IBM Quantum Awards Open Science Prize 2021 – Graph State Challenge: A generalized error mitigation (GEM) approach

Frima Kalyuzhner and Emanuele G. Dalla Torre

Department of Physics, Bar-Ilan University, Ramat Gan 5290002, Israel and
Center for Quantum Entanglement Science and Technology,
Bar-Ilan University, Ramat Gan 5290002, Israel

Adi Makmal

Faculty of Engineering, Bar-Ilan University, Ramat Gan 5290002, Israel and
Center for Quantum Entanglement Science and Technology,
Bar-Ilan University, Ramat Gan 5290002, Israel

Graph states are a fundamental building block of several quantum algorithms, including measurement based quantum computations, Bell inequality violations, and more. In NISQ devices, the fidelity of these states is limited by gate imperfections and measurement errors. For instance, in the IBM Casablanca quantum processor, the maximal fidelity obtainable for a 7-qubit graph state is 76%. The goal of this challenge is to improve this value by using either a more efficient state preparation, or better algorithms for error mitigation. In these notes we show how to improve the fidelity of the graph state by post-processing the quantum measurement using non-linear functions, expressed in terms of neural networks with finite biases and thresholds. This approach is shown to generalize known mitigation methods. To demonstrate the performance of this method, we train our network using a complete set of calibration measurements generated by the QISKIT simulator. We, then, apply the neural network on actual measurements of the graph state and achieve a fidelity of 98.9%.

I. MEASUREMENT-ERROR MITIGATION

In the current NISQ era, the practical applicability of quantum computers is hindered by gate infidelity and state-preparation and measurement (SPAM) errors: The former type of error refers to unwanted unitary and non-unitary processes that occur during the quantum algorithm; The latter type generally occurs in the translation from the quantum to the classical world. Standard quantum error correction schemes require hundreds of physical qubits to represent merely a single logical qubit and hence cannot be employed in existing intermediate-scale machines[1]. Instead, several classical post-processing schemes have emerged to mitigate SPAM errors, see e.g. [2–6].

[1] See, for example, Ref. [1] and references therein for a state-of-the-art proposal requiring 32,000 qubits to obtain quantum advantage.

The common approach to SPAM error mitigation relies on the assumption of a linear relation between p_{noisy} and p_{clean}, the output probabilities of the quantum algorithm, respectively, in the presence and in the absence of SPAM errors:

$$p_{\text{noisy}} = A_{\text{SPAM}}\, p_{\text{clean}}. \tag{1}$$

Here, for a system of n qubits, p_{noisy} and p_{clean} are real vectors of size 2^n, and A_{SPAM} is a $2^n \times 2^n$ matrix. The element $A_{\text{SPAM}}(m, l)$ represents the conditional probability to observe the state m in the noisy channel, given that the measurement outcome of the clean channel is the state l. Probability conservation requires that the sum of elements on each column of A_{SPAM} equals 1, or equivalently that A_{SPAM} is a *stochastic matrix*.

Under the assumption that the matrix A_{SPAM} depends solely on the physical properties of the quantum device and is constant for all quantum circuits, one can use calibration data to measure this matrix and get rid of all SPAM errors, using the inverse of Eq. (1):

$$p_{\text{clean}} = A^{-1}_{\text{SPAM}}\, p_{\text{noisy}}, \tag{2}$$

where A^{-1}_{SPAM} is the inverse of A_{SPAM}. For large quantum systems, this approach has two major problems: first, obtaining the calibration data for 2^n possible outcomes requires an exponentially large number of quantum circuits; and second, inverting a matrix of size $2^n \times 2^n$ is extremely costly (the best known algorithm scales as $O(d^{2.37})$, where $d = 2^n$) and noisy: because the calibration measurements are performed on a finite number of shots, the estimated A_{SPAM} is affected by sampling noise, which is then strongly enhanced in the inversion process.

To solve these problems, different methods were proposed to approximate A_{SPAM}, reducing both the number of required calibrations, and the computational cost required to invert this matrix. In particular, Ref. [5] introduced a method named continuous-time Markov process (CTMP), which is based on the physical assumption that error correlations between a set of qubits decrease with the size of the set. The first-order CTMP approximation assumes that SPAM errors on different qubits are independent, such that A_{SPAM} is block diagonal in the qubit base (see also [6]). The second order assumes that only two-qubit correlations exist, such that A_{SPAM} can be written as the exponent of the sum of two-qubit operators, and so on so forth. By implementing the CTMP approach on IBM quantum computers, Ref. [5] showed that the first order approximation leads to a significant improvement and the second order to a much smaller one, hence, validating the key assumption of this method. When applied to the problem at hand (a 7-qubit graph state on the Casablanca quantum computer), second-order CTMP leads to a fidelity improvement from 62% to 76%.

II. GENERALIZE ERROR MITIGATION (GEM)

In these notes we challenge the main assumption of the aforementioned SPAM error mitigation codes, namely that the relation between p_{clean} and p_{noisy} is linear. Our approach, which we term generalized error mitigation (GEM), enables a non-linear relation f_{NL} between these two quantities:

$$p_{\text{noisy}} = f_{\text{NL}}(p_{\text{clean}}). \tag{3}$$

The physical intuition behind GEM is that non-linear relations between different probabilities may exist and, hence, it is desirable to include them in the mitigation process. These relations may originate in the quantum mechanical nature of the underlying state. As a simple example, let us imagine a measurement device that gives 0 and 1, when the qubit is prepared in the $|0\rangle$ and $|1\rangle$ quantum states, respectively, but gives $0.5 + \epsilon$ if the state is in an equal superposition of the two states. This non-linear error cannot be handled by the linear mitigation approach, while it can be fixed with GEM.

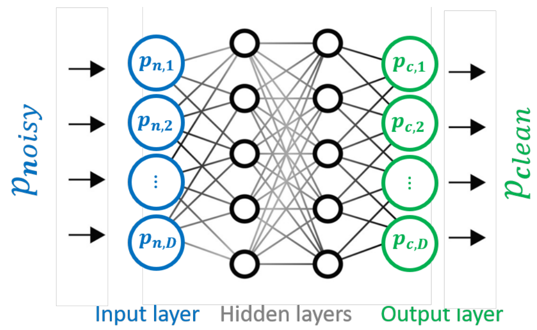

Figure 1: Feed-Forward, fully connected, NN illustration with two hidden layers. Information is processed from left to right: the leftmost layer is the input layer, representing a 2^n-dimensional noisy probability distribution, p_{noisy}, and the rightmost layer is the output layer, representing a 2^n-dimensional clean probability distribution, p_{clean}. With linear activation function and no biases, such a NN with zero hidden layers fully recovers the linear SPAM approach of Eq. (2).

Our approach additionally deviates from standard mitigation schemes in that it aims at *learning* an optimal relation f_{NL} between the clean and the noisy probability distributions, rather than constructing it directly using different sets of physical assumptions. Such a data-driven technique enables a search in a larger solution space and thereby adds another dimension of flexibility to our approach. Moreover, by construction, it bypasses the need to inverse exponentially large

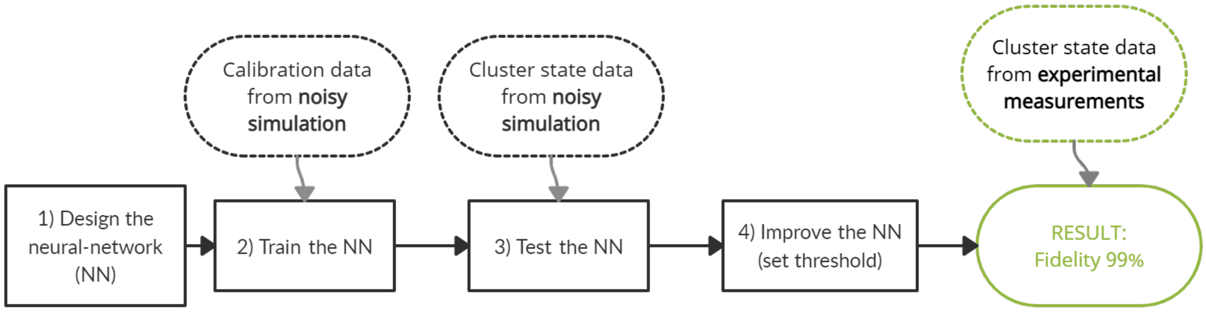

Figure 2: Schematic picture of our solution to the graph-state challenge

matrices. Specifically, we designed a neural network (NN) that learns a (close-to) optimal function $f_{\rm NL}$ based on a given (precomputed) set of K clean and noisy probability distribution pairs $\{(p^1_{\rm noisy}, p^1_{\rm clean}), ..., (p^K_{\rm noisy}, p^K_{\rm clean})\}$. The network is trained, given a noisy probability distribution $p_{\rm noisy}$, to output the corresponding clean probability $p_{\rm clean}$, see Fig. 1. In practice, we train a simple feed-forward, fully dense NN on the Keras software library [7], using the standard mean-square-error (MSE) cost function:

$$f_{cost}(p_{\rm noisy}, p_{\rm clean}) = \sum_j (p_{\rm noisy,j} - p_{\rm clean,j})^2. \tag{4}$$

It is instructive to note that the conventional SPAM mitigation approach of Eq. (2) is recovered from GEM, in the specific case of a NN with a single layer (no hidden layers), linear activation (no thresholds), and no biases. Even in this case, our approach is more robust than the conventional one because it substitutes brute-force (noisy) matrix inversion with NN optimization. We have verified numerically that if one assumes a noisy model of the form of Eq. (1), such a NN fully recovers $A_{\rm SPAM}$ with high precision (the entries of $A_{\rm SPAM}$ can be extracted from the network's internal weights). The GEM approach is therefore a strict generalization of the standard linear SPAM error mitigation scheme.

III. APPLYING GEM ON THE GRAPH STATE CHALLENGE

We now demonstrate the performance of GEM on the graph state challenge. Our approach is schematically shown in Fig. 2. Note that the training step is performed on a data set generated by the QISKIT simulator, while the test is performed on the actual device. The reason for this choice is that, according to the challenge guidelines, the calibration dataset includes only 22 out of

$2^7 = 128$ inputs[2]. As a first, non-trivial extension of Eq. (1), we consider a dense linear NN *with biases*, equivalent to

$$p_{\text{noisy}} = A\, p_{\text{clean}} + b \tag{5}$$

where A is a $2^7 \times 2^7$ matrix and b is a vector of size 2^7. In the optimization procedure, we add the physical assumption that A includes only non-negative terms. In contrast, b can be both positive and negative. As we will see, this will allow us to effectively mitigate error sources. The Keras code for this NN is:

```
model = Sequential([
    Dense(2**num_qubits, activation='linear', use_bias=True,
    kernel_constraint=tf.keras.constraints.NonNeg(),
    input_shape=(128,)),
])
opt = keras.optimizers.SGD(learning_rate=0.4, momentum=0.999)
```

The learning curve of this NN is shown in Fig. 3 and exponentially tends to zero[3]. We stop the training process after 3000 epochs and use the resulting NN to mitigate the experimental measurements of the graph state. This process leads to a fidelity of $77 \pm 3\%$, which is comparable to the CTMP result, of $76 \pm 4\%$.

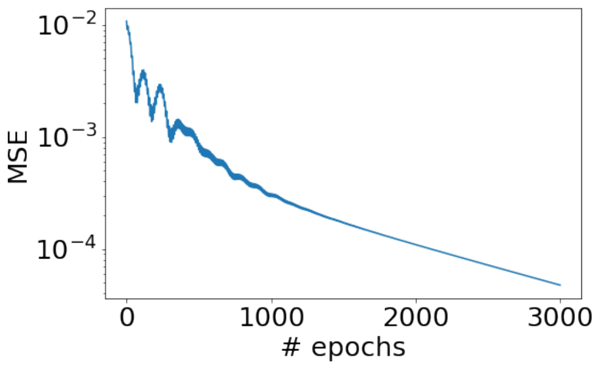

Figure 3: Mean square error (MSE) shown in log scale as a function of the epoch number for the biased NN. The fast decay indicates an efficient learning.

[2] Incidentally, we point out that the label '0000110' is apparently missing from the dataset, which includes all other state with 2 ones and 5 zeros. While this omission is not critical, due to the inclusion of the '1111111 state, this choice breaks the expected permutational symmetry of the problem and may affect the end result of the computation.

[3] The Jupyter notebooks used to train the NN and to improve the graph state can be found online at http://https://github.com/FrimaKal/ibmQuantumAward2021

The success of our biased NN model in improving the fidelity of the graph state looks very surprising. Which physical information have we fed to the NN to allow it to reach the same level of precision as state-of-the-art error mitigation, *without using direct calibration measurements?* As we now explain, the key feature of our NN is the presence of negative biases (indeed, if the biases are set to non-negative, the fidelity drops dramatically): when measuring the stabilizers of a graph state, the clean probability distribution is bimodal: uniform for some measurement outcomes, and zero for all others. In the presence of SPAM errors, the probability of all outcomes becomes finite, but remains small for the states whose clear probability was originally zero. Negative biases reduce the relative weight of outcomes that have a small probability, and effectively increase the weight of outcomes that have a large probability, thereby improving the average fidelity of the graph state.

Using this insight, we can further enhance the effect of the biases, by passing the error-corrected histograms through a constant threshold that dismisses all measurements outcomes that have a low probability. In the framework of NN, this non-linear threshold translates to employing the common rectified linear (ReLU) activation function, where we set a constant threshold of 0.01. The resulting GEM approach with threshold boosts the fidelity of the graph state to the astonishing value of $98.9 \pm 0.7\%$, see Fig. 4.

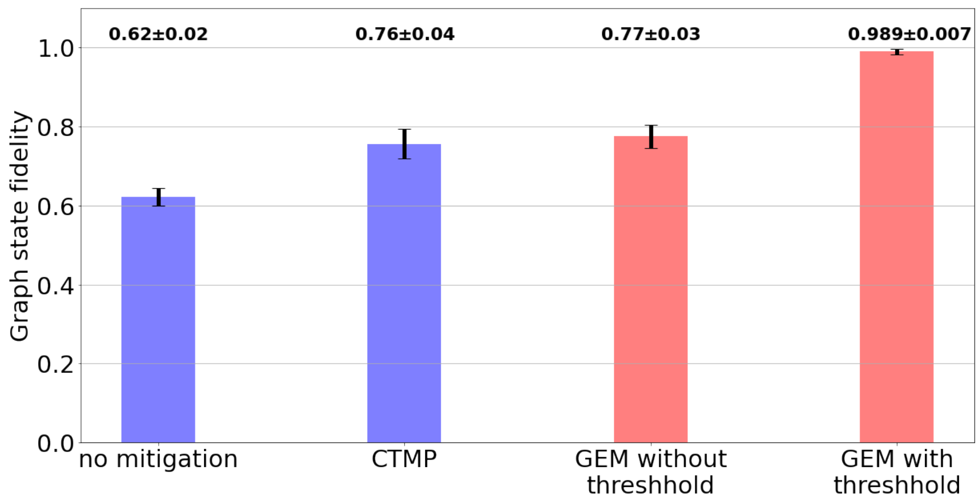

Figure 4: Graph state fidelities based with no mitigation, the second order CTMP, GEM without threshold, and GEM with threshold.

IV. SUMMARY AND CONCLUSION

In summary, in these notes we described a generalized mitigation approach (GEM), aimed at improving the measurement fidelity of NISQ devices. We assumed that the relation between the clean result and the noisy one is given by a generalized nonlinear equation, Eq. (3), which can be effectively learned by a neural network. The case of a one-layered, linear, dense, non-biased NN recovers the linear SPAM approach of Eq. (2), with the advantage that matrix inversion is substituted by a more robust neural network training protocol. We explored the role of biases and thresholds and found that they can improve the fidelity of a 7 qubit graph up to 98.9%. Interestingly, even in the case where SPAM noise is purely linear (such as in the case of the QISKIT noisy simulator), the GEM approach leads to an improved error mitigation. Our physical interpretation is that GEM can offer an effective description of dissipative processes that involve correlations between large number of qubits, which are not captured by the conventional approaches. To further explore the applicability of GEM, we plan to apply this method to other interesting entangled states, and to examine the effect of other, more advanced, types of NNs, by optimizing additional hyper-parameters, such as the number of hidden layers and choice of a non-linear activation functions. GEM has the potential to outperform common mitigation techniques and, if successful, lead to a major improvement in the performance of noisy quantum computations.

---

[1] C. Chamberland *et al.*, Building a fault-tolerant quantum computer using concatenated cat codes (2020), arXiv:2012.04108 [quant-ph].

[2] S. Endo, S. C. Benjamin, and Y. Li, Practical quantum error mitigation for near-future applications, Physical Review X **8**, 031027 (2018).

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

[4] 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 **11**, 1 (2020).

[5] S. Bravyi, S. Sheldon, A. Kandala, D. C. Mckay, and J. M. Gambetta, Mitigating measurement errors in multi-qubit experiments (2020), arXiv:2006.14044 [quant-ph].

[6] B. Nachman and M. R. Geller, Categorizing readout error correlations on near term quantum computers (2021), arXiv:2104.04607 [quant-ph].

[7] F. Chollet *et al.*, Keras (2015), https://github.com/fchollet/keras.