Approximating Stochastic Quantum Noise Through Genetic Programming

Asmar Muqeet[1,2](✉), Shaukat Ali[1,3](✉), and Paolo Arcaini[4]

[1] Simula Research Laboratory, Oslo, Norway
{asmar,shaukat}@simula.no
[2] University of Oslo, Oslo, Norway
[3] Oslo Metropolitan University, Oslo, Norway
arcaini@nii.ac.jp
[4] National Institute of Informatics, Tokyo, Japan

Abstract. Quantum computing's potential for exponential speedups over classical computing has recently sparked considerable interest. However, quantum noise presents a significant obstacle to realizing this potential, compromising computational reliability. Accurate estimation and mitigation of noise are crucial for achieving fault-tolerant quantum computation. While current efforts focus on developing noise models tailored to specific quantum computers, these models often fail to fully capture the complexity of real quantum noise. To this end, we propose an approach that uses genetic programming (GP) to develop expression-based noise models for quantum computers. We represent the quantum noise model as a computational expression, with each function corresponding to a specific aspect of the noise behavior. By function nesting, we create a chain of operations that collectively capture the intricate nature of quantum noise. Through GP, we explore the search space of possible noise model expressions, gradually improving the quality of the solution. We evaluated the approach on five artificial noise models of varying complexity and a real quantum computer. Results show that our approach achieved an error difference of less than 2% in approximating artificial noise models and 15% for a real quantum computer.

Keywords: quantum noise · quantum computing · genetic programming

1 Introduction

In recent years, Quantum Computing (QC) has gained significant attention due to its potential speed advantage over classical computing in solving specific problem classes more efficiently [1]. However, one of the major hurdles to achieving

The work is supported by the Qu-Test project (Project #299827) funded by the Research Council of Norway. S. Ali is also supported by Oslo Metropolitan University's Quantum Hub. P. Arcaini is supported by Engineerable AI Techniques for Practical Applications of High-Quality Machine Learning-based Systems Project (Grant No JPMJMI20B8), JST-Mirai.

G. Jahangirova and F. Khomh (Eds.): SSBSE 2024, LNCS 14767, pp. 56–62, 2024.
https://doi.org/10.1007/978-3-031-64573-0_5

quantum advantage is *quantum noise*, which adversely affects the computation of quantum computers, leading to undesired behaviors. Estimating noise in quantum computers has become crucial, as we are now testing novel quantum error correction and quantum error mitigation methods on current *noisy intermediate-scale quantum* (NISQ) devices [8]. Current efforts in noise estimation focus on identifying *noise models* that accurately represent the noise errors present in current NISQ computers [5,6]. However, existing noise models are simple approximations of real quantum noise [6]. A more detailed noise model that not only represents the noise errors in a quantum computer but also depicts the relationship between different noise errors, qubits, and gate operations would be immensely beneficial. Such a model would greatly enhance the precision of quantum error correction and mitigation techniques. Studies have shown that knowing a more detailed noise model for each qubit and gate operation for a particular quantum computer can significantly enhance the accuracy of quantum error mitigation methods [11,13].

In NISQ computers, noise has various forms, e.g., depolarizing, amplitude dampening, and phase dampening noise [4]. Each qubit and gate operation may experience multiple noise errors, which vary for different qubits and gate operations. In this paper, we propose an approach that uses *genetic programming* (GP) to create expression-based noise models for specific NISQ computers. We represent the quantum noise model as a computational expression consisting of a chain of function calls. Each function adds a specific noise error to particular qubits and gate operations. By nesting these functions within a computational expression, we capture the intricate relationships among various noise errors, qubits, and gate operations for specific quantum computer configurations. The GP process begins with an initial random population of candidate noise model expressions. Through the proposed fitness function and standard evolutionary operators, new candidates are evaluated and generated. This iterative process explores the search space of potential noise model expressions, gradually improving the quality of solutions until satisfactory noise models are obtained. Importantly, by representing noise models as computational expressions, their complex mathematical representation is abstracted, enhancing human comprehension. We evaluate our approach by approximating five artificially created noise models with varying strengths and approximating the noise of one real NISQ computer, IBM-Kyoto. Our approach approximated artificial noise models with less than a 2% difference across all models. Moreover, for IBM-Kyoto, our method outperformed the baseline, with a 15% difference in the noise model approximation compared to 40% for the baseline. In summary, our contributions are (1) the application of GP for approximating quantum noise and creating a more interpretable noise model; (2) an empirical evaluation with five artificial noise models and evaluating the applicability on a real quantum computer.

Related Work. Several efforts have been made to create noise models for NISQ computers. For example, Harper et al. [5] proposed a noise estimation method for quantifying noise in quantum systems and creating correlation matrices that describe the relationship of errors with different qubits. Harper et al. [6] proposed an algorithm for creating a noise model of sparse Pauli noise errors for Clifford quantum circuits. Moreover, Georgopoulos et al. [4] proposed the use of noise estimation circuits to model depolarizing noise error for a given quantum circuit, and Wise et al. [12] used deep learning to transform the output of a quantum circuit to a noisy output, resulting in a neural network-based noise model. However, the major limitation of all these methods is that they are difficult to analyze due to the closed nature of machine learning models and quantum states and, thus, cannot be directly utilized by quantum error mitigation methods.

2 Approach

The most common errors due to quantum noise are depolarizing, amplitude-dampening, and phase-dampening errors [4]. ***Depolarizing Error*** arises from the interaction of a quantum computer with its environment and describes the probabilistic process by which the quantum state of a qubit undergoes random rotations or flips, leading to computation errors [7]. Formally, for one qubit, it is defined as

$$\mathcal{D}(\rho) = (1 - p)\rho + \frac{p}{3}\left(X\rho X + Y\rho Y + Z\rho Z\right) \tag{1}$$

where p is the probability of the error, ρ is the density matrix, and X, Y, Z are the Pauli operations. Different qubits and gate operations can have different probabilities of the depolarizing error [7]. ***Amplitude damping*** error refers to the loss of energy from a quantum system to its environment, while ***Phase damping*** error refers to the loss of information from a quantum system to its environment without dissipating energy. Formally, both are represented as

$$\mathcal{E}(\rho) = E_0\rho E_0^\dagger + E_1\rho E_1^\dagger \tag{2}$$

where ρ is the density matrix, E_0 and E_1 are the Kraus operators. For amplitude damping, E_0 is represented by the matrix $\begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}$, and E_1 by $\begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}$. For phase damping, E_1 is $\begin{bmatrix} 0 & 0 \\ 0 & \sqrt{\gamma} \end{bmatrix}$ and E_0 is the same as amplitude damping. These matrices depict the potential outcomes of the damping process. γ is the damping parameter, which represents the probability of the qubit transitioning from the excited state to the ground state. Our genetic programming (GP)-based approach uses these three noise errors to create an individual representing a noise model.

Individual Representation. GP uses an evolutionary algorithm to evolve computer programs, that are represented and stored as syntax trees. These trees consist of interior nodes representing operations and terminal nodes representing inputs or parameters for these operations. In GP, operation nodes are denoted by a tuple $(\mathit{func}, \mathit{arity})$, where *func* defines the operation and *arity* specifies the number of arguments it can take. The arity of operation nodes in the syntax tree determines the number of child nodes each operation node can have. Our approach employs a variant called Strongly Typed Genetic Programming (STGP) [9]. In STGP, operation nodes additionally define the data type of their arguments and the return type of the operation, represented as (func, arity, $\mathit{argType}_1$, ..., $\mathit{argType}_n$, $\mathit{retType}$).

To construct a noise model for a quantum computer, we define operation nodes in STGP corresponding to common basis gates (such as rx, ry, rz, sx, cx) supported by real quantum computers. Figure 1 illustrates an example individual for STGP. In this figure, *Init* is the root node representing the initialization of an empty noise model. Dp_{RY} is an operation node taking two arguments: the qubit index and the probability p for a depolarizing noise error (see Eq. 1). For instance, Dp_{RY} with arguments $qubit_1$ and 0.3 indicates a depolarizing noise error on qubit 1 and gate operation ry with a probability of 0.3. The return value of each operation node, such as (Dp_{RX}, Dp_{CX}, Ap_{RX}), is the qubit number it acted upon. This enables combining different noise errors on a particular qubit and gate operation to create a more accurate representation of actual noise. For two-qubit gate operations like Dp_{CX}, it takes two qubit indexes as arguments, where the first index is the control qubit and the second index is the target qubit. The return value of two-qubit operation nodes is the index of the target qubit. One advantage of GP is its ability to convert the entire tree representation into a computational expression, defining how noise affects different qubits and gate operations. For instance, the individual in Fig. 1 can be converted into the following computational expression:

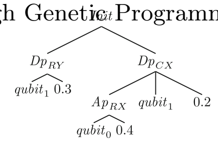

Fig. 1. An example individual for GP

$$(Init\ (Dp_{RY}\ qubit_1\ 0.3)\ (Dp_{CX}\ (Ap_{RX}\ qubit_0\ 0.4)\ qubit_1\ 0.2))$$

By utilizing Eqs. 1 and 2, this expression can be translated into the following noise representation from a qubit perspective:

$$Q_0 = \mathcal{E}(\rho)_{0.3} \qquad\qquad Q_1 = \mathcal{D}(\rho)_{0.3} \otimes (\mathcal{D}(\rho)_{0.2}|Q_0)$$

Fitness Function. Fitness is calculated by averaging the Hellinger distance between multiple quantum circuits. The Hellinger distance is widely used for assessing the output of quantum circuits under noise [2]. The fitness function is $\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sqrt{2}}|\sqrt{P_i}-\sqrt{X_i}|$, where n is the number of circuits used, P_i is the output of the i-th circuit from the real computer, and X_i is the output of the i-th circuit under the noise model. We utilize multiple quantum circuits to evaluate the fitness to avoid optimizing a noise model for a specific quantum circuit.

3 Experiment Design and Result

For our experiment[1], we implemented STGP using the DEAP framework [3], with default settings. By default, DEAP uses a half-and-half policy for initialization, one-point crossover and uniform mutation for genetic variations, and the tournament selection method for choosing the best individuals[2]. For the initialization policy, we set the minimum depth to zero and the maximum depth

[1] https://doi.org/10.5281/zenodo.11198788.
[2] https://deap.readthedocs.io/en/master/api/tools.html.

Table 1. Comparison of 10 runs of genetic programming with baseline

Noise Model	STGP		Random		Statistics	
	F_{avg}	F_{std}	F_{avg}	F_{std}	p_{value}	$\hat{A}_{12}$
Af1	**0.0024**	0.0014	0.44	0.0009	0.0002	Large
Af2	**0.0196**	0.0156	0.38	0.0013	0.0002	Large
Af3	**0.0054**	0.0021	0.21	0.0013	0.0002	Large
Af4	**0.0018**	0.0019	0.29	0.0011	0.0002	Large
Af5	**0.0007**	0.0006	0.20	0.0008	0.0002	Large
IBM-Kyoto	**0.148**	6.6e−05	0.41	0.0006	0.0002	Large

to two, and for the selection method, we set the tournament size to three. For quantum program execution and noise model creation, we utilized IBM's Qiskit framework.

As benchmarks, we created five random noise models with varying complexity, each composed of combinations of three selected noise errors. We evaluated our approach on a real quantum computer from IBM (IBM-Kyoto). Fitness was calculated using three quantum circuits (Amplitude Estimation, Phase Estimation, and Quantum Fourier Transform), computing the average Hellinger distance value, which ranges from 0 to 1, indicating no difference to maximum difference, respectively. To assess effectiveness, we compared our approach with a random baseline across 10 repeated runs. For statistical analysis, we use the Mann-Whitney test and Vargha Delaney $\hat{A}_{12}$ effect size as recommended in [10]. $\hat{A}_{12}$ is interpreted according to [10]: an effect size in the range $(0.34, 0.44]$ and $[0.56, 0.64)$ is considered *Small*; in $(0.29, 0.34]$ and $[0.64, 0.71)$ is considered *Medium*; in $[0, 0.29]$ and $[0.71, 1]$ is considered *Large*. STGP ran for 40 generations with a population size of 300, using default parameter values from DEAP for all other parameters. For a baseline comparison, we generated 12k random individuals, aligning with the generation and population size of STGP.

Result. Table 1 presents the results, where the columns F_{avg} and F_{std} indicate the average, and standard deviation of the best individual over 10 repeated runs. The best average values, highlighted in bold, signify a closer approximation to zero, indicating a better fit of the noise model. Our approach outperformed the baseline random method for all five artificial noise models, with statistically significant improvements indicated by a p_{value} of less than 0.05 and a $\hat{A}_{12}$ statistics with *Large* magnitude. It achieved average fitness values of less than 2%, with consistently low standard deviation across all models. For the real quantum computer (IBM-Kyoto), our approach outperformed the baseline with an average fitness of 15% compared to the 40% for baseline. The results demonstrate our approach's effectiveness in approximating the noise model based on program output for both artificial noise model and real quantum computer. This demonstrates that our approach effectively approximates the noise model of a real quantum computer using the expression representation of GP.

Limitations. While our approach effectively approximates the noise model, there was a notable difference between the results of artificial noise models and the real quantum computer. This is because our method considers only three types of noise errors, while real quantum computers may have additional error types like readout and random unitary errors. To enhance accuracy, we plan to include these additional error types. Additionally, utilizing only three quantum circuits for fitness calculation and using the same circuits for experiment evaluation limits generalization. Therefore, adding more circuits could provide a more comprehensive evaluation of the noise model.

4 Conclusion

We present an approach that uses genetic programming to generate expression-based noise models tailored for NISQ computers. By representing noise as a chain of function calls, our approach creates interpretable noise models that capture different noise errors affecting individual qubits and gate operations of a quantum computer. Our results demonstrate the effectiveness of our approach, achieving an approximation error below 2% for artificially generated noise models and 15% for real quantum computer noise. In the future, we aim to enhance our approach by incorporating additional noise errors and expanding the range of evaluated quantum circuits to gauge its effectiveness further.

References

1. Classical and quantum computing. In: Quantum Information. Springer, New York (2007). https://doi.org/10.1007/978-0-387-36944-0_13
2. Dasgupta, S., Humble, T.S.: Characterizing the reproducibility of noisy quantum circuits. Entropy **24**(2), 244 (2022)
3. Fortin, F.A., De Rainville, F.M., Gardner, M.A.G., Parizeau, M., Gagné, C.: DEAP: evolutionary algorithms made easy. J. Mach. Learn. Res. **13**(1), 2171–2175 (2012)
4. Georgopoulos, K., Emary, C., Zuliani, P.: Modeling and simulating the noisy behavior of near-term quantum computers. Phys. Rev. A **104**, 062432 (2021)
5. Harper, R., Flammia, S.T., Wallman, J.J.: Efficient learning of quantum noise. Nat. Phys. **16**(12), 1184–1188 (2020)
6. Harper, R., Yu, W., Flammia, S.T.: Fast estimation of sparse quantum noise. PRX Quant. **2**, 010322 (2021)
7. King, C.: The capacity of the quantum depolarizing channel. IEEE Trans. Inf. Theor. **49**(1), 221–229 (2003)
8. Martinis, J.M.: Qubit metrology for building a fault-tolerant quantum computer. npj Quant. Inf. **1**(1), 1–3 (2015)
9. Montana, D.J.: Strongly typed genetic programming. Evol. Comput. **3**(2), 199–230 (1995)
10. Tomczak, M., Tomczak, E.: The need to report effect size estimates revisited. An overview of some recommended measures of effect size. Trends Sport Sci. **21**(1), 19–25 (2014)

11. Urbanek, M., Nachman, B., Pascuzzi, V.R., He, A., Bauer, C.W., de Jong, W.A.: Mitigating depolarizing noise on quantum computers with noise-estimation circuits. Phys. Rev. Lett. **127**, 270502 (2021)
12. Wise, D.F., Morton, J.J., Dhomkar, S.: Using deep learning to understand and mitigate the qubit noise environment. PRX Quant. **2**, 010316 (2021)
13. Wood, C.J.: Special session: noise characterization and error mitigation in near-term quantum computers. In: 2020 IEEE 38th International Conference on Computer Design (ICCD), pp. 13–16 (2020)