Supplementary Information for Efficient Noise Mitigation Technique for Quantum Computing
Ali Shaib, Mohamad Hussein Naim, Mohammed E. Fouda, Rouwaida Kanj, and Fadi Kurdahi

This supplementary document supports the discussion in the main text by providing theoretical proof and technical details. Section 1 provides the proof of Lemma 1. Section 2 provides information about the Jensen-Shannon Divergence (JSD). Section 3 summarizes our proposed algorithm for the construction of $Q_m$. Section 4 compares the performance of $Q_m$ between using $p(|in\rangle)$ and $p_{ave}$. Section 5 presents the average JSD and standard deviation for each input state $|in\rangle$ over different depths.

I. PROOF OF LEMMA 1

**Lemma 1:** Let $\lambda$ and $p$ be the respective eigenvalues and error rates of a Pauli Channel with $n$ qubits, then

$$\lambda^m = WM^m |0\rangle$$

where $W$ is a Walsh-Hadamard Transform, and $M$ is a $2^n \times 2^n$ matrix such that $M_{ij} = p_{i \oplus j}$ ($\oplus$ is the bitwise exclusive-OR operator).

**Proof of Lemma 1:** For the LHS of Lemma 1, we have

$$\lambda = WP$$

Then

$$\lambda_i = \sum_{k=0}^{2^n-1} (-1)^{ik} p_k$$

And,

$$\lambda_i^m = \left(\sum_{k=0}^{2^n-1} (-1)^{ik} p_k\right)^m$$

For the RHS of Lemma 1, using Lemma 2, we get

$$WM^m |0\rangle = W\left(\frac{1}{2^n} WTW\right)^m |0\rangle = W\left(\frac{1}{2^n} WTmW\right) |0\rangle$$

$$= \Gamma^m W |0\rangle = \Gamma^m \bar{1}$$

Then

$$(WM^m |0\rangle)_i = \Gamma^m_{ii} = \left(\sum_{k=0}^{2^n-1} (-1)^{ik} p_k\right)^m = \lambda_i^m$$

Therefore,

$$\lambda^m = WM^m |0\rangle$$

**Lemma 2:** For a given $p$, $M$ can be written as
\[ M = \frac{1}{2^n} WTW \]

where \( W \) is Walsh-Hadamard Transform, and \( \Gamma \) is a diagonal matrix with \( \Gamma_{ii} = \sum_{k=0}^{2^n-1} (-1)^{ik} p_k \).

**Proof of Lemma 2:**

By applying Singular Value Decomposition (SVD), \( M \) can be written as \( M = VTV^{-1} \) where we assume that \( V = W \) and \( V^{-1} = \frac{1}{2^n} W \). The matrix \( \Gamma \) can then be written as

\[ \Gamma = V^{-1} MV \]

where \( M_{ij} = p_{i \oplus j} \). We just have to prove that \( \Gamma \) is a diagonal matrix. By the definition of the Walsh-Hadamard Transform, \( W_{ij} = (-1)^{ij} \), where \( ij \) is the bitwise inner product between two \( n \)-bit strings \( i \) and \( j \), modulo 2. We have

\[ \Gamma = \frac{1}{2^n} W M W = \frac{1}{2^n} RW \]

where \( R = WM \). Thus, \( R_{ij} \) can be written as

\[ R_{ij} = \sum_{r=0}^{2^n-1} W_{ir} M_{rj} = \sum_{r=0}^{2^n-1} (-1)^{ir} p_{r \oplus j} \]

\[ (RW)_{ij} = \sum_{s=0}^{2^n-1} R_{is} W_{sj} = \sum_{r=0}^{2^n-1} \sum_{s=0}^{2^n-1} (-1)^{ir \oplus js} p_{r \oplus s} = \sum_{k=0}^{2^n-1} (-1)^{jk} \left( \sum_{r=0}^{2^n-1} (-1)^{i \oplus j} r \right) p_k \]

After some simplifications,

\[ \Gamma_{ij} = \frac{1}{2^n} (RW)_{ij} = \left\{ \begin{array}{ll} \sum_{k=0}^{2^n-1} (-1)^{ik} p_k, & i = j \\ 0, & \text{otherwise} \end{array} \right. \]

Thus, \( \Gamma \) is a diagonal matrix.

**II. JENSEN-SHANNON DIVERGENCE (JSD)**

The JSD is a method used to measure the similarity between two probability distributions \( p \) and \( q \). The JSD ranges between 0 and 1. The lower the JSD, the closer the two distributions are. The JSD between \( p \) and \( q \) is calculated as

\[ JSD(p, q) = \frac{1}{2} D(p||m) + \frac{1}{2} D(q||m) \]

where

\[ m = \frac{1}{2} (p + q) \]

\[ D(P, Q) = \sum_x P(x) \log \left( \frac{P(x)}{Q(x)} \right) \]
III. CONSTRUCTING OF THE MITIGATION MATRIX

The following algorithm summarizes our proposed protocol for constructing the mitigation matrix $Q_m$ for a new depth $m$ not included in our training set of depths.

Algorithm 1 Proposed Protocol

Require: Choose a training set $T = [1, m_{max}]$.
1: Choose circuit depth $m \in T$.
2: Sample a random sequence $s \in S_m$. The set of all length $m$ sequences of one-qubit Clifford gates applied independently on each qubit, followed by an inverse gate for this sequence.
3: Obtain an estimate $\hat{q}(m, s, |in\rangle)$ of the probability distribution over the $2^n$ possible measurement outcomes.
4: Repeat steps 2-3 for $K$ times to obtain an estimate of the average probability distribution $\bar{q}(m, |in\rangle)$ where
   $$\bar{q}(m, |in\rangle) = \frac{1}{K} \sum q(m, s, |in\rangle)$$
5: Apply an input-specific permutation matrix $\Pi_{in}$ on $\bar{q}(m, |in\rangle)$ where
   $$\Pi_{in,j} = \begin{cases} 1 & \text{if } i \oplus j = in \\ 0 & \text{otherwise} \end{cases}$$
6: Apply Walsh-Hadamard transform $W$ on $\bar{q}(m, |in\rangle)$ to obtain
   $$\Lambda(m) = W\bar{q}(m, |in\rangle)$$
7: Repeat steps 1-6 for all $m \in T$.
8: For each parameter $\Lambda_i(m) \in \Lambda(m)$, fit the model
   $$\Lambda_i(m) = A_i\lambda_i^m$$
9: Apply an Inverse Walsh-Hadamard transform on $\lambda$ to obtain the error rates of the Pauli Channel
   $$p = W^{-1}\lambda$$
10: Construct the input-specific spam error matrix $N_{in} = W^{-1}AW$ where $A$ is a diagonal matrix with whose diagonal entries $A_i$.
11: Construct the input-specific average gate error matrix $M_{in}$ where $M_{in,i,j} = p_i \oplus j$ ($\oplus$ is the bitwise exclusive-Or operator).
12: Repeat Steps 1-11 for all $|in\rangle \in \{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$.
13: For all basis input states, estimate $\bar{q}(m', |in\rangle)$ as
   $$\bar{q}(m', |in\rangle) = N_{in}M_{in}^{m'} |in\rangle$$
14: Construct the mitigation matrix $Q_{m'}$ where
   $$Q_{m'} = [\bar{q}(m', |0\rangle), \bar{q}(m', |1\rangle), \ldots, \bar{q}(m', |2^n - 1\rangle)]$$
IV. OTHER QUANTUM COMPUTERS

In this section, we show the results of evaluating the accuracy of the model where we include the IBM Q Athens quantum computer. The Athens quantum computer is not included in the main manuscript since it is retired by IBM. Hence, we could not include the results of the mitigation. Similar to Figure 1 in the main manuscript, Figure 1 below presents the computed JSD for different quantum computers while varying $m_{max}$ with the addition of IBM Q Athens that demonstrates JSD values that range between 0.005 and 0.08. Figure 2 presents the average and standard deviation for the test JSD values for the different quantum computers including IBM Q Athens that demonstrates an average test JSD between 0.02 and 0.04 for the different $m_{max}$ values.

![Graphs showing JSD values for different quantum computers](image)

Fig. 1: $JSD(\hat{q}(m,|in)), q'(m,|in))$ for training sets of depths $T$ and testing sets of depths $T'$ with variable maximum training depth $m_{max} \in \{20, 50, 80\}$ on different IBMQ 5-qubit quantum computers.
Fig. 2: The average and standard deviation of $JSD(\hat{q}(m,|0\rangle), q'(m,|0\rangle))$; (a) over all depths $m \in [m_{\text{max}} + 1, 100]$ and (b) over depths $m \in [80, 100]$ while varying the maximum training depth $m_{\text{max}}$ on different IBM Q 5-qubit quantum computers.
V. Comparing the performance with \( p_{ave} \) vs \( p_{in} \)

In this section, we evaluate the mitigation power of \( Q_m \) when relying on the input specific error rate vectors \( p(\ket{in}) \) to construct \( M_{in} \) for all inputs compared to relying on the average error rate \( p_{avg} = \frac{1}{2^n} \sum_{in=0}^{2^n-1} p(\ket{in}) \) to construct \( M_{in} = M_{avg} \). \( N_{in} \) remains input specific. Similar to Figure 3 in the main manuscript, Figure 3 below compares the average JSD between the ideal outputs and each of the unmitigated outputs, the mitigated outputs by the MEM protocol, and the mitigated outputs by \( Q_m \) using \( p_{ave} \) in addition to the mitigated outputs by \( Q_m \) using \( p(\ket{in}) \). Figure 3 shows that the difference between using \( p_{ave} \) and \( p(\ket{in}) \) for the construction of \( Q_m \) is negligible for the different quantum computers indicating consistent SPAM-free error rates for all inputs.

Fig. 3: Average JSD between the ideal output \( \ket{in} \) and each of the unmitigated outputs \( \hat{q}(m,s,\ket{in}) \), mitigated outputs by the MEM protocol, and mitigated outputs by our proposed noise model using \( p(\ket{in}) \) and using \( p_{ave} \) for each depth \( m \) on IBM Q 5-qubit quantum computers.
VI. AVERAGE JSD FOR DIFFERENT INPUT STATES

In this section, we elaborate further on Figure 3 in the main manuscript and show the average JSD between the ideal output \(|in\rangle\) and the mitigated output by our proposed protocol for each depth \(m\) and input state \(|in\rangle\) (Figure 4). Figure 5 presents the average and standard deviation of JSD between the ideal output and mitigated output by our proposed protocol over all depths for each input state \(|in\rangle\). We notice that, for the different quantum computers, some input states demonstrate lower JSD values compared to other input states. The Lima quantum computer presents the best average JSD in the worst input state with a minimum of 0.0023 and a maximum of 0.107 followed by the Belem quantum computer showing a minimum of 0.074 and a maximum of 0.202 and the Manila quantum computer showing a minimum of 0.031 and a maximum of 0.280. Table 1 shows the average reduction in the test JSD using the proposed protocol over the unmitigated data is about 58.33\%, 61.04\%, and 85.82\% for the Belem, Manila, and Lima computers, respectively, compared to a reduction of 16.67\%, 27.12\%, and 72.77\% for the Belem, Manila, and Lima computers, respectively, using the MEM protocol. We report up to 69\% improvement and on average 40\% improvement compared to the MEM protocol.

![Figure 4: Average JSD between the ideal output |in⟩ and the mitigated outputs by our proposed noise model for each depth m and each input state |in⟩ on IBM Q 5-qubit quantum computers.](image-url)
Fig. 5: Average and standard deviation of $JSD$ over all depths $m$ between the ideal output $|i\rangle$ and the mitigated output by our proposed noise model for each input state $|i\rangle$ on IBM Q 5-qubit quantum computers.
TABLE I: Average test JSD between the ideal output $|in\rangle$ and each of the unmitigated output, mitigated by the proposed protocol output, and mitigated by the MEM protocol output for the different quantum computers.

| m  | Unmitigated JSD | MEM JSD | Mitigated by Proposed Protocol JSD | MEM improvement % | Proposed improvement % |
|----|-----------------|---------|-----------------------------------|-------------------|------------------------|
| 10 | 0.30            | 0.21    | 0.08                              | 30.00             | 73.33                  |
| 30 | 0.34            | 0.28    | 0.13                              | 17.05             | 55.88                  |
| 50 | 0.37            | 0.31    | 0.17                              | 16.22             | 54.05                  |
| 70 | 0.39            | 0.33    | 0.18                              | 15.38             | 53.85                  |
| 90 | 0.40            | 0.35    | 0.17                              | 12.30             | 57.50                  |
| mean | 0.36         | 0.30    | 0.15                              | 16.67             | 58.33                  |

| m  | Unmitigated JSD | MEM JSD | Mitigated by Proposed Protocol JSD | MEM improvement % | Proposed improvement % |
|----|-----------------|---------|-----------------------------------|-------------------|------------------------|
| 10 | 0.26            | 0.15    | 0.12                              | 42.31             | 53.85                  |
| 30 | 0.28            | 0.19    | 0.11                              | 32.14             | 60.71                  |
| 50 | 0.31            | 0.23    | 0.12                              | 25.84             | 61.29                  |
| 70 | 0.33            | 0.27    | 0.12                              | 18.18             | 63.64                  |
| 90 | 0.35            | 0.29    | 0.12                              | 17.14             | 65.71                  |
| mean | 0.31         | 0.23    | 0.12                              | 27.12             | 61.04                  |

| m  | Unmitigated JSD | MEM JSD | Mitigated by Proposed Protocol JSD | MEM improvement % | Proposed improvement % |
|----|-----------------|---------|-----------------------------------|-------------------|------------------------|
| 10 | 0.25            | 0.02    | 0.03                              | 92.00             | 88.00                  |
| 30 | 0.28            | 0.04    | 0.04                              | 85.71             | 85.71                  |
| 50 | 0.30            | 0.08    | 0.04                              | 73.33             | 86.67                  |
| 70 | 0.31            | 0.12    | 0.05                              | 61.29             | 85.87                  |
| 90 | 0.33            | 0.16    | 0.05                              | 51.22             | 84.85                  |
| mean | 0.29         | 0.08    | 0.04                              | 72.77             | 85.82                  |

VII. COMPARISON AGAINST RNZE

We compared the proposed technique against Richardson Zero Noise Extrapolation (RZNE) as requested. The following is the method we adopted to have as fair as comparison.

Richardson Zero Noise Extrapolation has two main components: noise scaling and then extrapolation. Global folding is a method for amplifying the noise by mapping a quantum circuit $G$ as follows [1]:

$$G \rightarrow GG^\dagger G$$

This makes the circuit longer, which adds more noise, while keeping its effect unchanged ($G^\dagger = G^{-1}$). Folding a circuit is done using a scale factor $\lambda$, an integer which corresponds to how much the length of the circuit is scaled. The minimum scale factor is one which corresponds to $G \rightarrow G$ (i.e. no folding), $\lambda = 3$ corresponds to $G \rightarrow GG^\dagger G$, $\lambda = 5$ corresponds to $G \rightarrow GG^\dagger GG^\dagger G$, and so on.

For a given state $\rho$, we denote by $P(|0\rangle) = \langle 0 | \rho | 0 \rangle$. For a given n-qubit circuit of depth $m$ and input $|0\rangle$, we perform RZNE as follows [2]:

1) Repeat steps 2-3 for each $\lambda \in [1,3,5,7,9]$.
2) Scale the circuit by $\lambda$.
3) Measure $P(|0\rangle)_{unmitigated}$.
4) Fit the results to the model:

$$E_{Rich}(\lambda) = c_0 + c_1 \lambda + \ldots + c_{n-1} \lambda^{n-1}$$

where $n$ is the given number of data points ($n = 6$ in this case).
5) Estimate $P(|0\rangle)_{mitigated}$ at $\lambda = 0$ i.e.

$$P(|0\rangle)_{mitigated} = E_{Rich}(0)$$

$S_m$ is the set of all length m sequences of one-qubit Clifford gates applied independently on each qubit, followed by an inverse gate for this sequence. This given, we performed the Richardson ZNE as follows:

1) Repeat steps 2-4 for each depth $m \in [1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$
Fig. 6: RZNE performed on a circuit of depth $m = 100$ executed on Qiskit Aer simulator with with noise model of IBMQ’s “Lima” five-qubit quantum computer. The true zero-noise value $P\bra{0} = 1$.

2) Sample 100 random sequences of gates $s \in S_m$.
3) Perform RZNE on each circuit to obtain $P(m, s, \bra{0})_{mitigated}$
4) Obtain an estimate of the average $P(m, \bra{0})_{mitigated}$ by averaging over all sequences

We compare the RNZE results with our following experiment [3]:
1) Repeat steps 2-4 for each depth $m \in [1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$.
2) Sample 100 random sequences of gates $s \in S_m$.
3) Obtain an estimate of the probability distribution over the $2^n$ possible measurement outcomes.
4) Obtain an estimate of the noisy average probability distribution $\hat{q}(m, \bra{0})$ by averaging over all sequences.
5) Follow our paper approach to generate as $Q_m = W^{-1} AW M^m$
6) Estimate the mitigated average probability distribution $\hat{q}(m)_{mitigated} = Q_m^{-1}\hat{q}(m \bra{0})$
7) Compute $P(m, \bra{0})_{mitigated} = \bra{0}\hat{q}(m_{mitigated})\bra{0}$

We generated our data by running 100 random circuits for each depth $m$ with 1024 shots per circuit using IBMQ Aer simulator with different noise models for each of the IBMQ Lima and Belem quantum computers [3]. We used $\bra{0}\rho\bra{0}$ as our evaluation metric since it’s used in [2] to evaluate RZNE.

While RZNE is circuit-dependent which means it can not be applied on any circuit of any depth without considering average noise, our protocol is generic since it can be used to mitigate the results of any random circuit of depth $m$. On the other side, RZNE is more computationally efficient since it does not need to compute the average noisy distributions over different circuit depths. It only requires replicating the circuit for different scaling factors. However, RZNE becomes relatively less efficient over large depths.

REFERENCES
[1] “https://mitiq.readthedocs.io/en/v0.2.0/guide/guide-zne.html,” Jan 2023.
[2] 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). IEEE, 2020, pp. 306–316.
![Graphs showing comparison of mitigation methods averaged over 100 randomized benchmarking circuits executed on Qiskit Aer simulator with noise model of (a) IBM Lima quantum computer and (b) IBM Belem quantum computer for five-qubit quantum computer. The true zero-noise value $P|0\rangle = 1$.](image)

Fig. 7: Comparison of mitigation methods averaged over 100 randomized benchmarking circuits executed on Qiskit Aer simulator with noise model of (a) IBM Lima quantum computer and (b) IBM Belem quantum computer for five-qubit quantum computer. The true zero-noise value $P|0\rangle = 1$.

[3] A. Asfaw, A. Corcoles, L. Bello, Y. Ben-Haim, M. Bozzo-Rey, S. Bravyi, N. Bronn, L. Capelluto, A. C. Vazquez, J. Ceroni, R. Chen, A. Frisch, J. Gambetta, S. Garion, L. Gil, S. D. L. P. Gonzalez, F. Harkins, T. Imamichi, H. Kang, A. h. Karamlou, R. Loredo, D. McKay, A. Mezzacapo, Z. Minev, R. Movassagh, G. Nannicini, P. Nation, A. Phan, M. Pistoia, A. Rattew, J. Schaefer, J. Shabani, J. Smolin, J. Stenger, K. Temme, M. Tod, S. Wood, and J. Wootton. (2020) Learn quantum computation using qiskit. [Online]. Available: http://community.qiskit.org/textbook