JOURNAL OF
EMERGING INVESTIGATORS

Article

Error mitigation of quantum teleportation on IBM quantum computers

Richard Chen[1], Xiangdong Chen[2]
[1]Canyon Crest Academy, San Diego, CA
[2]Taiwan Semiconductor Manufacturing Company, San Diego, CA

SUMMARY

Quantum computers can perform computational tasks beyond the capability of classical computers, such as simulating quantum systems in materials science and chemistry. Quantum teleportation is the transfer of quantum information across distances, relying on entangled states generated by quantum computing. It is becoming a more secure way of sending information, but there is noise in the results. We sought to mitigate the error of quantum teleportation which was simulated on IBM cloud quantum computers. We hypothesized that the noise on all IBM quantum computers could be mitigated with noise-mitigation matrices. We created a quantum teleportation circuit which ran with shots of 500, 1000, 5000, and 8192 for four different qubit states. We studied general error trends in each machine and created two types of noise-mitigation matrices: universal and machine-specific. We then compared the mitigated results of both types of matrices. We found that there was noise for every IBM quantum computer during the trials. The universal noise-mitigation matrix for quantum teleportation for the three tested machines decreased the error for most trials and varied between 1%-5% for most cases after mitigation. The machine-specific noise-mitigation matrix mitigated the error of most machines to just 1-2%, which was a dramatic decrease from unmitigated results (varied between 1%-16% across three machines). The error rates for the machine-specific matrices have less variability than the universal mitigation matrix. We concluded that a universal mitigation matrix could be found for the three machines, but the machine-specific noise mitigation matrices were able to achieve more accurate results.

INTRODUCTION

The computing power of classical computer chips is generally governed by Moore's law, which states that computer chips get two times faster every two years. However, in recent years, the increase in classical computing power is slowing down. As classical computers are reaching the limit of computation capabilities, computer scientists are turning more towards quantum computing. By using properties from quantum physics to create efficient algorithms, quantum computers can solve more complex problems that classical computers are not. These new algorithms include innovative simulations and different protocols using quantum properties (1).

Rather than using classical bits, which are 1s and 0s on classical computers, quantum computers use qubits, which can store quantum information. Quantum computers manipulate those qubits to perform complex calculations (2). Most qubits are represented in bra-ket notation, which can also be represented in vector notation. A qubit has two base states. The first is |0>, or $\begin{bmatrix}1\\0\end{bmatrix}$ in vector form. The second one is 1>, or $\begin{bmatrix}0\\1\end{bmatrix}$ in vector form (3, 4).

Logic gates, or logical operations, can be applied to the qubits to perform complex computations. Some gates include the NOT gate and Hadamard gate. The NOT gate is able to invert the inputted value. One version of the NOT gate is the controlled NOT gate. This type of NOT gate is a two-qubit gate. The first qubit is the control while the other qubit is the target. When the control is 0, the target is not changed, but if the control is 1, the target qubit is inverted. The Hadamard gate allows one to put the qubits in superposition, or in a state of both |0> and |1>, rather than just having states of |0> or |1> (5). These basic quantum gates are essential to quantum computing and allow the creation of computational algorithms.

With properties such as superposition and entanglement (two particles linked together and able to affect each other even when far apart), quantum teleportation is possible. In the quantum teleportation protocol, quantum information is transferred between people using both quantum entanglement and classical bit information. In a quantum teleportation scenario, Person 1 and Person 2 first need to go through Person 3 who sends them each half of an entangled qubit pair (**Figure 1**). Person 1 then applies quantum gates to the qubit |**Ψ**> that they want to send and to the entangled

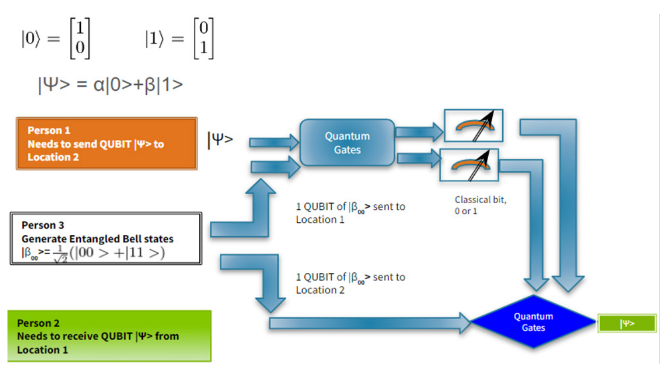

Figure 1: Quantum teleportation protocol. An overview of the quantum teleportation circuit and qubit states in vector notation. The entangled qubits are distributed to the two people and then the states are measured. After measurement, quantum gates are applied again to get the original qubit state.

qubit from Person 3. Person 1 then measures the values of the two qubits and sends the measured values to Person 2 as classical bits. Person 2 then applies quantum gates to their half of the entangled qubit pair from Person 3 based on the classical information they received in order to recover |**Ψ**> (6). Quantum teleportation can transmit information in a more secure manner because those without the entangled qubits are not able to interpret the classical bits that are sent (7). Additionally, no one can clone the information due to the no cloning theorem of quantum physics where the exact copy of an unknown quantum state cannot be made (8). Ergo, if quantum teleportation is utilized in future secure information transfer, the communication of ideas will become virtually un-hackable, and vital information cannot be lost during the transmission.

Qubits are highly volatile. Small fluctuations in energy in the qubit can lead to unintended |0> and |1> qubit state flips and errors in quantum computers (9). The volatility of qubits necessitates their storage at low temperatures to keep the information stable. These conditions are extremely hard to maintain since keeping a system at 0 K is physically impossible. Suboptimal storage conditions can cause large amounts of noise or error when a quantum circuit is run (10). The main goal of this study was to mitigate the error that arises in quantum teleportation. This was done with noise-mitigation matrices, which could mitigate the noise after quantum teleportation on IBM quantum computers, rather than keeping the system closer to 0 K. By using mitigation matrices, we could increase our confidence in having the correct output. These noise-mitigation matrices are vital to quantum computing because they are able to transform raw results to resemble the true outputs that occur under ideal conditions more closely. In the case of quantum teleportation, the use of a noise-mitigation matrix would allow the recipient of the quantum information to better decipher what is sent to them.

In this study, we hypothesized that there is noise on all IBM quantum computers, which can be mitigated with a noise-mitigation matrix. We decided to use the Belem, Santiago, and Lima machines since those computers were all 5 qubit capacity computers with varying compute speeds. We found a specified noise-mitigation matrix for each machine and a universal mitigation matrix for all machines, and they both drastically decreased the error. However, the universal mitigation matrix was not as efficient as the machine-specific matrices.

RESULTS

A quantum teleportation circuit was run on the Belem, Santiago, and Lima IBM cloud quantum computers with same shots of 500, 1000, 5000, and 8192 repeated 3 times for four different qubit states:|0>, |1>, |0> + |1> and |0> - |1>. We used Hadamard bases of |0> + |1> and |0> - |1> in this study to test the viability of the error mitigation matrices when qubits are put in superposition. All the cases had total shots of 44076, where one shot was the teleportation of one qubit through the machine, which was then measured. To find the error, the formula (Desired-Result)/Total was used. The desired outputs of |0> + |1> and |0> - |1> should have outputted the same amount of |0> and |1> since the probabilities of |0> and |1> are equal. After analysis, the Belem Machine had the highest error rates with errors of nearly 16% (**Table 1**). We also saw that the Lima Machine had the lowest error rates (6.59%) before the mitigation matrices were applied to the results. All these quantum machines also favored the |0> qubit state over the |1> qubit state, as the |0> qubit occurred more often in all the machines. This means that the error rates for the |1> qubit state was generally higher than that for the |0> qubit state.

	Belem Machine			Santiago Machine			Lima Machine		
State Sent	Output: 0	Output: 1	Error Rate	Output: 0	Output: 1	Error Rate	Output: 0	Output: 1	Error Rate
|0>	37989	6077	13.79%	41357	2719	6.17%	42557	1519	3.45%
|1>	6971	37105	15.82%	4649	39427	10.55%	23577	20499	6.59%
|0> + |1>	23661	20415	3.68%	23339	20737	2.95%	23577	20499	3.49%
|0> - |1>	24832	19244	6.34%	22642	21434	1.37%	23404	20672	3.10%

Table 1: Summary of machine outputs and error rate. The table shows the total outputs of each machine for every state that was tested, where each case has a total of 44076 shots.

Belem Machine		Santiago Machine		Lima Machine		Universal	
N	*N*$^{-1}$	*N*	*N*$^{-1}$	*N*	*N*$^{-1}$	*N*	*N*$^{-1}$
$\begin{bmatrix} 0.90 & 0.19 \\ 0.11 & 0.82 \end{bmatrix}$	$\begin{bmatrix} 1.16 & -0.26 \\ -0.16 & 1.26 \end{bmatrix}$	$\begin{bmatrix} 0.94 & 0.11 \\ 0.06 & 0.90 \end{bmatrix}$	$\begin{bmatrix} 1.07 & -0.13 \\ -0.07 & 1.13 \end{bmatrix}$	$\begin{bmatrix} 0.98 & 0.08 \\ 0.02 & 0.92 \end{bmatrix}$	$\begin{bmatrix} 1.03 & -0.08 \\ -0.03 & 1.08 \end{bmatrix}$	$\begin{bmatrix} 0.93 & 0.12 \\ 0.07 & 0.88 \end{bmatrix}$	$\begin{bmatrix} 1.09 & -0.15 \\ -0.09 & 1.15 \end{bmatrix}$

Table 2: Noise mitigation matrices. N represents the calculated noise matrix found in the study and N^{-1} represents the mitigation matrix for the corresponding noise matrix.

We first calculated the noise matrix N: $NC_{ideal} = C_{noise}$. Where C_{ideal} was the ideal state and C_{noise} was the state with noise. After finding the noise matrices for each machine, we calculated the noise-mitigation matrices by taking the inverse matrix of N. The noise-mitigation matrices were able to mitigate the error for all four qubit states (**Table 2**). Combining the shots for all three machines, we were also able to calculate a universal mitigation matrix. To reduce the error of the results, we multiplied the noise-mitigation matrix to the original output matrix. We applied the noise-mitigation matrices to the same set of data used to generate the matrices.

After applying the machine-specific mitigation matrix to the Belem Machine, the error rate from that machine was cut down to around less than 5% (**Figure 2**). This was a dramatic decrease from the original error, which was greater than 10%

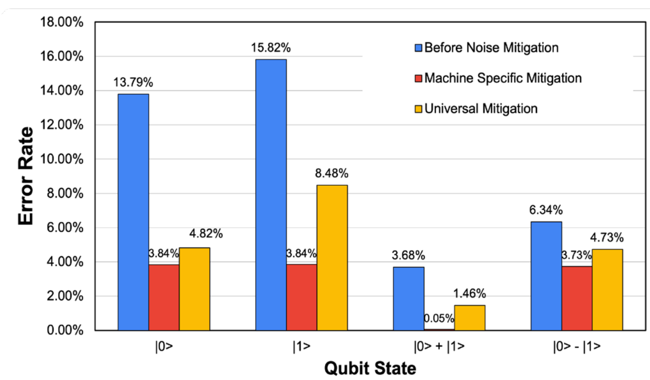

Figure 2: Belem Machine error comparison. Error rates on the Belem Machine before and after machine-specific and universal noise mitigation matrices were applied. Error rates were calculated with raw outputs and also when mitigation matrices were multiplied to the raw outputs.

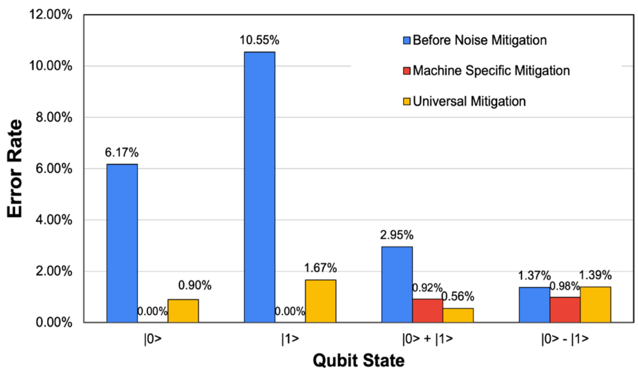

Figure 3: Santiago Machine error comparison. Error rates on the Santiago Machine before and after machine-specific and universal noise mitigation matrices were applied. Error rates were calculated with raw outputs and also when mitigation matrices were multiplied to the raw outputs.

for some cases. The universal mitigation matrix was also able to reduce the error of the Belem Machine to less than 5% in three out of the four tested qubit states.

Similarly, the Santiago Machine also had error rates which were above 5% before error mitigation, and those error rates were cut down to around 0% with the machine-specific noise-mitigation matrices (**Figure 3**). After applying the mitigation matrices to the cases of |0> + |1> and |0> - |1>, the error was also able to drop dramatically. The universal mitigation matrix was able to reduce the error down to less than 2% for all four tested qubit states.

The Lima Machine's error rates also dropped as the machine-specific noise-mitigation matrix was applied to the data (**Figure 4**). However, when the universal mitigation matrix was applied to the raw output, the error was not mitigated as much as the other two machines.

We saw that the Belem Machine had the most error for most qubit states after the mitigation matrix was applied (**Figure 5**). Conversely, the Santiago Machine and the Lima Machine had a much smaller error rate after mitigation. In the end, the universal noise-mitigation matrices were able to mitigate the error on all machines to less than 5% for most cases, which was a substantial decrease from the original output. With the machine-specific noise-mitigation matrices,

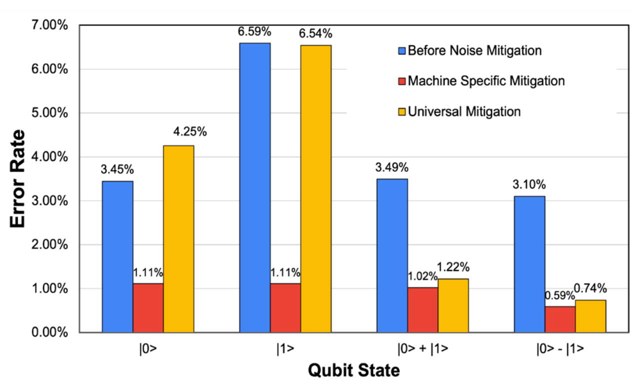

Figure 4: Lima Machine error comparison. Error rates on the Lima Machine before and after machine-specific and universal noise mitigation matrices were applied. Error rates were calculated with raw outputs and also when mitigation matrices were multiplied to the raw outputs.

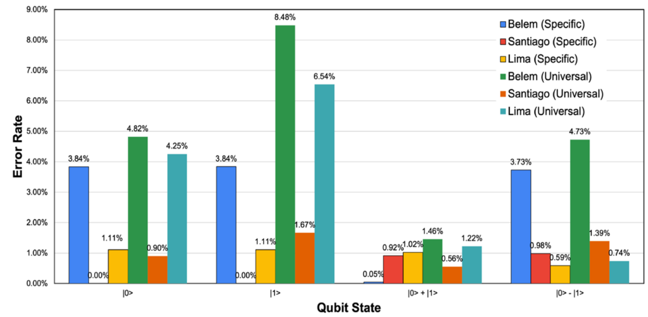

Figure 5: Summary of error rates after mitigation. Error rate comparison of all machines after the mitigation matrices were applied. Error rates were calculated when machine-specific and universal mitigation matrices were multiplied to the raw outputs.

the error was decreased to 1%-2% for most cases.

DISCUSSION

From the initial data, all three of the machines outputted the |0> qubit state more times than the |1> qubit state. This meant that the error for the |1> state was higher than the |0> state. The Lima Machine had the least error initially, and the Belem Machine had the most. This supported the fact that there was noise on every IBM quantum computer. From the data, we saw that the error for all the machines could be mitigated using mitigation matrices. The matrices that were specific to the machines were able to mitigate most of the error to less than 2%. Some of the machine-specific mitigation matrices were also able to lower the error down to around 0%. These machines included the Belem Machine for the case |0> + |1> and the Santiago machine for the cases |0> and |1>.

The universal mitigation matrix was found to speed up the process of error mitigation, since one matrix can be found for multiple machines. The same matrix can be applied to the same machine in the future and still effectively mitigate the error. While the universal mitigation matrix was able to mitigate the error for majority of cases, it was not as effective as the machine specific matrices, and it sometimes increased the error, as seen in the Lima machine graph (**Figure 4**). This was mainly because the universal mitigation matrix was not calibrated specifically for the Lima Machine data points, but it was generally calibrated for all three machines. Those who are studying a variety of machines may find the universal mitigation matrix more effective since they will not need to calibrate a matrix for every machine. However, those who are studying a single machine may be better off using a machine-specific mitigation matrix since it will be more accurate.

Adding more test cases and machines can decrease the error and create a more generalized error mitigation matrix. This matrix can then be applied to more machines in the future and mitigate the errors for more cases. In future studies, other aspects of computer science, such as machine learning, can be applied in order to find a more accurate error mitigation matrix for each machine. Additionally, more quantum states can be included, rather than just the 4 states in this trial, in order to create a more accurate mitigation matrix.

MATERIALS AND METHODS

IBM quantum computers were accessed via the cloud. A quantum circuit was created on the IBM quantum composer,

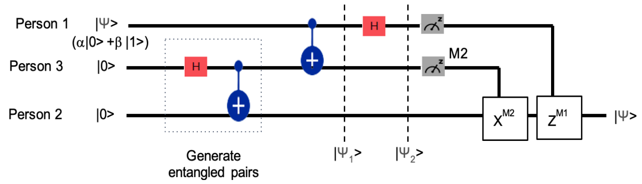

Figure 6: Quantum teleportation circuit. Simulated quantum teleportation circuit that was run on IBM quantum computers. Circuit was created with various logic gates which result in the teleportation of qubits between Person 1 and Person 2.

which simulated a quantum teleportation scenario (**Figure 6**). After creating the circuit, which sent the |0> state, the circuit was run on three different IBM quantum computers. The three machines (Belem, Santiago, and Lima) were all five-qubit machines with varying compute speeds. This meant that each computer could only store a maximum of 5 qubits. The trials were run with shots of 500, 1000, 5000, and 8192. In each shot, one of the qubits in the machine was teleported. The lowest value of 500 was chosen since it was a small enough number to make the data statistically meaningful. The maximum value of 8192 was chosen due to it being the maximum number of shots available on the quantum composer. This process was then repeated with other quantum states such as |1>, |0> + |1>, and |0> - |1>. The Hadamard bases of |0> + |1> and |0> - |1> were used in this study in order to test the viability of the error mitigation matrices when qubits are put in superposition.

The ideal output can be represented as a vector $C_{ideal} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ where α and β were the probabilities of |0> and |1> when there was no noise in the system. The actual output could be represented as $C_{noise} = \begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix}$ where α_1 and β_1 were the actual probabilities of |0> and |1> that were outputted. We first calculated the noise matrix N: $NC_{ideal} = C_{noise}$. After finding the noise matrix N with the output, we found the noise-mitigation matrix N^{-1} by taking the inverse matrix of N with the formula of

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ c & a \end{bmatrix}$$

We then applied the machine-specific and universal mitigation matrices to the recorded outputs from the quantum computers in order to see if the noise after quantum teleportation can be mitigated.

ACKNOWLEDGEMENTS

We would like to thank IBM Quantum Computing for giving open access to quantum computers and the quantum composer.

Received: June 4, 2022
Accepted: October 13, 2022
Published: May 15, 2023

REFERENCES

1. "On 'Quantum Supremacy.'" *IBM Research Blog*, 16 Aug. 2020, www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/.
2. Humble, Travis S., et al. "Quantum Computing Circuits and Devices." *IEEE Design & Test*, vol. 36, no. 3, 2019, pp. 69–94, doi:10.1109/mdat.2019.2907130.
3. Sotelo, Rafael. "Quantum Computing: What, Why, Who." *2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, 2019, doi:10.1109/chilecon47746.2019.8988080.
4. Abdelgaber, Nahed, and Chris Nikolopoulos. "Overview on Quantum Computing and Its Applications in Artificial Intelligence." *2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 2020, doi:10.1109/aike48582.2020.00038.
5. "Qiskit Textbook (Beta)." *Qiskit.org*, qiskit.org/textbook-beta/.
6. Nielsen, Michael A., and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2021.
7. Cacciapuoti, Angela Sara, et al. "When Entanglement Meets Classical Communications: Quantum Teleportation for the Quantum Internet." *IEEE Transactions on Communications*, vol. 68, no. 6, 2020, pp. 3808–3833, doi:10.1109/tcomm.2020.2978071.
8. Gupta, Mayank, and Manisha J Nene. "Quantum Computing: An Entanglement Measurement." *2020 IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI)*, 2020, doi:10.1109/icatmri51801.2020.9398441.
9. Ritter, Mark B. "The Promise and Challenges of Quantum Computing." *2020 International Symposium on VLSI Technology, Systems and Applications (VLSI-TSA)*, 2020, doi:10.1109/vlsi-tsa48913.2020.9203596.
10. "IBM Quantum." *IBM Quantum*, quantum-computing.ibm.com/.