = \sum_j (\lambda_j - \bar{\lambda})^2. \tag{27}$$

With respect to the zero noise value of the model $E_{\rm linear}(0)$, the estimator is unbiased. If the statistical uncertainty σ^2 for each y_j is the same, the variance for $\hat{E}_{\rm linear}(0)$ is:

$$\mathrm{Var}[\hat{E}_{\rm linear}(0)] = \sigma^2\left[\frac{1}{m} + \frac{\bar{\lambda}^2}{S_{\lambda\lambda}}\right]. \tag{28}$$

C. Richardson extrapolation

Richardson's extrapolation is also a particular case of polynomial extrapolation where $d = m-1$, i.e., the order is maximized given the number of data points:

$$E_{\rm Rich}(\lambda) = E^{(d=m-1)}_{\rm poly}(\lambda) = c_0 + c_1\lambda + \dots c_{m-1}\lambda^{m-1}. \tag{29}$$

This is the only case in which the fitted polynomial perfectly interpolates the m data points such that, in the ideal limit of an infinite number of samples $N \to \infty$, the error with respect to the true expectation value is by construction $O(m)$. Using the interpolating *Lagrange polynomial*, the estimator can be explicitly expressed as:

$$\hat{E}_{\rm Rich}(0) = \hat{c}_0 = \sum_{k=1}^{m} y_k \prod_{i\neq k} \frac{\lambda_i}{\lambda_i - \lambda_k}, \tag{30}$$

where we assumed that all the elements of $\boldsymbol{\lambda}$ are different.

The error of the estimator is $O(m)$ only in the asymptotic limit $N \to \infty$. In other words $O(m)$ corresponds to the bias term in Eq. (22). In a real scenario, N is finite, and the variance term in Eq. (22) grows exponentially as we increase m. This fact can be easily shown in the simplified case in which the noise scale factors are equally spaced, i.e., $\lambda_k = k\,\lambda_1$ where $k = 1, 2, \dots m$. Substituting this assumption into Eq. (30) we get:

$$\hat{E}_{\rm Rich}(0) = \sum_{k=1}^{m} y_k \prod_{i\neq k} \frac{i}{i-k} \; = \sum_{k=1}^{m} y_k\,(-1)^{k-1}\binom{m}{k}. \tag{31}$$

If we assume that each expectation value is sampled with the same statistical variance σ^2 as described in Eq. (18), since $\hat{E}_{\rm Rich}(0)$ is a linear combination of the measured expectation values $\{y_k\}$, its variance is given by:

$$\mathrm{Var}(\hat{E}_{\rm Rich}(0)) = \sigma^2 \sum_{k=1}^{m}\binom{m}{k}^2$$
$$= \sigma^2\left[\binom{2m}{m} - 1\right] \xrightarrow{m\longrightarrow\infty} \sigma^2\,\frac{2^{2m}}{\sqrt{\pi m}}, \tag{32}$$

where we used the Vandermonde's identity and, in the last step, the Stirling approximation.

The practical implication of Eq. (32) is that the zero-nose limit predicted by the Richardson's estimator is characterized by a statistical uncertainty which scales exponentially with the number of data points.

D. Poly-Exponential extrapolation

The poly-exponential ansatz of degree d is:

$$E^{(d)}_{\rm polyexp}(\lambda) = a \pm e^{z(\lambda)},\; z(\lambda) := z_0 + z_1\lambda + \dots z_d\lambda^d. \tag{33}$$

where $a, z_0, z_1, \dots z_d$ are $d+2$ parameters. From physical considerations, it is reasonable to assume that $E(\lambda)$ converges to a finite asymptotic value i.e.:

$$E(\lambda) \xrightarrow{\lambda\to\infty} a \quad \Longleftrightarrow \quad z(\lambda) \xrightarrow{\lambda\to\infty} -\infty. \tag{34}$$

There are two important scenarios: (i) where a is unknown and so a non-linear fit should be performed and (ii) where a is deduced from asymptotic physical considerations. For example, if we know that in the limit of $\lambda \to \infty$ the state of the system is completely mixed or thermal, it is possible to fix the value of a such that the poly-exponential ansatz (33) is left with only $d+1$ unknown parameters: $z_0, z_1, \dots z_d$. If the asymptotic limit a is known, we can apply the following procedure:

1) Evaluate $\{y'_k\} = \{\log(|y_k - a| + \epsilon)\}$, representing the measurement results in a convenient logarithmic space with coordinates (y'_k, λ_k), with a small regularizing constant $\epsilon > 0$.
2) The model of Eq. (33) in the logarithmic space (y'_k, λ_k) reduces to the polynomial $z(\lambda)$.
3) Estimate the zero-noise limit in the logarithmic space $\hat{z}(0) = \hat{z}_0$ with a standard polynomial extrapolation. If necessary different weights can be used for different scale factors, taking into account the non-linear propagation of statistical errors.
4) Convert back to the original space, obtaining the final estimator $\hat{E}(0) = a \pm e^{\hat{z}(0)}$.

This allows us to map a non-linear regression problem into a polynomial fit that is linear with respect to the parameters and therefore much more stable. However, many reasonable alternative approaches exist like maximum likelihood optimization. Alternatively a Bayesian approach could be used, especially if we have prior information about the parameters of the model.

E. Exponential extrapolation

Exponential extrapolation is a particular case of the more general poly-exponential method. It corresponds to the model:

$$E_{\text{exp}}(\lambda) = E^{(d=1)}_{\text{polyexp}}(\lambda) = a \pm e^{z_0 + z_1 \lambda} = a + b e^{-c\lambda}, \quad (35)$$

where the set of real coefficients a, b, c is a way of parametrizing the same ansatz, alternative but equivalent to a, z_0, z_1. This model was discussed in [6] and is generalized by our extrapolation framework. In particular, increasing the order d, for example to $d = 2$, and using the poly-exponential model (33) we can capture small deviations from the ideal exponential assumption, possibly obtaining a more accurate zero-noise extrapolation.

F. Benchmark comparisons of ZNE methods

Benchmarks comparing the performance of ZNE methods are given in Table II. In all cases, besides for Richardson extrapolation, ZNE improves on the unmitigated noise value, however the performance varies significantly. Furthermore, one scaling or extrapolation method does not strictly dominate others.

Different extrapolation methods are compared on IBMQ's London superconducting quantum processor in Fig. 6. Here random gate folding scales the noise of 50 different two-qubit randomized benchmarking circuits. The ideal expectation value for all circuits is 1. The order 2 polynomial fit, and the exponential fit outperform Richardson extrapolation.

Scaling	**Extrapolation**	**Error % (dep.)**	**Error % (amp. damp.)**
none	unmitigated	29.9 ± 5.1	16.7 ± 4.0
circuit	linear ($d=1$)	14.6 ± 4.6	5.40 ± 2.3
circuit	quadratic ($d=2$)	6.35 ± 3.6	3.53 ± 3.4
circuit	Richardson ($d=3$)	17.6 ± 11	17.9 ± 16
circuit	exponential ($a=0.25$)	2.73 ± 1.9	2.06 ± 1.6
circuit	adapt. exp. ($a=0.25$)	$\mathbf{1.27 \pm 1.1}$	2.69 ± 2.8
at random	linear ($d=1$)	15.6 ± 5.3	5.20 ± 2.4
at random	quadratic ($d=2$)	5.54 ± 4.4	8.00 ± 8.1
at random	Richardson ($d=3$)	30.0 ± 24	24.0 ± 18
at random	exponential ($a=0.25$)	2.84 ± 1.8	$\mathbf{0.95 \pm 1.0}$
at random	adapt. exp. ($a=0.25$)	1.77 ± 1.4	2.18 ± 1.2
from left	linear ($d=1$)	14.4 ± 4.5	5.16 ± 2.3
from left	quadratic ($d=2$)	6.73 ± 3.7	3.88 ± 3.7
from left	Richardson ($d=3$)	18.4 ± 12	16.1 ± 13
from left	exponential ($a=0.25$)	3.17 ± 2.1	2.19 ± 2.0
from left	adapt. exp. ($a=0.25$)	1.43 ± 1.1	3.08 ± 3.6

TABLE II: Average of 20 different two-qubit randomized benchmarking circuits with mean depth 27. The percent mean absolute error from the exact value of 1 is reported for a depolarizing noise with $p = 1\%$ and an amplitude damping channel with $\gamma = 0.01$. For all non-adaptive methods we used $\lambda = \{1, 1.5, 2, 2.5\}$. Adaptive extrapolation was iterated up to 4 scale factors. All the results reported in this table are obtained with exact density matrix simulations. The best result for each noise model is highlighted with a bold font, while errors larger than the unmitigated one are red colored.

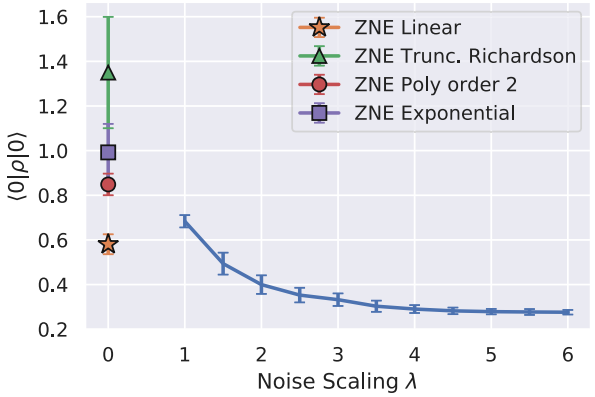

Fig. 6: Comparison of extrapolation methods averaged over 50 two-qubit randomized benchmarking circuits executed on IBMQ's "London" five-qubit chip. The circuits had, on average, 97 single qubit gates and 17 two-qubit gates. The true zero-noise value is $\langle 0|\rho|0\rangle = 1$ and different markers show extrapolated values from different fitting techniques.

In fact, Fig. 6 shows the expectation value for Richardson extrapolation when only the first 3 data points are considered. Instability in the Richardson extrapolation for more points, as described in Section III-C, causes nonphysical results when applied to all the measured data. This is an example in which vanilla Richardson extrapolation is not sufficient to provide stable results.

Algorithm 2: Generic adaptive extrapolation

Data: An initial set of m noise scale factors $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, \dots \lambda_m\}$, with $\lambda_j \geq 1$, m sample numbers $\boldsymbol{N} = (N_1, N_2, \dots N_m)$ and a maximum number of total samples N_{max}.

Result: A mitigated expectation value

begin

```
/* Initialization                                   */
```
$\boldsymbol{y} \longleftarrow \emptyset$;

for $\lambda_j \in \boldsymbol{\lambda}$ **do**

  $y_j \longleftarrow ComputeExpectation(\lambda_j, N_j)$;

  $Append\,(\boldsymbol{y}, y_j)$;

```
/* Adaptive loop                                    */
```
$N_{\text{used}} \longleftarrow 0$;

while $N_{used} < N_{max}$ **do**

  $\boldsymbol{\Gamma}^* \longleftarrow BestFit(E_{\text{model}}(\lambda; \boldsymbol{\Gamma}), (\boldsymbol{\lambda}, \boldsymbol{y}))$;

  $\lambda_{\text{next}} \longleftarrow NewScale(\boldsymbol{\Gamma}^*, \boldsymbol{\lambda}, \boldsymbol{y})$;

  $N_{\text{next}} \longleftarrow NewNumSamples(\boldsymbol{\Gamma}^*, \boldsymbol{\lambda}, \boldsymbol{y})$;

  $y_{\text{next}} \longleftarrow ComputeExpectation(\lambda_{\text{next}}, N_{\text{next}})$;

  $Append\,(\boldsymbol{\lambda}, \lambda_{\text{next}})$;

  $Append\,(\boldsymbol{y}, y_{\text{next}})$;

  $N_{\text{used}} \longleftarrow N_{\text{used}} + N_{\text{next}}$;

return $E_{\text{model}}(0; \boldsymbol{\Gamma}^*)$;

IV. Adaptive zero noise extrapolation

In Section III, we considered only non-adaptive extrapolation methods. However, in order to reduce the computational overhead, we can choose the scale factors and the number of samples in an adaptive way as described in Algorithm 2.

Differently from the non-adaptive case, in this adaptive procedure (Alg. 2) the measured scale factors $\boldsymbol{\lambda}$ are not monotonically increasing. Indeed in the adaptive step, λ_{next} can take any value (above or equal to 1). In particular, λ_{next} could also be equal to a previous scale factor λ_j, for some j. In this case, the additional measurement samples N_{next} will improve the statistical estimation of $E(\lambda_j)$.

Now, we present an example of adaptive extrapolation which is based on the exponential ansatz $E_{\text{exp}}(\lambda) = a + be^{-c\lambda}$ that we have already introduced in Eq. (35). We also assume that the asymptotic value a is known. This implies that at least two scale factors should be measured to fit the parameters b and c. We first consider this particular case and then we generalize the method to an a arbitrary number of scale factors, which will be chosen in an adaptive way.

A. Exponential extrapolation with two scale factors

We assume only two scale factors λ_1 and λ_2 (typically, λ_1 is 1). As discussed in Section III, we can estimate the corresponding expectation values, $E(\lambda_1)$ and $E(\lambda_2)$, with a statistical uncertainty of $\sigma_1^2 = \sigma_0^2/N_1$ and $\sigma_2^2 = \sigma_0^2/N_2$, respectively. Here, we are implicitly assuming that the single shot variance σ_0^2 is independent of λ, such that the estimation precision is only determined by number of samples N_1 and N_2. The measurement process will produce two results y_1 and y_2, whose statistical distribution is given by Eq. (18).

Since the parameter a is known, we can use the points (λ_1, y_1) and (λ_2, y_2) to estimate b and c of Eq. (35). The two estimators $\hat{b}$ and $\hat{c}$ can be determined by the unique ansatz interpolating the two points, whose parameters are:

$$\hat{c} = \frac{1}{\lambda_2 - \lambda_1} \log \frac{y_1 - a}{y_2 - a}\,, \tag{36}$$

$$\hat{b} = (y_1 - a)^{\frac{\lambda_2}{\lambda_2 - \lambda_1}} (y_2 - a)^{-\frac{\lambda_1}{\lambda_2 - \lambda_1}}\,. \tag{37}$$

The corresponding estimator for the zero-noise limit is $\hat{E}_{\text{exp}}(0) = a + \hat{b}$ where, since a is known, the error is only due to the statistical noise of $\hat{b}$.

This estimator depends on the empirical variables y_1, y_2, with statistical variances $\sigma_1^2 = \sigma_0^2/N_1$ and $\sigma_2^2 = \sigma_0^2/N_2$ respectively. Such measurement errors will propagate to the estimator $\hat{b}$. To leading order in σ_1^2 and σ_2^2, we have:

$$\text{MSE}(\hat{b}) = \left(\frac{\partial \hat{b}}{\partial y_1}\right)^2 \sigma_1^2 + \left(\frac{\partial \hat{b}}{\partial y_2}\right)^2 \sigma_2^2\,. \tag{38}$$

The explicit evaluation of Eq. (38), yields:

$$\text{MSE}(\hat{b}) = \frac{\sigma_0^2}{(\lambda_2 - \lambda_1)^2} \left[\frac{\lambda_2^2\, e^{2c\lambda_1}}{N_1} + \frac{\lambda_1^2\, e^{2c\lambda_2}}{N_2}\right]. \tag{39}$$

The previous equation shows that the error depends on the choice of the scale factors λ_1 and λ_2 but also on the associated measurement samples N_1 and N_2.

1) Error minimization

Let us first assume that we have at disposal only a total budget $N_{\text{max}} = N_1 + N_2$ of circuit evaluations and that λ_1 and λ_2 are fixed. Minimizing Eq. (39), with respect to N_1 and N_2, we get:

$$\begin{aligned} N_1 &= N_{\text{max}} \frac{\lambda_1}{\lambda_1 + \lambda_2\, e^{-c(\lambda_2 - \lambda_1)}} \\ N_2 &= N_{\text{max}} \frac{\lambda_2\, e^{-c(\lambda_2 - \lambda_1)}}{\lambda_1 + \lambda_2\, e^{-c(\lambda_2 - \lambda_1)}} \end{aligned} \tag{40}$$

and the corresponding error becomes:

$$\text{MSE}(\hat{b}) = \sigma_0^2 \left[\frac{\lambda_2\, e^{c\lambda_1} + \lambda_1\, e^{c\lambda_2}}{\lambda_2 - \lambda_1}\right]^2. \tag{41}$$

This error can be further minimized with respect to the choice of the scale factors. Since λ_1 is usually fixed to 1, we optimize over λ_2, leading to the condition:

$$e^{c(\lambda_2 - \lambda_1)} (c(\lambda_2 - \lambda_1) - 1) - 1 = 0. \tag{42}$$

We can solve the previous equation numerically, obtaining:

$$c(\lambda_2 - \lambda_1) = \alpha, \tag{43}$$

where $\alpha \simeq 1.27846$ is a numerical constant. For a fixed λ_1, the previous condition determines the optimal choice of the scale factor λ_2 which minimizes the zero-nose extrapolation error. From a practical point of view, Eqs. (40) and (43) can only be used if we have some prior knowledge about c. This motivates the following adaptive algorithm.

Algorithm 3: Adaptive exponential extrapolation

Data: An exponential model $E_{\text{exp}}(\lambda) = a + be^{-c\lambda}$ with a known/estimated a. A maximum number of total samples N_{max}, a fixed number of samples per iteration N_{batch} and a minimum scale factor λ_1 (typically equal to 1).

Result: A mitigated expectation value

begin

$c \longleftarrow 1$; /* Initial guess */

$\alpha \longleftarrow 1.27846$; /* Alpha in Eq. (43) */

$\mathbf{data} \longleftarrow \emptyset$;

$N_{\text{used}} \longleftarrow 0$;

while $N_{used} < N_{max}$ **do**

$\lambda_2 \longleftarrow \lambda_1 + \alpha/c$;

$N_1 \longleftarrow N_{\text{batch}} \times \frac{c\,\lambda_1/\alpha}{c\,\lambda_1+\alpha-1}$;

$N_2 \longleftarrow N_{\text{batch}} \times \frac{(1+c\,\lambda_1/\alpha)(\alpha-1)}{c\,\lambda_1+\alpha-1}$;

$N_{\text{used}} \longleftarrow N_{\text{used}} + N_1 + N_2$;

$y_1 \longleftarrow ComputeExpectation(\lambda_1, N_1)$;

$y_2 \longleftarrow ComputeExpectation(\lambda_2, N_2)$;

$Append\,(\mathbf{data}, (\lambda_1, y_1))$;

$Append\,(\mathbf{data}, (\lambda_2, y_2))$;

/* New estimate of c */

$c \longleftarrow BestFit(E_{\text{exp}}(\lambda; a, b, c), \mathbf{data})$;

return $E_{\text{exp}}(0; a, b, c)$;

B. An adaptive exponential extrapolation algorithm

Algorithm 3 is an adaptive exponential algorithm based on the exponential ansatz $E_{\text{exp}}(\lambda) = a + be^{-c\lambda}$, where a is a known constant. Figure 7 shows a comparison of adaptive exponential extrapolation with non-adaptive exponential extrapolation. At almost all sample levels, adaptive extrapolation outperforms the non adaptive approach.

V. Conclusion

We make zero-noise extrapolation digital, developing the unitary folding framework to run error mitigation with instruction set level access. We then demonstrate improved performance through a set of non-adaptive and adaptive extrapolation methods. We emphasize that zero-noise extrapolation is in general an inference problem with many avenues for further optimization.

While ZNE has previously been benchmarked on randomized benchmarking circuits or VQE, we give benchmarks of ZNE on MAXCUT problems solved with QAOA. This allows us to smoothly benchmark the performance of ZNE on larger variational quantum circuits then have been considered previously.

We also consider specialization of zero-noise extrapolation to different noise models, using calibration noise as an example. With more sophisticated multi-parameter noise models (such as a combination of calibration noise and amplitude dampening), it is likely that multi-dimensional noise extrapolation [28] will be of interest.

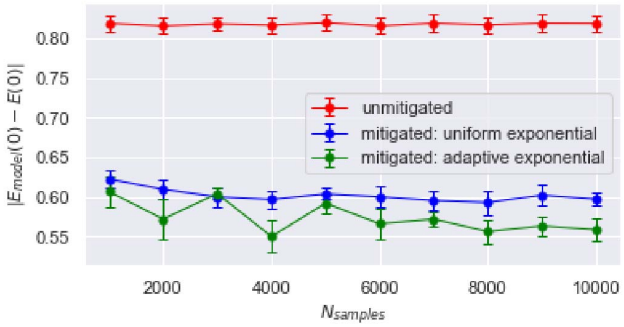

Fig. 7: Comparison of adaptive and non-adaptive exponential zero noise extrapolation, given a fixed budget of samples. The adaptive method generally produces a more accurate extrapolation with less samples. On the other hand, in this example, the advantage of adaptivity is not particularly large. Likely, this is due to the fact that the scale factors used for the non-adaptive method are already quite good and not far from their optimal values. Data was generated by exact density matrix simulation of 5-qubit randomized benchmarking circuits of depth 10 under 5% depolarizing noise and measured in the computational basis. Noise was scaled directly by access to the back-end simulator rather than with a folding method.

This work is a first step towards viewing zero-noise extrapolation as an inference problem and has opportunities for extension. Priors or constraints from observable, noise or circuit structure could be included. Data could be gathered from similar executions over time so that inference includes a historical database of previous computations.

Error-mitigation is likely to remain a critical toolkit for the NISQ-era quantum programmer. Improving and benchmarking these techniques will likewise remain an important task.

Acknowledgment

We thank Jonathan Dubois and Lorenza Viola for helpful discussions on the calibration noise model and Nathan Shammah for feedback on the manuscript. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing under Award Number DE-SC0020266 and by IBM under Sponsored Research Agreement No. W1975810. We thank IBM for providing access to their quantum computers. The views expressed in this paper are those of the authors and do not reflect those of IBM.

References

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

[2] B. M. Terhal, "Quantum error correction for quantum memories," 4 2015.

[3] D. Gottesman, "An introduction to quantum error correction and fault-tolerant quantum computation," pp. 13–58, 2010.

[4] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," 8 2012.

[5] K. Temme, S. Bravyi, and J. M. Gambetta, "Error Mitigation for Short-Depth Quantum Circuits," *Physical Review Letters*, vol. 119, p. 180509, 11 2017.
[6] 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.
[7] J. J. Wallman and J. Emerson, "Noise tailoring for scalable quantum computation via randomized compiling," *Physical Review A*, vol. 94, no. 5, p. 052325, 2016.
[8] E. Knill, "Quantum computing with realistically noisy devices," *Nature*, vol. 434, no. 7029, pp. 39–44, 2005.
[9] 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.
[10] 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.
[11] 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.
[12] 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.
[13] H. Ball, M. J. Biercuk, A. Carvalho, R. Chakravorty, J. Chen, L. A. de Castro, S. Gore, D. Hover, M. Hush, P. J. Liebermann, *et al.*, "Software tools for quantum control: Improving quantum computer performance through noise and error suppression," *arXiv preprint arXiv:2001.04060*, 2020.
[14] T. J. Green, J. Sastrawan, H. Uys, and M. J. Biercuk, "Arbitrary quantum control of qubits in the presence of universal noise," *New Journal of Physics*, vol. 15, no. 9, p. 095004, 2013.
[15] Y. Li and S. C. Benjamin, "Efficient Variational Quantum Simulator Incorporating Active Error Minimization," *Physical Review X*, vol. 7, 6 2017.
[16] 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.
[17] R. S. Smith, M. J. Curtis, and W. J. Zeng, "A practical quantum instruction set architecture," *arXiv preprint arXiv:1608.03355*, 2016.
[18] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, "Efficient Z gates for quantum computing," *Physical Review A*, vol. 96, 8 2017.
[19] X. Fu, L. Riesebos, M. Rol, J. van Straten, J. van Someren, N. Khammassi, I. Ashraf, R. Vermeulen, V. Newsum, K. Loh, *et al.*, "eqasm: An executable quantum instruction set architecture," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 224–237, IEEE, 2019.
[20] 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.
[21] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski, "Cloud quantum computing of an atomic nucleus," *Physical review letters*, vol. 120, no. 21, p. 210501, 2018.
[22] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm," 11 2014.
[23] J. P. Barnes, C. J. Trout, D. Lucarelli, and B. D. Clader, "Quantum error-correction failure distributions: Comparison of coherent and stochastic error models," *Phys. Rev. A*, vol. 95, p. 062338, Jun 2017.
[24] C. B. Mendl and M. M. Wolf, "Unital Quantum Channels - Convex Structure and Revivals of Birkhoff's Theorem," *Communications in Mathematical Physics*, vol. 289, pp. 1057–1086, Aug. 2009.
[25] J. True Merrill and K. R. Brown, "Progress in compensating pulse sequences for quantum computation," *arXiv e-prints*, p. arXiv:1203.6392, Mar. 2012.
[26] Z.-H. Wang, W. Zhang, A. M. Tyryshkin, S. A. Lyon, J. W. Ager, E. E. Haller, and V. V. Dobrovitski, "Effect of pulse error accumulation on dynamical decoupling of the electron spins of phosphorus donors in silicon," *Phys. Rev. B*, vol. 85, p. 085206, Feb 2012.
[27] Z.-H. Wang and V. V. Dobrovitski, "Aperiodic dynamical decoupling sequences in the presence of pulse errors," *Journal of Physics B Atomic Molecular Physics*, vol. 44, p. 154004, Aug. 2011.
[28] M. Otten and S. K. Gray, "Recovering noise-free quantum observables," *Physical Review A*, vol. 99, no. 1, p. 012338, 2019.