Quantum State Tomography of an Entangled Two-qubit System

Oscar Stommendal

N2402294F@e.ntu.edu.sg

SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

PH3406 Open Quantum Systems

Project Report

April 2, 2025

1 Introduction

The last decade, the field of quantum information and computing has seen rapid development. Quantum computers have showed the potential to outperform classical computers in certain tasks, and many believe that we are on the edge of a quantum revolution [1] [2]. In December 2024, several publicly traded companies within the quantum computing area reached new records in share prices [3]. So, the interest in quantum computing is growing, also outside of academia, and the field is attracting more and more attention.

However, the technology is still in its first phase, and faces many challenges [4]. These include the need for error correction due to noise, upscaling of the number qubits and hardware development. In this report, we will focus on *quantum state tomography* (QST). This is a method used to characterize the state of a quantum system, which has several important applications within quantum computing, for example in error correction.

2 Theory

This section provides the necessary background to understand the rest of the report. We will start by introducing the basic concepts of quantum computing, followed by a more detailed explanation of QST.

2.1 Introduction to Open Quantum Systems

When studying closed systems within quantum mechanics, we describe the system by a state vector $|\psi\rangle$, which evolves in time according to the Schrödinger equation,

$$i\hbar\frac{d}{dt}|\psi\rangle = \hat{H}|\psi\rangle . \tag{1}$$

Here, $\hat{H}$ is the Hamiltonian, which describes the energy of the system. However in reality, no system is completely isolated, and we must consider the interaction with the environment. This leads to the concept of open quantum systems, where the system state is described by a density matrix $\hat{\rho}$ [5]. This matrix is essentially a generalization of the state vector, and contains the same information, with the difference that it can also describe mixed states. Mixed states are states that are not pure (like the ones in closed systems), but rather a statistical mixture of pure states. The density matrix is defined as

$$\hat{\rho} = \sum_i p_i |\psi_i\rangle \langle\psi_i| , \tag{2}$$

where p_i is the probability of the system being in the state $|\psi_i\rangle$. When performing QST, we are essentially trying to reconstruct this density matrix from measurements of the system. An appropriate measure to test the result of the tomography is the so called *fidelity*, which is a measure of how close the reconstructed state is to the ideal state $\hat{\rho}$. The fidelity is defined as

$$\mathcal{F}(\hat{\rho}, \hat{\rho}_{\text{exp}}) = \text{Tr}\left(\sqrt{\sqrt{\hat{\rho}}\hat{\rho}_{\text{exp}}\sqrt{\hat{\rho}}}\right) , \tag{3}$$

where $\hat{\rho}_{\text{exp}}$ is the reconstructed state and Tr() the trace [5]. If $\mathcal{F}$ is equal to 1, the reconstructed state is identical to the ideal state, and if it is 0, the states are orthogonal.

2.2 Quantum State Tomography

In quantum mechanics, the state of the considered system is usually of highest interest, as this contains all the system information. However, due to the nature of quantum mechanics, once we measure the outcome of a system, the state collapses to one of the possible outcomes [5]. So, the root of the problem considered in this project is that no measurement can differentiate between non-orthogonal states like $|0\rangle$ and $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. This means that we cannot measure the state directly, but rather have to reconstruct it from repeated measurements of the system. This is where QST comes into play.

The basic idea of QST is to prepare the system in the same way many times and measure the outcome. The density matrix $\hat{\rho}$ is then reconstructed from the measurement data. In principle, the measurements are done using the Pauli operators, which together with the identity operator form a basis for the space of operators [5]. By measuring the outcome of these different basis operators many times, the density matrix can be reconstructed. For example, in the case of one qubit, $\hat{\rho}$ can be written as

$$\hat{\rho} = \frac{1}{2}\left(\mathbb{I} + \sum_{i=1}^{3} \langle\sigma_i\rangle\sigma_i\right), \tag{4}$$

where $\mathbb{I}$ is the identity matrix, σ_i are the Pauli operators and $\langle\sigma_i\rangle$ are the expectation values of the Pauli operators. This can then be generalized to more qubits.

3 Methodology

The foundation of the method in this project is taken from Qiskit's experiment manual on QST. Some adjustments have been made to the code, in order to fit the specific goals here. The code is written in `Python` and uses the `Qiskit` library for creating and transpiling circuits, the `Qiskit Aer` library for simulating the problems with noise and the `Qiskit Experiments` library for performing the state tomography [6] [7] [8]. For reference, the complete source code is available on GitHub.

3.1 Quantum Circuit

The quantum state used in this project is an entangled two-qubit system, initialized using the circuit in Figure 1 below.

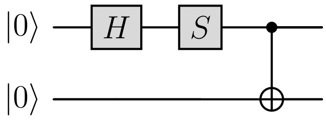

Figure 1: The quantum circuit used to initialize an entangled two-qubit system. Note that the input state is $|00\rangle$. This figure was created using the `quantikz` package [9].

The circuit consists of a Hadamard gate, an S gate and a CNOT gate. The Hadamard gate puts the first qubit in a superposition state between $|0\rangle$ and $|1\rangle$ and the S gate gives an extra phase of $\pi/2$ to the state $|1\rangle$. Lastly, the CNOT with target on the second qubit will change this from 0 to 1 if the first qubit is in state $|1\rangle$. Thus, with input state $|00\rangle$ on an ideal quantum computer, this circuit will produce the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + i\,|11\rangle) \equiv \frac{1}{2}\begin{bmatrix}1 & 0 & 0 & -i\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ i & 0 & 0 & 1\end{bmatrix} = \hat{\rho}. \tag{5}$$

In Qiskit, this is easily implemented according to the code snippet below.

Code Snippet, Quantum Circuit

```
# State preparation circuit
qc = qiskit.QuantumCircuit(2, 2) # Create a quantum circuit with 2 qubits
↪  (the other argument is for the classical register, not used in the
↪  state tomography)
qc.h(0) # Apply a Hadamard gate on the first qubit
qc.s(0) # Apply an S gate on the first qubit
qc.cx(0, 1) # Apply a CNOT gate between the first and second qubit
```

3.2 Noise Modelling

In order to get more realistic results, some experiments will be run with noise, as this is a common problem in quantum computing. Noise can come from several sources, such as the environment or the hardware, and can affect the qubits in different ways [5]. In this project, we focus on two noise models: *depolarizing noise* and *measurement noise*. The depolarizing noise basically replaces the original state with the maximally mixed state, $\hat{\rho}_M = \alpha\mathbb{I}$ to some degree, as we will see later [5]. The measurement noise, on the other hand, is a type of noise that occurs when the measurement of the qubit is not perfect, and the result is not always correct. These two models are especially interesting in QST as they affect the resulting state, which is what we are trying to reconstruct. Below follows a brief explanation of how the noise models were implemented in Qiskit together with some theoretical background on the models.

3.2.1 Depolarizing Noise

The depolarizing noise is added by applying a depolarizing error (or channel) to each gate in the circuit. The code snippet below shows how to implement this in Qiskit.

Code Snippet, Depolarizing Noise Model

```
p = 0.1 # Error probability
error_1 = depolarizing_error(p, 1) # 1-qubit gate error
error_2 = depolarizing_error(p, 2) # 2-qubit gate error

# Add errors to noise model
noise_model = NoiseModel() # Create a noise model
noise_model.add_all_qubit_quantum_error(error_1, ['h', 's']) # Add 1-qubit
↪  gate errors to the noise model
noise_model.add_all_qubit_quantum_error(error_2, ['cx']) # Add 2-qubit gate
↪  errors to the noise model

noise_backend = AerSimulator(noise_model=noise_model) # Noise simulator
trans_qc = transpile(qc, noise_backend) # Transpile the circuit to the
↪  noise model (in order to apply the noise)
```

Here, the error rate is given by a parameter p, which is the probability of an error occurring [5]. Mathematically, the state after the depolarizing error is given by

$$\mathcal{D}(\hat{\rho}) = \frac{p\,\mathbb{I}_d}{d} + (1-p)\hat{\rho}, \tag{6}$$

where $\mathbb{I}_d$ is the identity matrix in dimension d. Studying the code example above, for the Hadamard and S gate, $d = 2$ and for the CNOT gate, $d = 4$. This means that the state is mapped to a linear combination of the original state and the maximally mixed state $\hat{\rho}_{mixed} = {}^{\mathbb{I}}/_{d}$. During the experiments, the value $p = 0.1$ was used.

3.2.2 Measurement Noise

The measurement noise is added by creating a readout error object, which is added to the noise model. This error object is a matrix that describes the probability of the measurement being correct or incorrect. For one qubit, this is the 2 × 2 matrix

$$\epsilon = \begin{bmatrix} \mathcal{P}(0|0) & \mathcal{P}(0|1) \\ \mathcal{P}(1|0) & \mathcal{P}(1|1) \end{bmatrix}. \tag{7}$$

So, the diagonal elements are the probabilities of the correct measurement, and the off-diagonal elements are the probabilities of incorrect measurements. In the experiments, a probability of $\mathcal{P}(0|0) = \mathcal{P}(1|1) = 0.9$ for the correct measurement and $\mathcal{P}(0|1) = \mathcal{P}(1|0) = 0.1$ for the incorrect measurement was used (for a more fair comparison, this is the same error probability as for the depolarizing noise). The code snippet below shows how to implement this in Qiskit. One could have added this error when measuring both qubits simultaneously, but for simplicity, we consider the error on each qubit separately.

Code Snippet, Measurement Noise Model

```
probs = np.zeros((Nq, Nq)) # Create an error probability matrix, with Nq being the number of qubits
for i in range(0, Nq):
    for j in range(0, Nq):
        if i == j:
            probs[i, j] = 0.9 # Probability of correct measurement
        else:
            probs[i, j] = 0.1 # Probability of incorrect measurement

readout_error = ReadoutError(probs) # Create a readout error object

noise_model = NoiseModel() # Create a noise model
noise_model.add_all_qubit_readout_error(readout_error) # Add the error to both qubits

noise_backend = AerSimulator(noise_model=noise_model) # Noise simulator
trans_qc = transpile(qc_ghz, noise_backend) # Apply the noise
```

3.3 Quantum State Tomography

When performing QST, we use the `StateTomography` class in the `Qiskit Experiments` library [8]. This takes the quantum circuit as input, and returns a result object that for example contains the reconstructed density matrix and the fidelity between the ideal and the reconstructed state. The code snippet below shows the implementation.

Code Snippet, Quantum State Tomography

```
1       backend, qc = get_QST_qc(Nq, noise=noise, noise_choice=noise_choice) # Get
        ↪  the circuit and backend
2
3       qst = StateTomography(qc) # Create a state tomography object
4       qst_data = qst.run(backend=backend, shots=counts).block_for_results() # Run
        ↪  the state tomography
```

In the code above, `get_QST_qc()` is a function that returns the quantum circuit and the backend, depending on the noise model chosen. If the experiment is performed without noise, specified by setting the variable `noise=False`, the default `AerSimulator()` backend is used. The `noise_choice` parameter is used to choose between the two noise models, and the `counts` parameter is the number of measurements taken. During the experiments, a fixed value of `counts` $= 10\,000$ were used. For the input system $|\psi\rangle$ in Equation 5, the tomography was performed 4 times – one time without noise, one with depolarizing noise, one with measurement noise and one with both noise models.

3.3.1 Alternative Tomography Fitters and Parallel Experiments

Except for the experiments described above, some additional experiments were performed. Firstly, one was done to compare the performance of different tomography fitters, which are the algorithms used to reconstruct the density matrix $\hat{\rho}$ from the measurement data. The fitters used in this project are `linear_inversion` and `cvxpy_gaussian_lstsq`. The first one is a least squares fitter, while the second one is a more advanced fitter (of the same type) that uses constrains on $\hat{\rho}$ to ensure that it is positive semidefinite (i.e. that it has non-negative eigenvalues) [8]. The first fitter also ensures this, however by executing some extra calculations. Using `Qiskit`, the fitter can be set by adding the line

```
qst.analysis.set_options(fitter='cvxpy_gaussian_lstsq')
```

between line 3 and 4 in the code above. The fitters were then tested on $|\psi\rangle$ in Equation 5 subject to both depolarizing and measurement noise implemented as before.

Secondly, a so called parallel experiment was performed. As the name suggests, this is an experiment which runs several sub-experiments in parallel. As QST gets more time-consuming with increasing number of qubits, this could be a way to save time [5]. In this case, the experiment was done on a circuit with 5 qubits, shown in Figure 2 below.

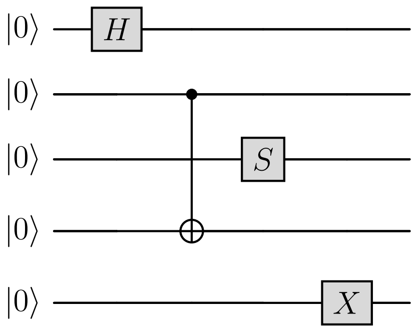

Figure 2: The 5-qubit quantum circuit used in the parallel experiments. Note that the input state is $|00000\rangle$. This figure was created using the `quantikz` package [9].

The parallel experiment was done with depolarizing noise, implemented as before. The code snippet below shows how to implement the parallel experiment (with the `backend` variable being the depolarizing noise backend). Similar to before, a number of `counts`= 10 000 shots were used together with the `linear_inversion` fitter. The reconstructed density matrices were then plotted and compared to the ideal state via the fidelity.

Code Snippet, Parallel Experiments

```
gates = [qiskit.circuit.library.HGate(), qiskit.circuit.library.SGate(), qiskit.circuit.library.CXGate(), qiskit.circuit.library.XGate()] # Gates in the circuit

subexps = [StateTomography(gates[0], physical_qubits=(0, )), StateTomography(gates[1], physical_qubits=(2, )), StateTomography(gates[2], physical_qubits=(1, 3)), StateTomography(gates[3], physical_qubits=(4, ))] # QST sub-experiments

# Create and run a parallel experiment
parexp = ParallelExperiment(subexps, flatten_results=False)
pardata = parexp.run(backend, shots=10000).block_for_results()
```

4 Results

Below follows the results from the experiments with and without noise for the QST with different fitters, along with the parallel experiments. Note that, unlike `Qiskit's` experimental manual on this topic, the axis labels also include the braket notation.

4.1 Without Noise

Figure 3 shows the reconstructed density matrix $\hat{\rho}$ of the entangled two-qubit system after performing the QST without noise. The fidelity between the ideal and the reconstructed state was $\mathcal{F} = 0.9966$.

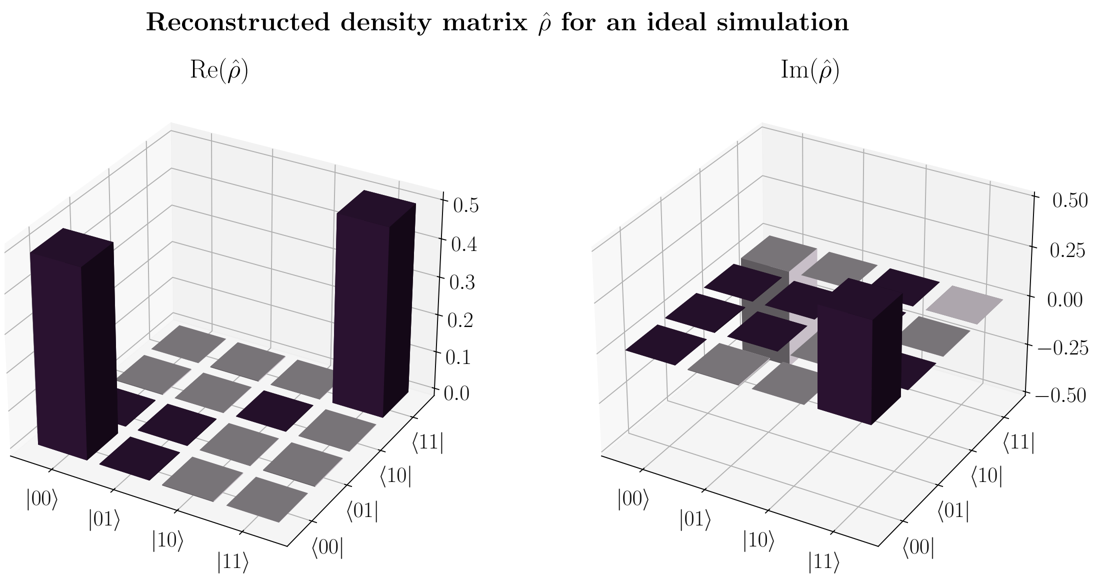

Figure 3: The reconstructed density matrix $\hat{\rho}$ of the entangled two-qubit system after performing the quantum state tomography without noise.

4.2 With Noise

Table 1 shows the fidelity and fitting time after performing the QST with depolarizing noise, measurement noise and both noise models for different tomography fitters. The corresponding reconstructed density matrices $\hat{\rho}$ with the different noise models are shown in Figures 4, 5 and 6, respectively. Note that the fidelity is significantly lower when noise is added to the system and that no figures are shown for the `cvxpy_gaussian_lstsq` fitter, as we essentially only need the fidelity to compare the different fitters.

Table 1: The fidelity $\mathcal{F}(\hat{\rho}, \hat{\rho}_{\text{exp}})$ and fitting time after performing the quantum state tomography with depolarizing noise, measurement noise and both noise models for different tomography fitters.

Noise Model(s)	**Tomography Fitter**	**Fidelity $\mathcal{F}(\hat{\rho}, \hat{\rho}_{\text{exp}})$**	**Fitting Time**
Depolarizing	`linear_inversion`	0.7689	0.0062
Measurement	`linear_inversion`	0.7308	0.0046
Both	`linear_inversion`	0.5777	0.0056
Depolarizing	`cvxpy_gaussian_lstsq`	0.7726	0.0355
Measurement	`cvxpy_gaussian_lstsq`	0.7330	0.0239

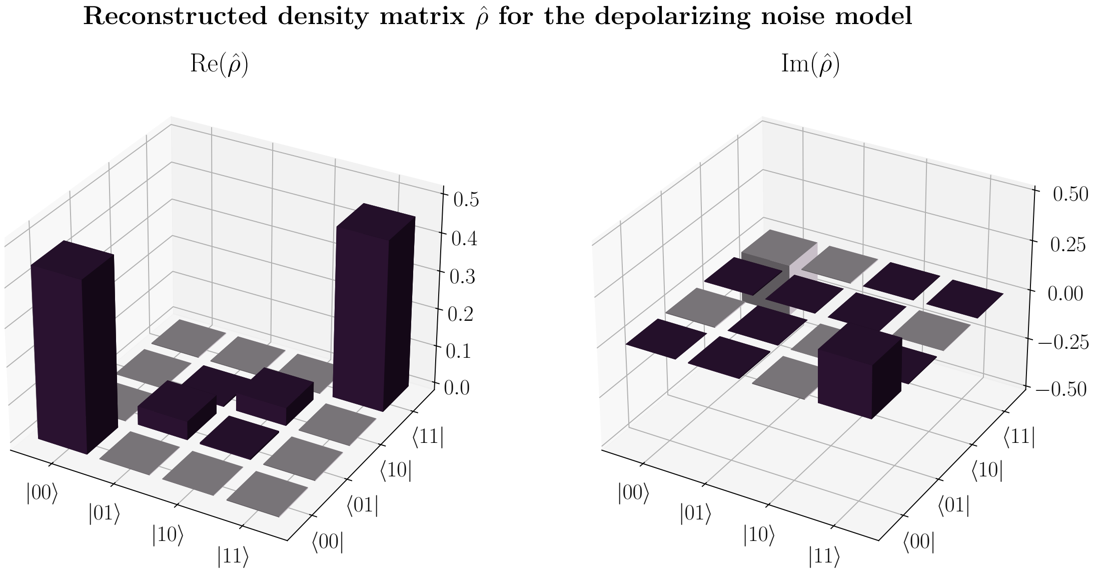

Figure 4: The reconstructed density matrix $\hat{\rho}$ of the entangled two-qubit system after performing the quantum state tomography with depolarizing noise. In this case, the `linear_inversion` fitter was used.

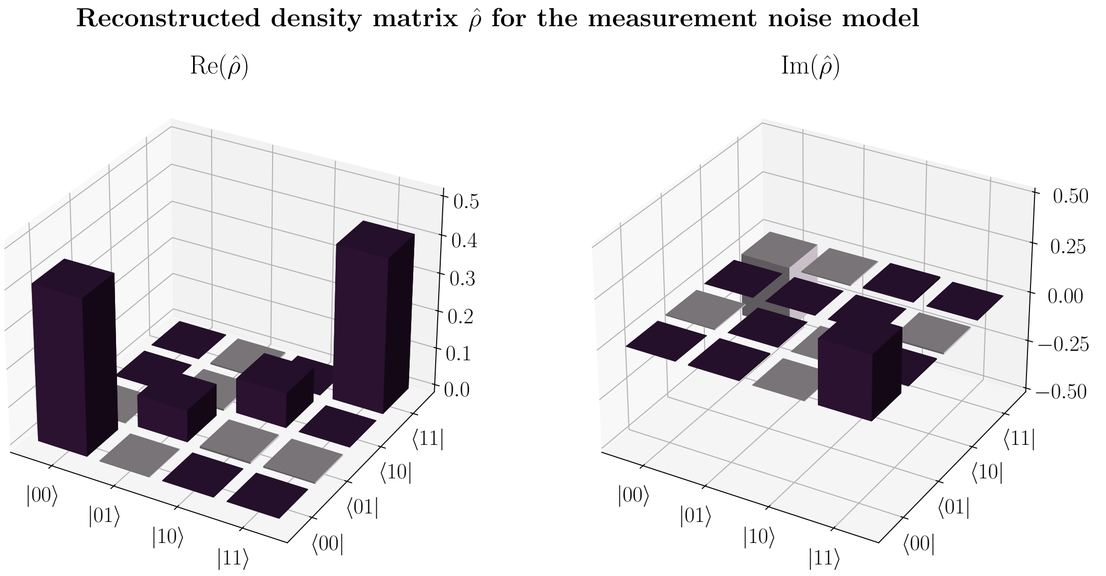

Figure 5: The reconstructed density matrix $\hat{\rho}$ of the entangled two-qubit system after performing the quantum state tomography with measurement noise. In this case, the `linear_inversion` fitter was used.

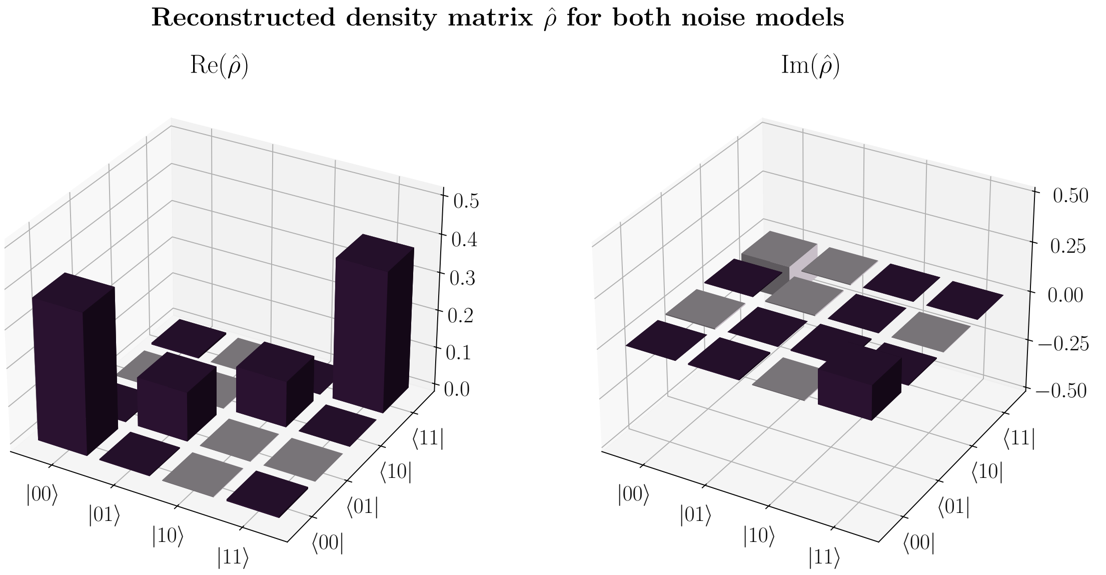

Figure 6: The reconstructed density matrix $\hat{\rho}$ of the entangled two-qubit system after performing the quantum state tomography with both depolarizing and measurement noise. In this case, the `linear_inversion` fitter was used.

4.3 Parallel Experiments

Table 2 shows the fidelity after performing QST with depolarizing noise on each qubit for the parallel experiments. The corresponding reconstructed density matrices $\hat{\rho}$ are shown in Figures 7, 8, 9 and 10 in Appendix A.

Table 2: The fidelity after performing the quantum state tomography with depolarizing noise for the parallel experiments.

Qubit(s)	**Gate**	**Fidelity $\mathcal{F}(\hat{\rho}, \hat{\rho}_{\mathbf{exp}})$**
1	$-\boxed{H}-$	0.9065
2 & 4	$-\boxed{CX}-$	0.9258
3	$-\boxed{S}-$	0.9489
5	$-\boxed{X}-$	0.9476

5 Discussion

Firstly, we notice a substantial decrease of the fidelity when noise is introduced into the system, decreasing significantly from nearly 1 in the noiseless case. This is of course expected, as the noise models change the state of the qubits, which makes the reconstruction harder. For both fitters, we see that the fidelity is slightly lower for the measurement noise model compared to the depolarizing noise model. A possible explanation for this is that the measurement noise model affects the actual measurements, which is the primary source for the tomography algorithm when it tries to reconstruct the density matrix. The depolarizing noise model, on the other hand, only affects the state of the qubits, which is not as crucial for the algorithm. Furthermore, the fidelity is lowest when both noise models are used, which is also expected, as this yields more general noise.

Moreover, the affect of the noise models can be seen clearly when comparing the ideal state's density matrix in Figure 3 with Figures 4, 5 and 6. As expected, both the depolarizing and measurement noise model significantly increases the diagonal terms in the density matrix. For the depolarizing noise model, this can be understood when studying Equation 6 which describes the state after depolarizing noise. Here, the first term is proportional to the identity matrix, i.e. it increases the diagonal terms, while the second term reduces the entries of the original state. For the measurement noise model, it is easy to show why the diagonal terms increase. Our ideal output state in Equation 5 will yield two possible noisy outcome states,

$$\begin{cases} |\psi\rangle_1' = \frac{1}{\sqrt{2}}\left(|10\rangle + i\,|01\rangle\right), & \text{if measurement noise on the 1st qubit} \\ |\psi\rangle_2' = \frac{1}{\sqrt{2}}\left(|01\rangle + i\,|10\rangle\right), & \text{if measurement noise on the 2nd qubit.} \end{cases} \tag{8}$$

After some simple math, one can derive that the corresponding density matrices increase the real parts of the two diagonal terms associated with $|01\rangle$ and $|10\rangle$. In the same way, one will discover that the imaginary parts of these density matrices will cancel out each other, as the result also shows. As a final note, the appearance of the density matrix with both noise models is no surprise, as that is a combination of the two previous cases.

As for the different tomography fitters, we see that the fidelity is slightly higher for the `cvxpy_gaussian_lstsq` fitter compared to the `linear_inversion` fitter. This is likely due to the more advanced nature of the `cvxpy_gaussian_lstsq` fitter, as mentioned earlier. On the other hand, the fitting time is also higher for the `cvxpy_gaussian_lstsq` fitter, which is expected for the same reason as above.

Lastly, in the parallel experiments we see that the fidelity is higher in comparison to the experiments with the entangled two-qubit system. This is likely due to the fact that the parallel experiments are performed on separate qubits, which makes the reconstruction easier. As for the density matrices displayed in Appendix A, we see a smaller noise impact than in the case of the entangled two-qubit system. Since the gate on each qubit in some cases has little to none effect on the input state, the noise model has less impact on the reconstructed density matrix. For example, the CNOT gate on qubits 2 and 4 only flips the second qubit if the first qubit is in state $|1\rangle$. Since the input state is $|00\rangle$, the gate has no effect on the state, and the noise impact is negligible.

6 Conclusion

In this project, we have studied the concept of quantum state tomography on an entangled two-qubit system using different approaches implemented via `Qiskit`. In the simulations, two different noise models were used to increases the realism of the experiments. In both cases, the fidelity of the reconstructed state decreased significantly, which is expected. Also, the impact of the noise models on the density matrix was clearly visible and in line with the theoretical expectations.

When comparing different tomography fitters, the fidelity was slightly higher for the `cvxpy_gaussian_lstsq` fitter compared to the `linear_inversion` fitter. However, this came at the cost of a longer fitting time. Lastly, the parallel experiments showed that the fidelity was higher compared to the entangled two-qubit system, which is likely due to the fact that the parallel experiments were performed on separate qubits.

In conclusion, QST is a powerful tool for reconstructing the density matrix of a quantum system. This is essential for verifying quantum states in many applications, including quantum computing and quantum communication. Moreover, parallel experiments can be a useful tool to save time when performing QST on multiple qubits. However, in all cases, it is important to consider the impact of noise and the tomography fitters used, as they can have a significant impact on the results.

References

[1] B. Baker, "Quantum computers shown to outperform classical computers," *IoT World Today*, Oct. 2024. [Online]. Available: https://www.iotworldtoday.com/quantum/quantum-computers-shown-to-outperform-classical-computers.

[2] J. Carson, "The next quantum revolution," *MIT Spectrum*, Spring 2024. [Online]. Available: https://betterworld.mit.edu/spectrum/issues/2024-spring/the-next-quantum-revolution/.

[3] A. Bylund, "Why quantum computing stocks skyrocketed in december," *The Motley Fool*, Jan. 2025. [Online]. Available: https://www.fool.com/investing/2025/01/04/why-quantum-computing-stocks-skyrocketed-in-decemb/.

[4] M. Swayne, "What are the remaining challenges of quantum computing?" *The Quantum Insider*, Mar. 2023. [Online]. Available: https://thequantuminsider.com/2023/03/24/quantum-computing-challenges/.

[5] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary Edition. Cambridge University Press, 2010, ISBN: 978-1107002173.

[6] IBM, *Ibm quantum documentation*, Accessed: 2025-03-16, 2025. [Online]. Available: https://docs.quantum.ibm.com.

[7] Qiskit Development Team, *Qiskit aer documentation*, Accessed: 2025-03-16, 2025. [Online]. Available: https://qiskit.github.io/qiskit-aer/index.html.

[8] Qiskit Development Team, *Qiskit experiments documentation*, Accessed: 2025-03-16, 2025. [Online]. Available: https://qiskit-community.github.io/qiskit-experiments/index.html.

[9] A. Kay, *Tutorial on the quantikz package*, 2023. arXiv: 1809.03842 [quant-ph]. [Online]. Available: https://arxiv.org/abs/1809.03842.

A Additional Results

Figures 7, 8, 9 and 10 below show the reconstructed density matrix $\hat{\rho}$ of the entangled two-qubit system after performing QST with depolarizing noise for qubits 1, 2 & 4, 3 and 5, respectively, during the parallel experiments. The fidelities are shown in Table 2.

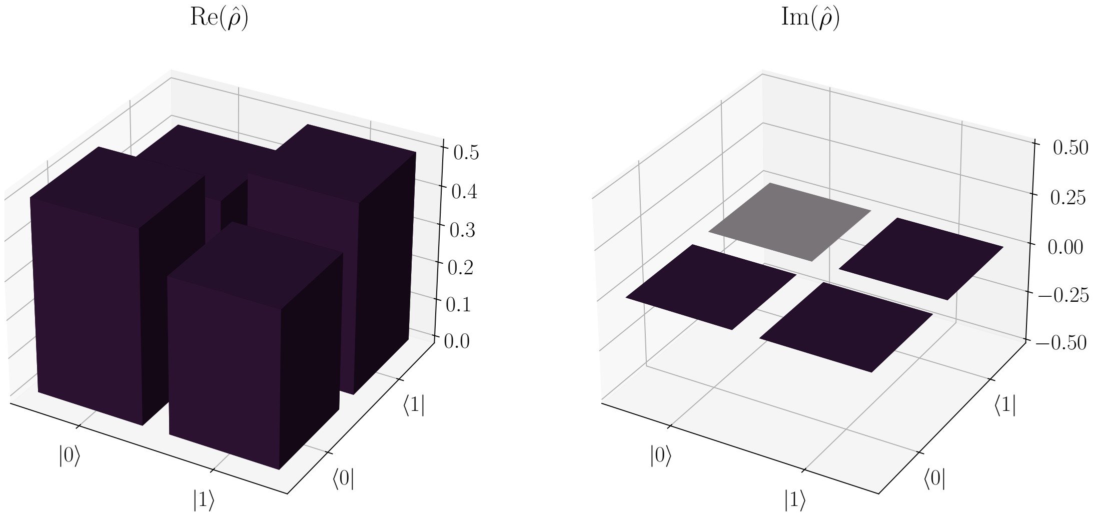

Figure 7: The reconstructed density matrix $\hat{\rho}$ of the entangled two-qubit system after performing QST on the first qubit with depolarizing noise in the parallel experiment.

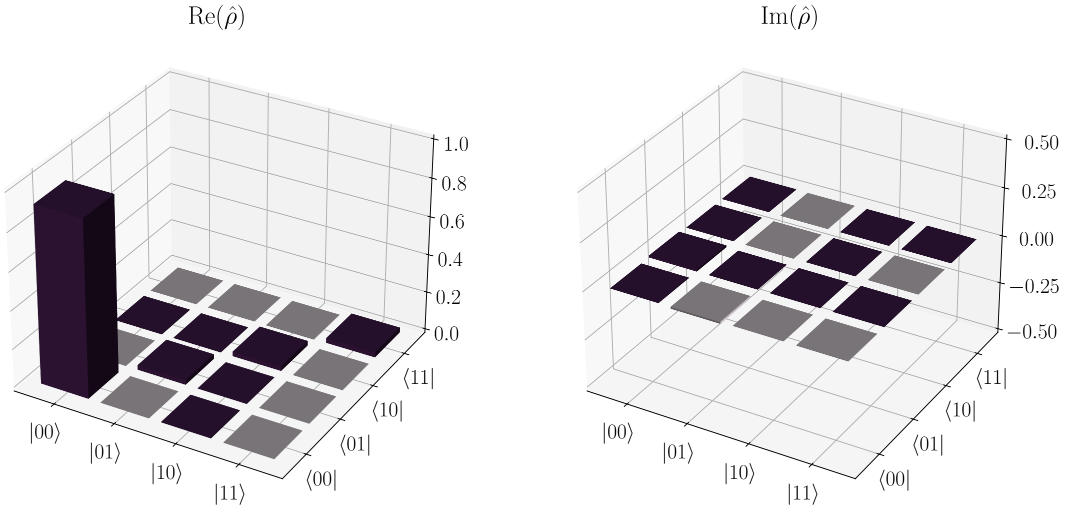

Figure 8: The reconstructed density matrix $\hat{\rho}$ of the entangled two-qubit system after performing QST on the second and fourth qubit with depolarizing noise in the parallel experiment.

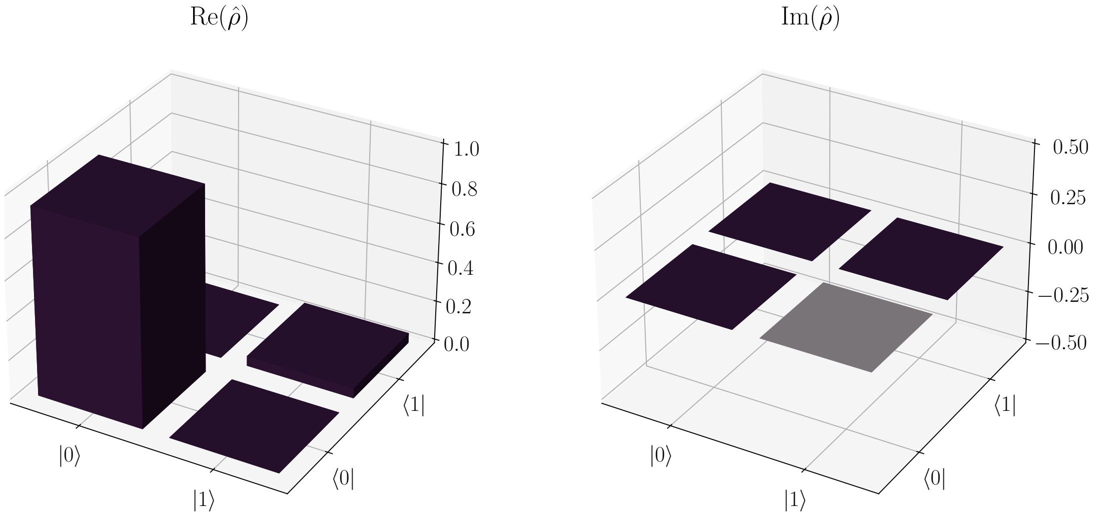

Figure 9: The reconstructed density matrix $\hat{\rho}$ of the entangled two-qubit system after performing QST on the third qubit with depolarizing noise in the parallel experiment.

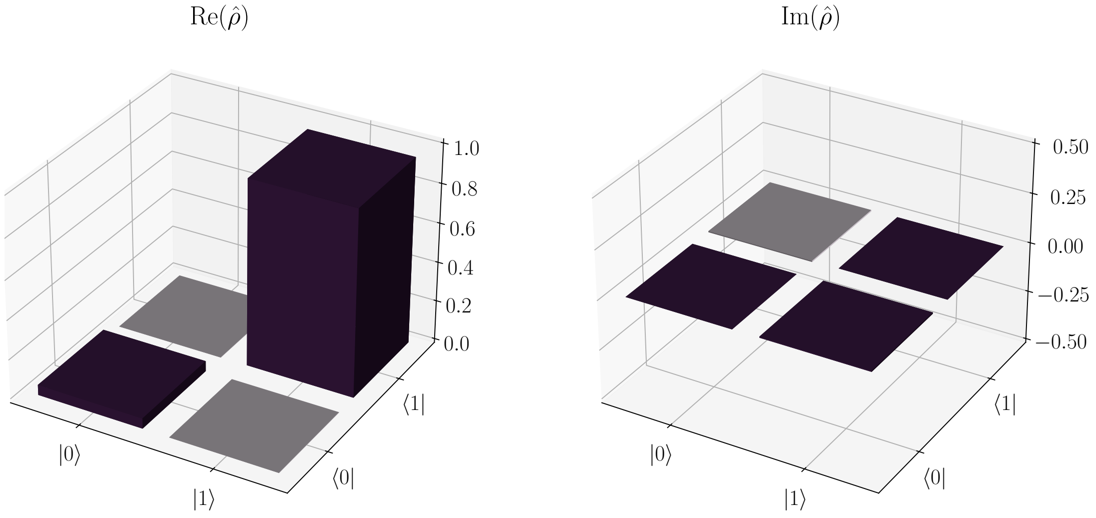

Figure 10: The reconstructed density matrix $\hat{\rho}$ of the entangled two-qubit system after performing QST on the fifth qubit with depolarizing noise in the parallel experiment.

B Notes on AI Assistance

The text in this report is my own work. However, some parts of the project source code was generated with the help of GitHub Copilot. This includes the function for plotting the density matrix and basic functions for saving and loading data. Although, I have reviewed and modified the generated code to fit my goals.