Gate Set Tomography Under the Influence of Time-Correlated Noise

Victor van Wieringen,
Delft University of Technology
4721500

May 7th 2024

Abstract

This thesis investigates the robustness of Gate Set Tomography (GST) under the influence of time-correlated (non-Markovian) noise. GST is a widely used protocol for characterizing quantum gates, yet its efficacy traditionally relies on the assumption of Markovian noise, where errors are memoryless. This research challenges this assumption by introducing non-Markovian noise into simulated GST experiments using a custom-developed Python library. The study compares baseline GST performance under Markovian noise with GST results under various non-Markovian noise conditions. The findings highlight significant discrepancies in GST's error reporting and reveal a distinct trade-off between GST's accuracy and consistency in practical, non-Markovian settings. This work contributes to the broader understanding of error characterization in quantum computing and provides a robust framework for future studies in quantum gate fidelity under realistic noise conditions.

Contents

1 Introduction

In the process of realizing a large-scale quantum processor, there exists a need for a standard set of metrics to give a qualitative assessment of a quantum device. By adhering to such a set of standard metrics and protocols, the performance and reliability of a quantum device can be quantified in such a way that not only meaningful comparisons between devices can be performed, but the result gives some direct insight into the performance and reliability of the device itself. Among the standard set of such benchmarking protocols that are in use today, Gate Set Tomography (GST), although being very resource-intense, attempts to characterize a set of quantum gates completely and self-sufficiently [3]. One of the underlying assumptions which is at the core of the GST protocol however, is that the time-correlation of the noise is much shorter than that of the benchmarked quantum process, known as Markovian noise. Seeing as this assumption can't actually be guaranteed in experimental conditions, there's an emerging need to explore beyond these assumptions. This thesis delves into the impact of time-dependent noise on the effectiveness and reliability of GST, aiming to assess and quantify its robustness to deviations from purely Markovian noise behavior. To this end, a software suite has been developed in the form of a standalone Python library which is used to investigate the effects of time-correlated noise on the robustness and interpretability of GST.

1.1 Implementation Overview

The implementation of this project was carried out using Python, resulting in a comprehensive software suite designed as a Python library. The primary goal of the library is to enable users to test the GST protocol under various conditions involving correlated noise. Users have considerable flexibility in specifying the types of noise they wish to test. Based on this noise, users can generate a set of gates that are temporally correlated by the noise. These gates can then be saved to disk, allowing users to run extensive sets of GST experiments on the same gates. The library includes robust tools for analyzing the results. Analysis involves establishing a baseline, which is calculated from the generated gates before the GST experiments are conducted. Additionally, the analysis includes running numerous GST experiments and extracting and averaging the data across multiple runs to ensure reliability and accuracy. Significant effort was dedicated to making the software performant. Performance bottlenecks were accounted for by making extensive use of multi-threading, allowing the software to run especially well on systems with a large number of CPU cores. This optimization ensures that the library can handle large-scale experiments efficiently. The software is designed for researchers who want to establish baselines for their experiments. By allowing researchers to specify the kind of noise present in their systems, the software aims to closely emulate real-world conditions. This feature helps in providing a more accurate representation of what is happening in actual quantum systems. In summary, this software suite was developed with generality, extensibility, and performance in mind, making it a

valuable tool for future research and experimentation in GST analysis. This thesis aims to demonstrate the software's capabilities through various experiments, showcasing its practical applications and effectiveness.

1.2 Noise Intermediate-Scale Quantum (NISQ)

Quantum computation represents a fundamental shift from classical computing, leveraging the principles of quantum mechanics to process information in a fundamentally different way from how classical computation works. At the heart of quantum computing are quantum bits, or qubits, which differ from their analogous classical bits in their ability to exist not only in binary states of 0 or 1 but also in superpositions of both states simultaneously. This fundamental difference allows quantum computers to handle exponentially more information than classical computers for certain types of problems, providing the potential to solve computational challenges that are currently intractable. The ability to solve problems on a quantum computer that are infeasible to calculate on a classical computer is referred to as "quantum advantage" and is one of the key goals of contemporary research [2].

The realization of true quantum advantage however is dependent on our ability to perform high-fidelity quantum logic operations. Achieving the necessary low error rates in quantum processors with many coupled qubits is a significant challenge due to the inherent susceptibility of qubits to noise. The current generation of quantum devices operates within the so-called Noisy Intermediate-Scale Quantum (NISQ) era, a term coined by John Preskill to highlight the prevalence of noise as a significant barrier to the realization of quantum advantage [9]. In this phase of quantum technology, the fidelity of quantum operations is continually undermined by various noise sources. Central to the progression of quantum computing beyond the NISQ era and towards this new era of quantum advantage then is the ability to accurately classify and characterize these errors within these current-generation quantum systems. To address this need, a lot of different error classification and characterization schemes exist, known collectively as Quantum Characterization, Validation & Verification (QCVV) schemes. Although it wouldn't be difficult to come up with a simple experiment that measures a certain characteristic of a quantum device, more involved schemes exist within QCVV that go beyond extracting simple metrics. Many of these schemes are in widespread use today and have seen an increased amount of standardization. Among them is Gate Set Tomography (GST), the main subject of interest for this thesis.

1.3 Gate Set Tomography

While most protocols in Quantum Characterization, Verification, and Validation (QCVV) offer limited insights into the quantum devices they assess, typically providing basic metrics such as dephasing time and operation fidelities, Gate Set Tomography (GST) stands out as a comprehensive benchmarking protocol.

GST is designed to provide a complete view of a quantum computer's capabilities by reconstructing every gate in the gate set. This thorough approach makes GST's results especially valuable, as they offer the most comprehensive information among QCVV schemes, thereby giving researchers the most complete insight into the shortcomings of their device.
This all sounds great of course, but it should be noted that GST is not without its drawbacks:

1. **Poor scalability**: GST scales notoriously poorly, as the amount of data that the analysis needs explodes with the number of qubits subject to this analysis. As a consequence, complete characterizations of devices with more than 3 qubits are largely infeasable because of this issue[4].

2. **Assumption of Markovian noise**: At the core of GST's methodology is the assumption that all noise affecting the system is Markovian, meaning the noise is memoryless[4]. This assumption simplifies the analysis, and might even improve it, but does not always hold in practice. Real-world quantum systems invariably exhibit some degree of non-Markovian noise, which involves memory effects and correlations over time. By disregarding this type of noise, GST risks oversimplifying the system's dynamics, potentially leading to results that are but poor approximations of the actual characteristics of a device.

1.4 Problem statement

Despite its effectiveness under ideal conditions, GST's performance in practical quantum systems, where noise often exhibits non-Markovian characteristics, is poorly understood. Non-Markovian noise introduces memory effects where past states influence future states, a scenario fundamentally assumed non-existent in the standard GST protocol. This raises a critical concern: How predictably does GST perform in the presence of non-Markovian noise? There is a notable gap in the existing research regarding the impact of time-correlated noise on GST, a gap that this thesis aims to explore.

1.5 Research Goals

This study will assess how the GST protocol performs when confronted with non-Markovian noise, which is prevalent in quantum systems. A software implementation has been developed in Python as a critical part of this thesis, simulating the effect of time-correlated noise on GST. This allows for an in-depth analysis of the protocol's resilience and the potential need for adjustments or enhancements. The software serves not only as a tool for this specific investigation but also provides a framework for the broader quantum computing community to test and refine error characterization methods under the specific condition of correlated noise.

2 Theoretical Background

This section provides a comprehensive overview of the relevant theory underlying this thesis, particularly focusing on Gate Set Tomography (GST) and its context within quantum characterization, verification, and validation (QCVV). This section is structured to first introduce the foundational aspects of quantum circuits and the unique challenges presented by noise in quantum systems. It then delves into various tomographic techniques, with a detailed examination of GST and its components such as fiducial circuits, germs, and the critical processes of gauge fixing and error estimation.

2.1 Quantum circuits, Operators, and the Gate Set

Quantum computation is most commonly formalized through the concept of quantum circuits. Understanding this formalization is crucial as it forms the backbone of how quantum algorithms are implemented on physical systems.

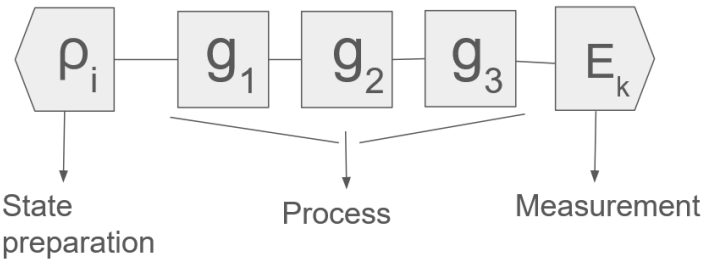


Figure 1: Abstract representation of a quantum circuit featuring state preparation, a number of gates and finally measurement

Every program that can be run on a quantum computer can be represented as a quantum circuit, in which a computation is a sequence of quantum gates, which are reversible transformations on a quantum mechanical system. This model mirrors classical computation, which uses a sequence of logical operations on a known initial state to process information. Quantum circuits can be decomposed into three parts as is shown in Figure 1:

1. **State Preparation**: The process where qubits are prepared in a known standard state ρ_i, typically the ground state ρ_0. Most commonly, initialization is achieved by allowing the qubits to relax into their ρ_0 state. This step ensures that the computation starts from a predictable and reproducible state.

2. **Process**: The process part of the circuit consists of a sequence of Quantum Logic Gates, these are the fundamental building blocks of quantum computation, analogous to logic gates in classical circuits. Quantum gates

manipulate the state of qubits through precise, controlled operations. These will be discussed in the next chapter in more detail.

3. **Measurement**: The final step in a quantum circuit where the state of qubits is measured. The outcome of these measurements provides the results of quantum computations. Measurement collapses the state of a qubit into one of the basis states, yielding classical bits of information.

2.1.1 Quantum States and Operators

In order to move from a conceptual understanding of a quantum circuit, to its mathematical formalism, we will first look at the representation of a quantum state. A quantum system comprising one qubit will have a state that can be represented as a vector in a two-dimensional complex vector space. The most common basis states for a single qubit are $|0\rangle$ and $|1\rangle$:

- **Zero state ($|0\rangle$)**:
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$
- **One state ($|1\rangle$)**:
$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

As part of the process, different quantum logic gates will act on this state and transform the system from one state to another. In the mathematical sense, these logic operators are represented by unitary matrices. Unitary operators are essential because they preserve the norm of the quantum state, ensuring that probabilities sum to one.

Example: Applying a Hadamard Gate to a $|0\rangle$ State
The Hadamard gate (H) is a single-qubit gate that creates superposition states from the basis states [11]. It is represented by the following unitary matrix:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

To understand how the Hadamard gate acts on a quantum state, consider applying it to the $|0\rangle$ state:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Thus, after applying the Hadamard gate, the $|0\rangle$ state is transformed into a superposition state [11]:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

This example illustrates how unitary operators like the Hadamard gate transform quantum states, enabling the creation of complex quantum algorithms.

2.1.2 Superoperators

In the case of GST, the choice is made to work with the mathematical formalism of superoperators rather than operators. This approach provides several benefits, we will explain several of these benefits without going into detail too much.

For a superoperator to represent a physically possible operation, it must satisfy two key properties: complete positivity and trace preservation [15].

- **Trace Preservation (TP):** This condition ensures that the total probability over all basis states remains 1. In other words, it guarantees that superoperators map quantum states to physically allowed states, preserving the total probability and ensuring no loss or gain of information.
- **Complete Positivity (CP):** Complete positivity ensures that probabilities remain non-negative, preventing physically impossible states. This condition is crucial for maintaining the physical realism of the quantum system, as it avoids the emergence of negative probabilities.

These properties can be enforced comparatively easy using superoperators, making them particularly suitable for GST, as they ensure the physical validity of the quantum operations being modeled. Additionally, in our simulations, superoperators allow for the averaging over many different gate operations, providing a more comprehensive understanding of the overall system behavior. This capability is essential when dealing with large numbers of gates and complex noise environments, as it helps extract meaningful averages and insights from the data.

2.1.3 Understanding the Gate Set

The gate set is a fundamental concept within the framework of quantum circuits and is central to the operation of GST, as its name implies. The gate set, denoted as $G = \{\rho_i, g_j, E_k\}$, represents a complete set of capabilities that a quantum computer possesses [4]. This set is comprehensive, encompassing all the different kinds of circuit elements that the quantum computer supports:

- **State preparation (ρ_i)**: These are the different initial states in which qubits can be prepared at the start of a computation. While the most common initial state is the ground state ρ_0, quantum computers may support various other initializations to enable diverse computational strategies.
- **Process (g_j)**: These include all possible quantum gates that the quantum computer can perform. Quantum gates are the analogs of classical logic gates but for quantum operations, allowing for the manipulation and control of qubit states in complex ways that enable quantum algorithms.
- **Measurement Operations (E_k)**: This category includes all possible measurements that can be executed on the quantum states at the end

of the circuit. Measurements in quantum computing are crucial as they convert quantum information into classical information, thus providing the output of the quantum computation.

This comprehensive description of a quantum computer's capabilities through its gate set is crucial because it uniquely defines the types of circuits that can be run on the machine.

2.2 (Non-)Markovian Dynamics

As previously noted, noise represents the primary bottleneck in the performance of current generation quantum devices. Extensive research has been conducted to identify the origins and characteristics of this noise in the context of quantum systems. A key method for characterizing noise involves analyzing its Fourier transform; in the frequency domain, different sources of noise exhibit distinct responses.

The Power Spectral Density function (PSD): $S(\omega)$, describes how the power of a signal is distributed across different frequencies within the power spectrum [14]. This distribution is crucial for understanding and simulating the physical processes underlying the noise in quantum systems.

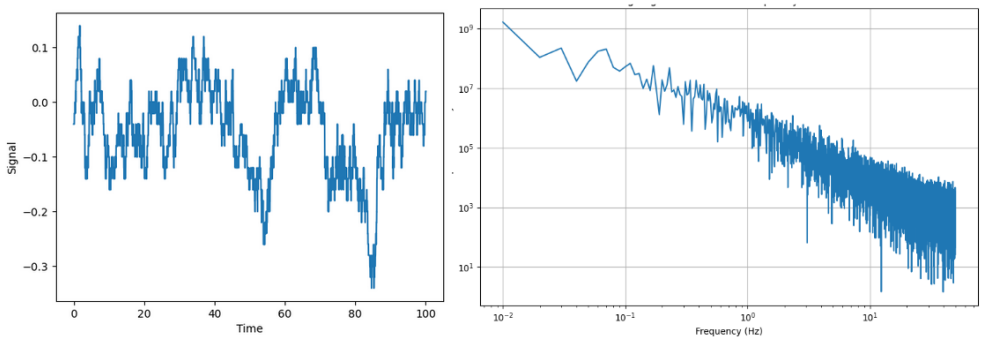


Figure 2: Example of an arbitrary 1/f noise-signal in time domain on the left, as well as its corresponding PSD on the right

Among the various classes of noise, white noise is the simplest to understand, characterized by a constant PSD across all frequency components. This type of noise is known as Markovian or non-correlated noise. If this were the only type of noise present, our research would be unnecessary.

However, quantum devices also encounter other types of noise, notably pink noise, which follows a power law distribution described by $\frac{1}{f^{\beta}}$. Pink noise, which is apparent in all sorts of electrical systems, has enigmatic origins[8]. While several explanations have been proposed, its true source appears to be a combination of different stochastic processes[13]. Since GST in its core, assumes all the noise in the system is uncorrelated, the correlated nature of pink noise

raises significant concerns for the GST protocol, highlighting the need for this research.

2.3 Tomographic techniques

Having introduced the concept of the gate set, we now transition to a more detailed exploration of the GST protocol. To fully understand GST, it is important to see it as part of a broader class of tomographic protocols, each designed to reconstruct a different part of the Gate Set:

1. **State Tomography**: This technique focuses on reconstructing the state preparation set of the system. It provides insights into how quantum states are prepared, which is fundamental for ensuring the accuracy of quantum computations from the outset.

2. **Process Tomography**: Aimed at reconstructing the set of gates a quantum computer can perform, this method assesses the transformations that quantum states undergo during computation [6]. Understanding these transformations is crucial for the development and refinement of quantum gates.

3. **Measurement Tomography**: This approach is concerned with reconstructing the measurement operators that a computer uses to observe quantum states. Accurate measurement is critical for interpreting the outcomes of quantum computations reliably.

2.3.1 State Tomography

State tomography is arguably the easiest to understand out of all of the tomographic techniques. The primary goal of state tomography is to reconstruct, or build a detailed description of, the different state preparations included in a quantum computer's gate set: ρ_i[4] [1].

In order to achieve this, the protocol simply combines the state preparation operator we want to reconstruct (ρ_i) with every different measurement operator in $\{E_k\}$ in the gate set as a circuit. It then takes this set of circuits and runs them for a large number of shots. Assuming we have enough measurement operators in our gate set with which to measure the state after ρ_i, we can simply reconstruct ρ_i through a simple linear inversion process, in which we treat the circuit outcomes as probabilities[4].

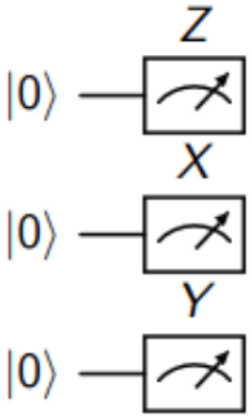


Figure 3: Example of a set of circuits that might be used for reconstructing the $|0\rangle$ state preparation using a set of linearly independent measurement operators

The core problem that State Tomography suffers from however, is that in order for our results to be meaningful, we have to assume perfect knowledge of the set of measurement operators, or we simply have to assume they are inherently error-less [4]. The assumption that measurement operators are perfectly known and perform flawlessly is a simplification that simply does not hold true in experimental conditions. This approximation is necessary to make the estimation possible, but does introduce potential for large discrepancies between the reported reconstruction and the actual underlying physical implementation [6].

2.3.2 Chicken-egg problem

State Tomography is not the only tomographic technique that suffers from this central flawed assumption. In fact, every single one of the three techniques outlined above, State Tomography, Process Tomography, as well as Measurement Tomography, suffer from the same fundamental problem [4]. These techniques function according to the same principal as State Tomography, they try to reconstruct a small part of the gate set by combining it with all the other parts of the gate set, which for the purpose of simplicity, are assumed perfectly known. This is illustrated further in Figure 4 down below.

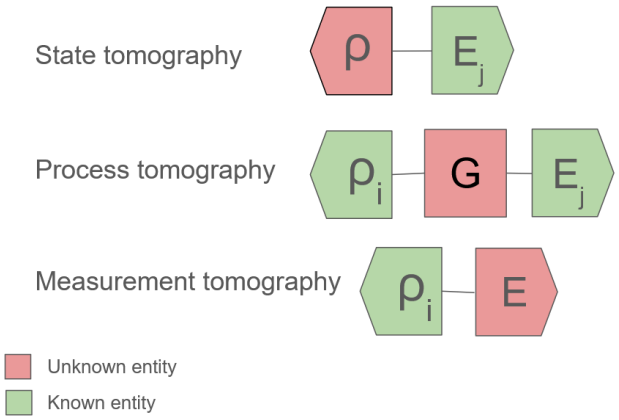


Figure 4: Schematic showing three different tomographic techniques and the parts of the gate set they assume to be perfectly known

This fundamental problem, where one can only reconstruct a part of the gate set by perfect knowledge of the rest of the gate set, is the problem that GST tries to solve and is its fundamental raison d'être. Instead of taking the different parts of the gate set at a time, it tries to provide a complete and integrated characterization that encompasses state preparation, gate operations, and measurement processes all at once. How this is done will be explained in the next section.

2.4 Gate Set Tomography

Gate Set Tomography (GST) represents a significant advancement in the field of Quantum Characterization, Verification, and Validation (QCVV). Unlike other tomographic techniques that aim to reconstruct individual components of the gate set, GST ambitiously seeks to reconstruct the entire gate set of a quantum computer simultaneously and self-sufficiently. This means reconstructing state preparation and measurement (SPAM) errors, as well as errors in the quantum gates.

2.4.1 Comprehensive approach of GST

GST's methodology involves creating an exhaustive set of quantum circuits designed specifically to amplify and identify different types of errors within the quantum gate set. This approach is notably more involved than other tomographic methods. The exact generation scheme which is used for generating these circuits is quite an involved process and can differ from implementation to implementation [7], therefore it is not detailed in this thesis due to its complexity. The basic principle of this set of circuits however, is that each circuit tries to amplify a different error. The design of this set of circuits is crucial

as each is tailored to probe specific errors that might be present in the gate set. Instead of relying on linear inversion, the analysis phase of GST involves a log-likelihood optimization process, where the most likely reconstructed gate set is determined based on the experimental data [4].

2.4.2 Fiducial circuits and Germs

In practical quantum computing environments, the native gate set often does not provide an informationally complete basis; for example, many quantum computers primarily offer initialization and measurement in the Z-direction only. This limitation makes it difficult to formulate enough different circuits to amplify certain errors in the gate set. To overcome this, GST introduces "fiducial circuits." These circuits are composed of elements from the existing gate set and are designed to extend the system's capabilities into a fully informationally complete framework. Fiducial circuits can be categorized into two types:

- Fiducial State Preparation Circuits: These circuits utilize the available gate set to prepare the quantum system in specific basis states necessary for a comprehensive examination.
- Fiducial Measurement Circuits: Similarly, fiducial circuits allow for measurements in bases beyond the native capabilities of the system.

Beyond fiducial circuits, GST also constructs "germs", which are short combinations of smaller gate elements. These germs serve a dual purpose: they test the quantum system's response to repeated applications of certain gate combinations, and they amplify the errors in a controlled and measurable way. Instead of directly testing on the elements in the gate set, GST tries to reconstruct the engineered fiducial circuits and germs. The results from these evaluations are then used to infer the properties and potential errors associated with the individual elements of the gate set. This method allows GST to provide a detailed and accurate characterization of the gate set, despite the inherent limitations in direct state initialization or measurement capabilities.

2.4.3 Relative reconstruction and Gauge Fixing

A distinctive aspect of GST is its approach to measuring each gate relative to others within the same set, similar to other tomographic techniques but more comprehensive. It is crucial to understand that this measurement is done up to a gauge transformation, which is an essential part of the GST process. This transformation acknowledges that different representations of the gate set, though mathematically distinct, can be functionally equivalent. Take for example any unitary matrix M, it can be shown that by applying the following transformation [4]:

$$\begin{aligned} \langle E_k| &\to \langle E_k|M^{-1}, \\ |\rho_j\rangle &\to M|\rho_j\rangle, \\ G_i &\to MG_iM^{-1}. \end{aligned}$$

, we get a new gate set, which is functionally equivalent to the old gate set. This illustrates that any element of the gate set can undergo a transformation by a consistent matrix, resulting in a new gate set that, while appearing different, performs identically in a functional context. This leads to the critical final step in GST known as gauge fixing.

The goal of gauge fixing in GST is to find a unitary tranformation matrix M, which aligns the reconstructed gate set as closely as possible with the target gate set. This step is vital as the initial gate set that the GST might return does not necessarily resemble the true underlying gate set. However, this process might lead to underreporting of errors or misattribution of errors among gates. For instance, if a single gate exhibits errors, GST might partially attribute these errors to other gates due to this process of gauge fixing.

2.5 Germ Amplification and the Markovian noise assumption

To enhance the accuracy of GST, the analysis does not solely rely on measuring all possible pairs of germs with every combination of initialization and measurement. Instead, it incorporates longer circuits where germs are repeated, in a process known as "Germ amplification". This is illustrated in Figure 5, where the Identity-gate is repeated n times. This strategy is pivotal for improving error estimation through the observation of how errors accumulate over time [4].

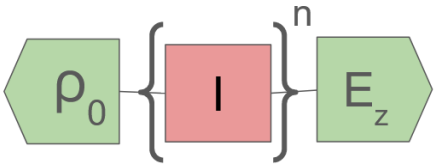


Figure 5: Example of a circuit that employs germ amplification, by varying the amount of repetitions (n), the effects of the error in the Identity-gate on the circuit can be amplified.

The utilization of longer circuits allows for a more detailed examination of how errors in a germ amplify as the circuit extends, leading to better approximations. However, it should be noted that for this approach to be viable, the assumption is made that errors are multiplicative. This assumption is unreliable at best, when considering the impact of correlated noise. To illustrate this, let's take the circuit from Figure 5 as an example, and imagine what happens when we run it for different amounts of repetitions of I, first with an identity gate experiencing purely Markovian noise and then one which is subject to non-Markovian noise processes.

1. Markovian Noise: Under the assumption of Markovian noise, the error introduced by the initialization can be thought of as being compounded

throughout the length of the circuit. Typically, this results in an exponential decay of the state's fidelity, converging towards 0.5 as the circuit length increases. This behavior, characteristic of exponential decay due to repeated application of the same error, is depicted in Figure 6

2. Correlated Noise: When dealing with correlated noise, the outcome of the experiment is different. As shown in Figure 7, the decay pattern in these scenarios is not exponential but follows a Gaussian decay. This difference is crucial as GST fundamentally assumes that errors can be multiplied and represented as an exponential decay over successive applications of a germ.

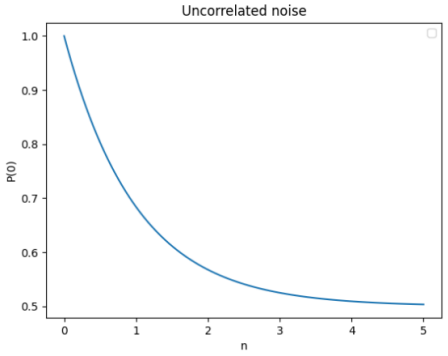


Figure 6: The results of running the circuit in Figure 5 in the presence of Markovian noise, showing exponential decay

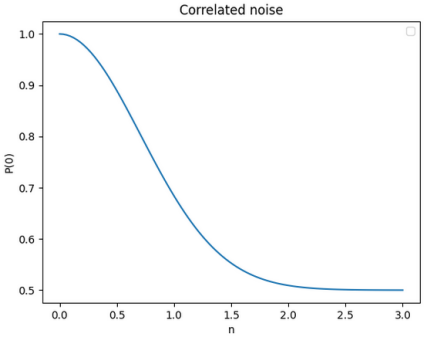


Figure 7: The results of running the circuit in Figure 5 in the presence of correlated noise, showing Gaussian decay

In realistic situations, the observed outcome of the experiment will show a mix between exponential and Gaussian decay, depending on the amount of correlation in the underlying noise processes.

2.5.1 Consequences of the Markovian assumption

The assumption of error multiplicability works well under Markovian conditions where noise effects are memoryless and independent of previous states. This assumption facilitates the use of germ amplification, which generally enhances the reliability of GST results under Markovian noise. However, this approach encounters significant challenges when applied to non-Markovian (correlated) noise:

- Fitting Challenges: In the presence of correlated noise, fitting an exponential decay model to inherently non-exponential decay data can lead to very large discrepancies. The GST framework, while robust under Markovian assumptions, may not accurately reflect the underlying dynamics in

non-Markovian scenarios. The usage of error amplification by repeating germs might not lead to improved results, and might even be detrimental.

- Model Discrepancy: By trying to fit an exponential decay to Gaussian-like decay data, GST might not only misestimate the severity of errors but also misattribute the source of errors across different components of the gate set. This misestimation could lead to incorrect conclusions about the system's reliability and error-resilience.

This model discrepancy is something that GST luckily can detect itself already, and is a good first step towards dealing with non-Markovian behaviour. After running the GST analysis, a "model violation" is reported, which is a measure of how likely it is that the experiment was affected by some kind of non-Markovian noise.

In summary, while the strategy of using longer circuits in GST significantly enhances error analysis under ideal Markovian conditions[4], it presents limitations and challenges when extended to systems with correlated noise. The effects of this is what this thesis aims to explore.

2.5.2 Fiducial pair reduction

Another critical element of GST is the process known as fiducial pair reduction. Fiducial pair reduction focuses on reducing the number of combinations of preparation and measurement fiducials used in experiments. By carefully selecting which fiducial pairs are necessary and which can be omitted, researchers can reduce the total number of experiments[10]. This is particularly valuable because each GST sequence can be resource-intensive. Despite the reduction in sequences, the chosen subsets of fiducial pairs are designed to be still informationally complete for the aspects of the gate set that are most prone to errors. This means they can capture all the necessary information to fully characterize the quantum gates, particularly the errors, without redundant measurements. With fewer circuits to evaluate then, the computational load decreases. This makes the analysis faster and can facilitate the handling of larger or more complex quantum systems within practical time frames.

2.6 Software Dependencies

The functionality and effectiveness of our Python package for investigating time-correlated errors in GST are greatly enhanced by two major software dependencies:

1. **PyGSTi**: PyGSTi is a robust Python library tailored specifically for implementing GST [5]. PyGSTi simplifies GST implementation by offering a user-friendly framework designed to handle its complexity, making it extremely useful for conducting the advanced calculations and analysis required in our studies. It allows for the generation of HTML reports in which the GST results can be inspected visually in an intuitive way. It

is used in the implementation to run the bulk of GST related analysis: initializating GST experiments, as well as performing the GST analysis itself.

2. **DM Solver**: A software suite developed at QuTech, this software component is critical for solving the time-dependent Schrödinger equation, a fundamental equation in quantum mechanics that describes how the quantum state of a physical system changes over time. DM Solver utilizes a C backend to enhance processing power, enabling it to handle what is quite the computationally intensive task. The increased processing power is crucial for accurately simulating the dynamics of quantum gates under the influence of various noise models. We use it in the implementation to convert our noisy gates from a Hamiltonian representation into a Unitary Matrix representation.

3 Implementation overview

The methodology of this research revolves around developing and utilizing a software suite designed to investigate the robustness of the GST protocol under conditions of correlated noise. This software suite has been developed in Python and has as its primary aim to facilitate a detailed exploration of how GST responds to different types of time-correlated noise, which are known to challenge the standard assumptions of the GST framework. This chapter will go into the overall architecture of the software suite with regards to how the experiments are performed. For the actual implementation details, please refer to Section 4.

3.1 Gate Generation

In order to investigate the effects of time-correlated noise on a GST experiment in simulation, there is a need to be able to generate series of gates whose errors are correlated over time. The generation of these gates is computationally intensive, as it involves calculating the unitary transformations for each gate under the influence of the specified noise by solving the time-dependent Schrodinger equation. These calculations ensure that the gates realistically reflect the dynamics expected in a noisy quantum environment. This fundamental stage involves the separate generation of quantum gates prior to conducting GST experiments. This approach is strategic; by separating the gate generation from the GST experiments, it allows the pre-generated gates, complete with their noise characteristics, to be saved to disk. These stored gates can then be readily loaded for use in multiple GST experiments. The separation of gate generation from the GST experiments is a deliberate design choice that enhances the flexibility and efficiency of the research process. This approach not only facilitates multiple experimental runs using the same set of noise-impacted gates but also significantly reduces the computational load by leveraging pre-computed and stored quantum gates. The process is thus distinct from the actual experiments

to enhance efficiency and reusability. The gate generation operates in different stages as outlined below:

3.1.1 Noise signal generation

Users specify the characteristics of the noise impacting the quantum gates by defining the Power Spectral Density (PSD) characteristic of the noise. As part of the gate generation the PSD is converted into a time-domain signal by using the approach outlined in Section 4.1, this process is shown schematically in Figure 8. After the generation of this time-correlated noise signal, the noise signal can then be used to generate a sequence of time-correlated gates.

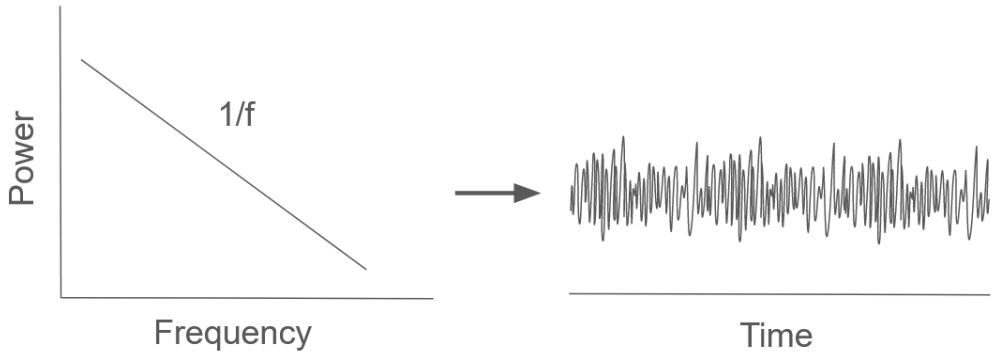


Figure 8: Schematic showing the conversion of a PSD (Left) to its corresponding time-domain signal (Right)

3.1.2 Batch Generation of Gates

Following the generation of the time-correlated noise signal, it is divided into discrete time slices. Each slice corresponds to a specific interval during which the quantum gate operations occur. This segmentation facilitates the generation of a separate gate for each time slice, embedding the noise characteristics of that particular segment into the gate's operation.

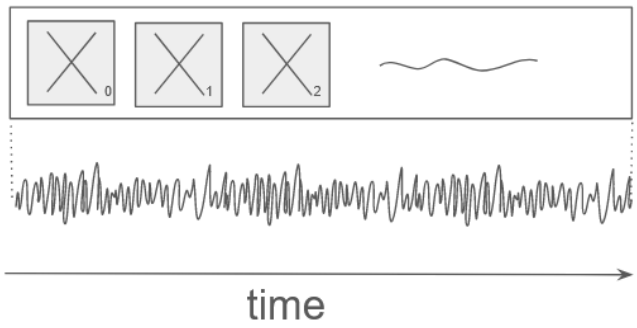


Figure 9: Schematic showing the correspondence between the time-correlated noise and the resulting sequence of time-correlated gates.

This approach ensures that the sequence of generated gates reflects a temporal correlation of errors, these sets of temporally correlated gates can then be used in the simulation to accurately reflect time-correlated effects.

3.2 Running GST

This is the stage in which actual GST experiments are performed. This includes generating the circuits needed for the GST experiment, simulating these circuits with the help of the previously generated gates, as well as the analysis of the corresponding results.

3.2.1 Experiment Initialization and Circuit Generation

Using the Python package known as PyGSTi, we initiate standardized GST experiments by specifying various options that influence the design and complexity of the circuits. This process allows for the generation of a comprehensive list of circuits tailored to our specific GST parameters. Options such as fiducial pair reduction schemes, gate set specifications, and maximum circuit depth are set during this phase. After generating the necessary circuits, and (optionally) reducing them using a Fiducial Pair Reduction scheme, the circuits are ready to be simulated.

3.2.2 Circuit Simulation

In the circuit simulation phase, each circuit generated during the initialization stage is simulated a number of times, referred to as shots. Circuits are simulated one by one, and for every gate in a circuit, we increment a time variable. The simulation happens as is shown in Figure 10. Where gates are taken from the sequence of gates in order, in this way, we can simulate the process of simulating GST over a longer timespan.

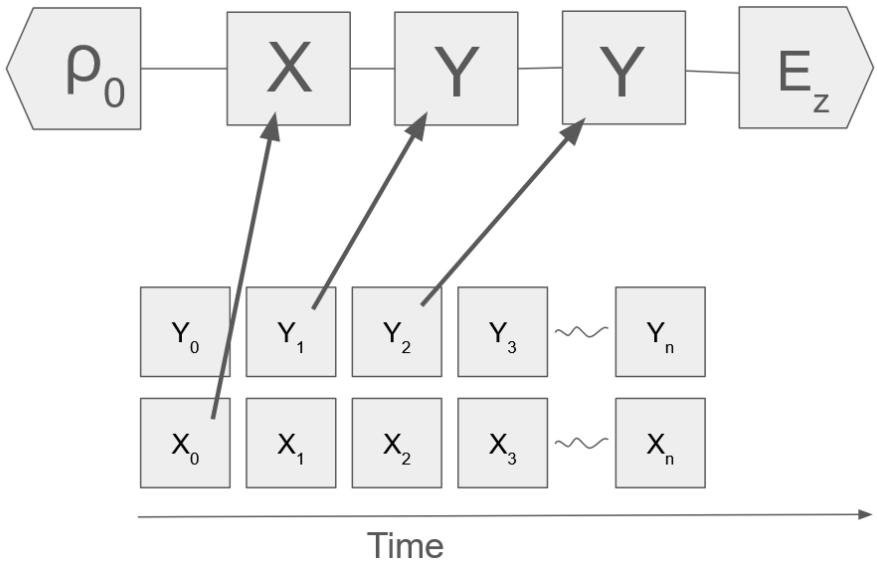


Figure 10: Example of a circuit being evaluated; Circuit simulation happens as a time-ordered process in which the gates in the circuit are taken from a long sequence of pre-rendered, time-correlated gates.

3.2.3 GST Analysis

After the circuits are simulated, the results are fed back into PyGSTi, which then performs a detailed analysis and generates a comprehensive report. This report contains the GST data that we will compare to the data we extract directly from our generated gates.

4 Implementation details

As part of the experiment, a software suite has been developed that specializes in investigating the robustness of the Gate Set Tomography (GST) protocol. This section will go over the different capabilities of the software, outlining the exact functionality of each of the components as well as explaining how all everything is brought together to generate the results.

4.1 Noise generation

In order to subjugate the generated gates to a certain type of correlated noise, we allow the user to specify the noise in the system by specifying its Power Spectral Density (PSD). This section will deal with the specifics of how this PSD representation is converted into a noise signal in time domain which has the proper characteristics. By allowing the user to specify a custom PSD for the noise, we allow them to, for instance, model noise with a distribution $\frac{1}{f}$,

which is common in quantum systems. But in principle, the user is allowed to specify any type of arbitrary and exotic noise correlations that can be captured by a PSD.

4.1.1 Fourier Transform and Noise Simulation

To simulate noise according to a specified Power Spectral Density (PSD), we begin with the Fourier transform of a signal, $x(t)$. The Fourier transform, denoted as $\widehat{x}(\omega)$, transforms the time-domain signal into the frequency domain, providing a representation of the signal's frequency components:

$$\widehat{x}(\omega) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi\omega x} dx$$

This transformation is essential for linking the time-domain characteristics of the noise with its frequency-domain representation, facilitated by the Spectral Density Function.

4.1.2 Linking Fourier Transform to Spectral Density Function

The relationship between the Fourier transform of the noise $\widehat{x}(\omega)$, and the Spectral Density Function is established through a probabilistic model [12]. Each frequency component of the Fourier transform, $\widehat{x}(\omega)$, is modeled as a complex Gaussian random variable:

$$\widehat{x}(\omega) = \mathcal{N}(0, \frac{1}{2}S(\omega)) + i\mathcal{N}(0, \frac{1}{2}S(\omega))$$

Here, $\mathcal{N}(0, \frac{1}{2}S(\omega))$ represents a normal distribution with mean zero and variance $\frac{1}{2}S(\omega)$. This formulation implies that each real and imaginary part of the Fourier components is independently distributed according to the Gaussian distribution, with the variance directly derived from the PSD.

In practice, generating noise from the Spectral Density Function involves computing the inverse Fast-Fourier transform of $\widehat{x}(\omega)$, like so:

$$x(t) = \text{IFFT}(\widehat{x}(\omega))$$

By specifying $S(\omega)$, and using the relationship established above, we can generate a time-domain signal that accurately reflects the desired noise characteristics. This approach not only ensures that the simulated noise adheres to the theoretical model but also allows for flexibility in specifying different types of noise distributions.

4.2 Gate generation

In the process of gate generation, we employ the Density Matrix Solver (DM-Solver), a software tool designed to generate numerical solutions of the Schrödinger

equation. This solver is particularly adept at handling time-dependent Hamiltonians. By inputting the time dependent Hamiltonian corresponding to a quantum gate, the DMSolver computes the corresponding unitary matrix representing this gate.

4.2.1 Gate Specification

Hamiltonians are defined by specifying the frequency of a qubit, along with optional driving signals characterized by their strengths and timings. By choosing appropriate parameters, it is possible to create a time-dependent Hamiltonian that corresponds to an ideal quantum gate.

The DMSolver can convert this time-dependent Hamiltonian into a unitary matrix, representing the gate equivalent to the Hamiltonian. For example, when generating the unitary matrix corresponding to an ideal X gate, the Hamiltonian is defined as follows, with parameters T_{start} and T_{end} representing the times during which the driving signal starts and ends, respectively:

$$H_{ideal}(t) = \begin{cases} E_z \cdot \sigma_z & \text{if } t < T_{start} \\ E_z \cdot \sigma_z + A\cos(\omega t) \cdot \sigma_x & \text{if } T_{start} \leq t \leq T_{end} \\ E_z \cdot \sigma_z & \text{if } t > T_{end} \end{cases}$$

The DMSolver then takes this time-dependent Hamiltonian and converts it into the unitary matrix corresponding to the ideal gate. Given that the problem is solved numerically, we model it as a discrete problem, meaning the Hamiltonian has a distinct number of steps. The DMSolver begins with the identity matrix to represent the initial unitary gate, U_0. For each time step in the simulation, the DMSolver incorporates the effects of the current Hamiltonian into the gate:

$$U_t = U_{t-1} \cdot e^{-i\dot{H}(t) \cdot dt}$$

At the end of the process, U_n represents the unitary matrix corresponding to the ideal version of the gate.

4.2.2 Adding Noise

While the DMSolver offers capabilities to incorporate various noise models directly, the decision was made to generate the noise using the scheme described previously in Section 4.1. The approach we take involves first crafting the Hamiltonian corresponding to an ideal version of the gate. This ensures that the initial model represents a noise-free, theoretical baseline to which noise can then be added and studied. Following the establishment of the ideal Hamiltonian, we overlay our previously generate noise signal on top, this is done by adding the noise in the Z-direction. This is representative of fluctuations in the energy levels of the qubit. For a noise signal $S(t)$, the resulting noisy Hamiltonian thus looks like this:

$$H(t) = H_{ideal}(t) + \sigma_z \cdot S(t)$$

After embedding the noise into our process, we engage the DMSolver to convert the time-dependent hamiltonian into its corresponding unitary, representing an implementation of a single gate execution on our simulated quantum computer.

4.2.3 Numerical Error Reduction

One of the problems that soon became apparent during the generation of gates, was the fact that even when taking the Hamiltonian of a gate corresponding to it's ideal version, so the one without any noise added, would return a gate with significant errors when compared to the actual target gate. This numerical error is present in most numerical simulations, but was too large to ignore in the case of our ideal gates as can been seen in Figure 11.

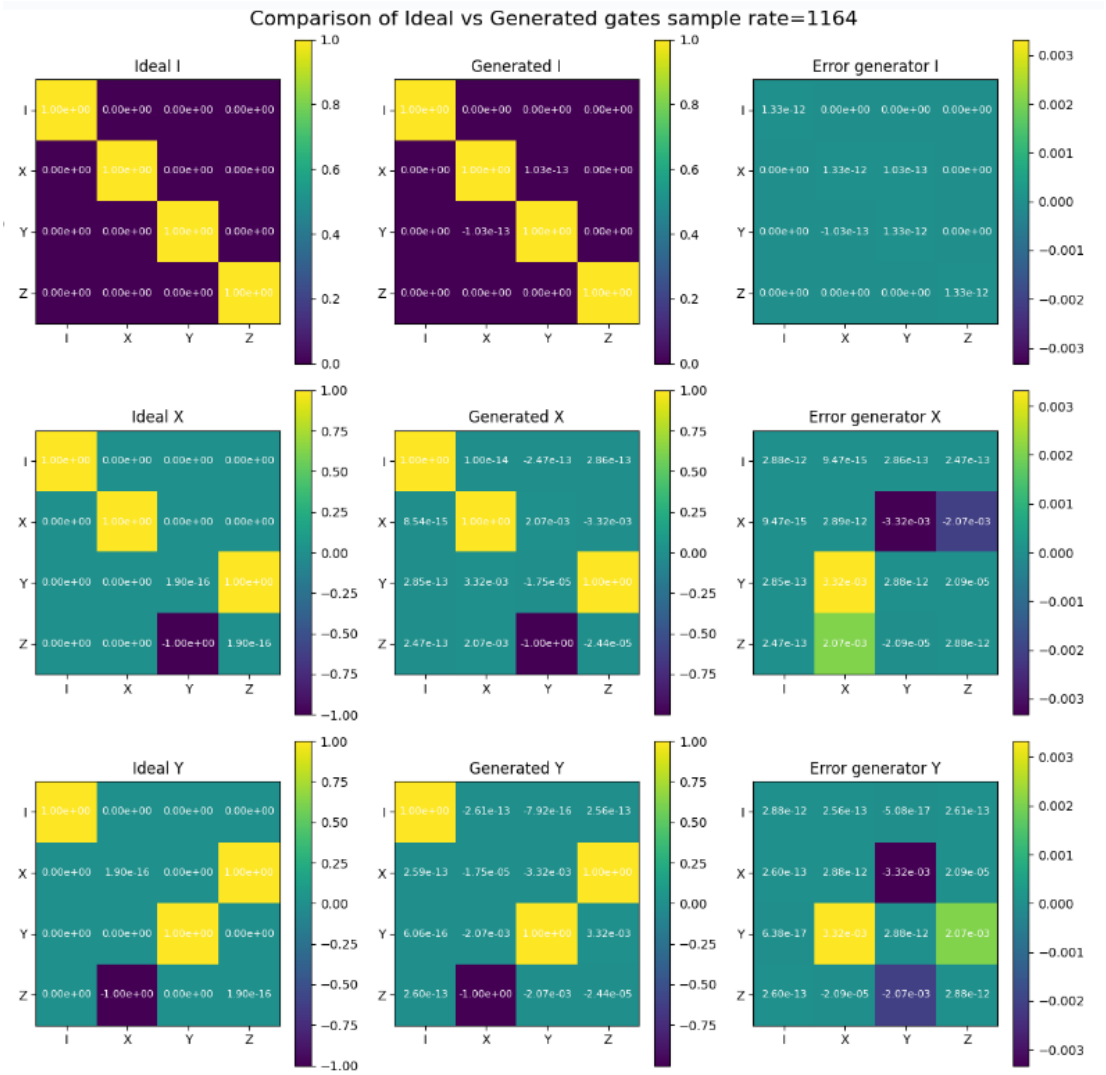


Figure 11: Gate generation for 'ideal' gates using previously mentioned generation scheme, the error generators on the right show relatively large discrepancies.

To understand why this happens we must look at the case of the ideal X-gate, whose Hamiltonian has the following general form:

$$H_{ideal}(t) = \sigma_z \cdot E_z + \sigma_x \cdot Acos(\omega t)$$

As it turns out, when doing a numerical simulation, we have to sample from this function for each time-step in our simulation. The reason for these big numerical errors is that when trying to calculate the time-dependent Schrodinger equation for time t, we are sampling from $H(t)$ and using this result to calculate

the time-evolution between for the time-slice $[t, t + \delta t]$. This is particularly problematic when sampling from the time-dependent Hamiltonian for the X-gate and Y-gates as is shown in the Figure 11, because of the Cosine-term which pops up in its Hamiltonian. When doing sampling this way, we essentially assume the function stays constant for the time-span $[t, t + \delta t]$, which it doesn't.

In order to account for the fluctuation of the Hamiltonian during this time span, and improve the estimation of the Hamiltonian for this time-span, instead we sample at both $H(t)$, as well as $H(t+dt)$ and average between the two values we get.

This makes it so the final ideal Hamiltonian becomes:

$$H'_{ideal}(t) = \frac{1}{2} \cdot (H_{ideal}(t) + H_{ideal}(t + \delta t))$$

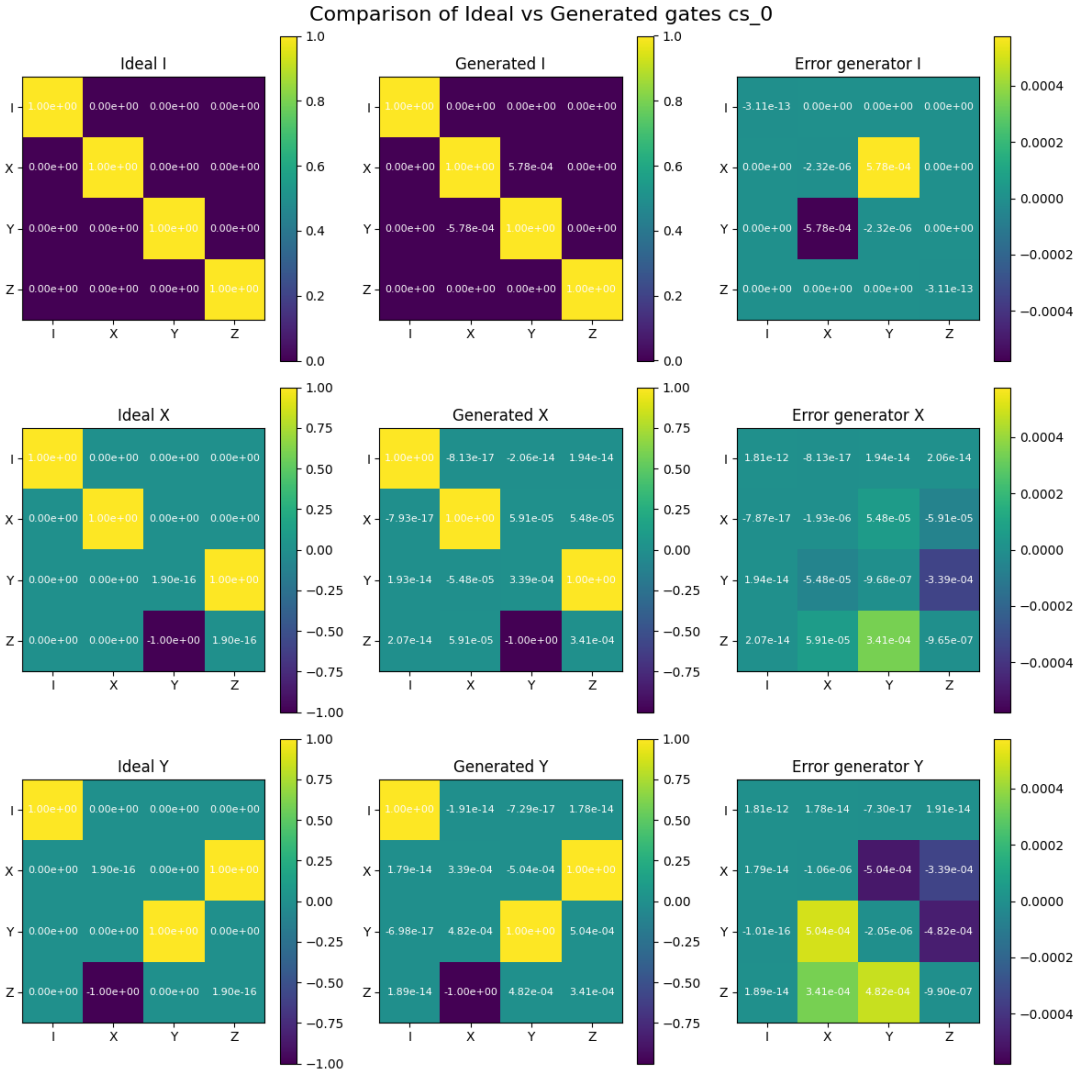


Figure 12: Gate generation for 'ideal' gates using updated ideal Hamiltonian, errors significantly decrease when compared with the previous generation scheme.

As can be seen from Figure 12, this greatly improves the unitaries we get when trying to generate ideal gates. With the ideal version showing as much as a factor 10 improvement in terms of absolute error. This in turn makes the gates which are affected by noise more representative, since we are now sure the noise is dominating in terms of generated errors.

4.2.4 Correlating gates together over time

We start by generating a long string of correlated noise based on the scheme described in Section 4.1. As described previously, each gate in the sequence starts out we begin with the ideal Hamiltonian, which defines the gate's intended operation without any noise. To introduce the noise, we then superpose the noise signal along the z-axis by incorporating the multiplication of the noise profile with the Pauli Z operator.

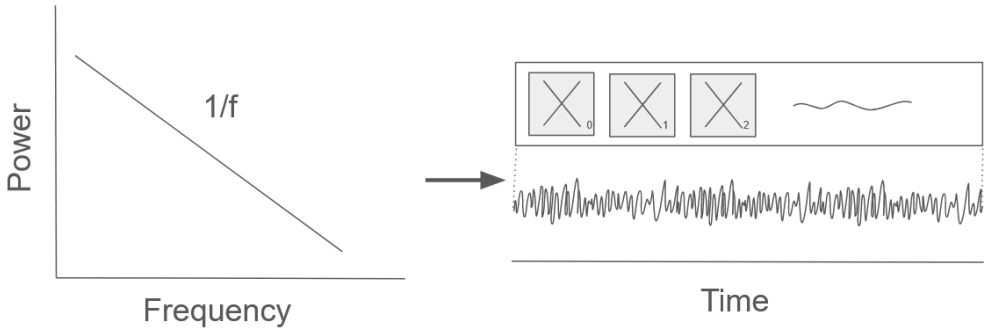


Figure 13: Simple representation of converting a PSD into a time-correlated sequence of gates

This method allows us to produce a long string of time-correlated gates, which will be used in subsequent stages. However, challenges arises when scaling this approach. For a single GST experiment, we require gate sequences with lengths in the order of millions. Since we're generating the noise all at once, which is a fundamental consequence of the way we generate it. Since sample sizes are in the order of 10.000 samples per gate, and we need millions of gates in order to run a full experiment of GST, and we have to generate all the noise at once in RAM, storing the amount of generated data necessary for the noise corresponding to a single long gate sequence proves to be completely unfeasible. This is why the approach has been modified in order to overcome this memory bottleneck.

4.2.5 Longer gate-generation schemes

To address this limitation posed by memory constraints then, we have developed a slightly different noise generation scheme. This new approach involves segmenting the noise generation process to reduce the immediate memory load as is shown in Figure 14. The new scheme generates noise according to the PSD uniquely on the gate level, this so-called fast noise, $X_{\text{fast}}(x)$, is uncorrelated between the gates and simply gets generated uniquely per gate. Then, in order to correlate the gates, so-called 'slow-noise' is generated, $X_{\text{slow}}(x)$, which is noise on the gate-sequence level. This means every gate is correlated only by using

a static noise signal added together with a non-static uniquely generated noise signal.

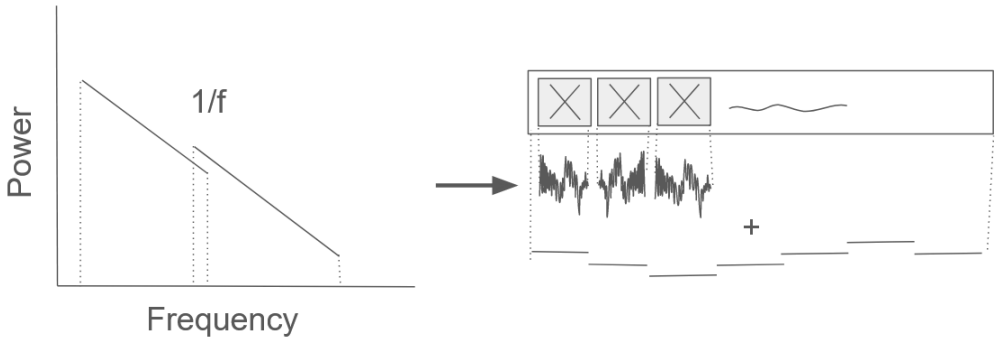


Figure 14: Schematic showing how the noise is split up into a 'fast' and a 'slow' component, and these are used in combination to generate the final set of gates.

Splitting the noise in two parts like this however does come with its own challenges, one of which is the exact way in which the noise should be scaled, since we're now adding together two signals, we need to be able to make sure that they form a larger coherent whole and the power of each of the respective signals is balanced according to the PSD. By taking the integral of the PSD over all frequencies, the total energy of the noise signal can be calculated, this leads to the follow expression:

$$\int_{1/N}^{1/\delta t} S(f)\delta f \cdot X(t) = \int_{1/N}^{1/T_g} S(f)\delta f \cdot |X_{\text{slow}}(x)| + \int_{1/T_g}^{1/\delta t} S(f)\delta f \cdot |X_{\text{fast}}(x)|$$

Leading to the expression for our resulting signal $X(t)$ which looks as follows:

$$X(t) = (\int_{1/N}^{1/\delta t} S(f)\delta f)^{-1} \cdot (\int_{1/N}^{1/T_g} S(f)\delta f \cdot |X_{\text{slow}}(x)| + \int_{1/T_g}^{1/\delta t} S(f)\delta f \cdot |X_{\text{fast}}(x)|)$$

With $|X_{\text{slow}}(x)|$ and $|X_{\text{fast}}(x)|$ being the normalized sequence-level and gate-level noise signals respectively, T_g being the duration of one gate, and dt being the length of a single time step in the simulation. This relationship follows from simply normalizing both signals, and integrating over the PSD in order to calculate the total power that each of the signals should add to the overall noise. The signals are then scaled accordingly.

By implementing the noise generation scheme like this, we allow ourselves to generate much much longer sequences of gates, something which we need in order to run GST experiments with sufficiently many shots per circuit.

4.3 Circuit generation

Using PyGSTi, we initiate standardized GST experiments by specifying various options that influence the design and complexity of the circuits. This process allows for the generation of a comprehensive list of circuits tailored to our specific GST parameters. The relevant options we can choose during this stage are outlined below.

4.3.1 Fiducial Pair Reduction schemes

Fiducial pair reduction (FPR) schemes are a crucial feature of our software, designed to optimize the computational efficiency and experimental tractability of GST experiments. Users can choose from several FPR schemes, each with its own advantages and suitable applications:

1. **No Fiducial Pair Reduction (No FPR)**: This is the most straightforward option, involving no reduction in the set of fiducial pairs. While it results in overcomplete and potentially lengthy experiments, it ensures that no potential information is lost, making it ideal for comprehensive analysis where computational resources are not a constraint.

2. **Per Germ Fiducial Pair Reduction (PFPR)**: PFPR intelligently selects fiducial pairs to be removed for all powers of a given germ but does so uniquely per germ. This approach acknowledges that different germs amplify different aspects of the model space, thus requiring tailored fiducial pair reductions. PFPR is less computationally demanding than global schemes and offers a practical balance between experimental completeness and efficiency.

4.3.2 Number of shots

The number of shots—or the number of times each circuit is run in an experiment—affects the statistical reliability of the GST results. While there is no one-size-fits-all answer for the ideal number of shots, our software allows users to specify a custom number based on their specific needs. Unless specified otherwise, the bulk of results in the thesis have been gotten with a number of shots of 1000. Meaning every circuit within a GST simulation is ran a 1000 times.

4.4 Circuit simulation

For every gate in our gate set, we have to generate a corresponding time-dependent gate sequence. This involves everything that was described previously, repeated for each of the quantum gates that are part of the gate set. Each gate is then stored and can be indexed by the time at which it should be applied, allowing for precise simulation of the quantum circuit.

4.4.1 Initialization of the Quantum Circuit

Every circuit simulation begins with the quantum system in its ground state. In the case of a single qubit, this state is represented as $\rho_0 = |0\rangle\langle 0|$. This initialization assumes perfect state preparation, which means the qubit is precisely in the desired starting state without any deviations or errors.

4.4.2 Gate Application and State Evolution

As the simulation progresses, we systematically apply each gate from the generated sequences to the quantum state. During simulation we keep track of a global time variable t, this simply gets incremented for every gate that we apply as part of our simulation. For every circuit in the simulation, we apply its gates in order, making sure to get the gate from the gate sequence from the appropriate time, like so:

$$\rho_n = U(n \cdot \delta t)\rho_{n-1}U^\dagger(n \cdot \delta t)$$

Where U(t) is the unitary transformation corresponding to the type of gate we're currently evaluating as part of the current circuit at time t , and ρ_{t-1} is the state of the system just before this gate is applied. This process iterates through all the gates in the circuit, in order, updating the state ρ at each step.

4.4.3 Measurement and Outcome Probability

Upon completion of the circuit, the final quantum state must be measured to obtain the outcome of the computation. Since all measurements are performed along the z-axis, the final state ρ is projected accordingly:

$$Pr(\rho) = Tr(\sigma_z\rho)$$

Where $Pr(\rho)$ is the probability of measuring the qubit in the ground state. From this probability the final outcome of the experiment is then decided by taking a value sampled from a uniform distribution between [0, 1] and comparing it to the measured probability. In other words, let $y \sim \text{Uniform}(0, 1)$ Then, the output of our circuit after measuring is given by:

$$\text{Measurement} = \begin{cases} 1 & \text{if } y \leq \Pr(x) \\ 0 & \text{otherwise} \end{cases}$$

4.5 GST Analysis

For the actual GST analysis, as mentioned before, our software utilizes the PyGSTi software. This section describes how we use PyGSTi for GST analysis, how we generate reports, and the additional tools we've developed to enhance data extraction and interpretation.

4.5.1 Utilizing PyGSTi for Analysis

After running the simulations, the resulting data is processed through PyGSTi to evaluate the performance and reliability of quantum gates within the context of our simulated environment. Once the simulation data is ready, PyGSTi is employed to analyze these results. PyGSTi's algorithms assess various metrics and parameters that describe the fidelity and accuracy of the quantum gates used in the simulations. Usually PyGSTi's capability to generate comprehensive HTML reports that contain detailed information about the GST analysis of our run is enough, since people are usually interested in the results of a single run. These reports are visual and interactive, making it easier to navigate through the data and understand the intricate details of the GST findings. However, we find for our purposes, we are mostly interested in how consistent the data that PyGSTi reports is, to this end, we had to come up with a way of extracting the generated data and aggregating it over multiple runs of GST for a more statistically sound result.

4.5.2 Data extraction, aggregation and analysis

Although PyGSTi provides extensive data analysis capabilities, extracting specific data from multiple runs for comparative analysis isn't directly supported. To address this limitation, our software supports extracting the data from these generated reports that PyGSTi produces in order to aggregate the data for use in statistical analysis.

In order to calculate the baseline with which to compare the results we get back from GST, we calculate some key properties from the generated sequences of gates. Maybe the most crucial of these is the Error Generator which GST also reports directly, the error generator L for an ideal gate G_0 and a noisy gate G' is defined as

$$G' = e_0^L.$$

By putting the gates from our generated sequence into super-operator form using the Kronecker product, we can average over them to get the average gate representation in super-operator form:

$$G' = \frac{1}{T}\sum_{n=1}^{T} U(n) \otimes U^*(n)$$

By inverting the equation above we can then extract the error generator from the sequence of gates. We can directly compare this with the results from our GST runs, offering a cohesive view of the discrepancies between the actual and reported performance of the quantum gates.

5 Results

In this section, the various simulated experiments conducted during this research are explained in detail. Each experiment's setup, results, and a brief discussion

are provided to highlight the key findings. The goal of this thesis is to investigate the dynamics of Gate Set Tomography (GST) in settings with non-Markovian noise dynamics. While GST is known to be reliable in settings where the noise is Markovian, little is known about its performance in environments with noise that exhibits non-Markovian characteristics. This thesis aims to explore this knowledge gap.

To achieve this goal, we implemented a Python library capable of simulating a large number of GST experiments and aggregating the data over multiple runs to observe emerging patterns. The experiments each go into different settings for GST, such as varying the maximum circuit depth, the number of shots, and batch sizes, among others. Each of these settings was chosen to investigate specific aspects of the performance and accuracy of GST using the software package PyGSTi.

For each experiment, the methodology is clearly outlined to ensure reproducibility. The results are then presented, accompanied by graphs and tables for clarity. Following the presentation of results, each experiment includes a small discussion that interprets the findings, addresses any anomalies, and explores the implications. This approach allows for a thorough examination of how different parameters affect the performance of GST.

The final discussion at the end of the thesis will integrate these individual findings, providing a comprehensive overview of GST's effectiveness under different parameters.

5.1 Baseline Experiment

The first experiment is done in order to clearly show the effects of non-Markovian noise on a simple GST-experiment. For this we have setup a basic experiment consisting of a basis set consisting of a single-qubit X, Y, and Identity gate (short XYI), all subjected to noise described by a power spectral density $S(f) = \frac{10^{-6}}{f}$. In which the prefactor "10^{-6}" directly scales the variation of the noise, and is hence quadratically tied to the standard deviation of the resulting noise.

To establish a baseline that shows purely Markovian noise, we shuffle the order of the gates. By accessing the gates in a random order, we remove any time-correlated effects in the gate set. This shuffled gate sequence is then used in a GST experiment to create a baseline under Markovian noise conditions.

The output of this baseline experiment is compared to the output of an experiment using the same gates, but with their original order preserved, thereby retaining the non-Markovian characteristics. This comparison allows us to clearly distinguish the effects of non-Markovian noise on the GST experiment.

5.1.1 Experimental Setup

The experiment employs a gate set specifically configured for a single qubit, including the following operations:

- **State Preparation**: Initialization is strictly in the $|0\rangle$ state.

- **Quantum Gates**: The gate set comprises the $R_x(\pi/2)$, $R_y(\pi/2)$, and the Identity (I) gate, referred to within GST as X, Y, and I respectively.
- **Measurement**: All measurements are conducted solely in the Z-axis.

This Gateset is included as a standard package in the PyGSTi software, and will be referred to as the XYI-gateset in the rest of this paper.

Furthermore, all results were obtained by averaging the results of a 100 separate GST runs. This is done to be able to for example extract the average output, as well as the standard deviation of this output.

5.1.2 GST Output Comparison

First, we present the results of the GST experiments under approximately Markovian noise conditions. Following this, we will show the effects observed in the presence of non-Markovian noise. The results show the calculated baseline error generators as well as the GST reported error generators.

Markovian noise
In the figure below, looking at the left column, which shows the error generators calculated straight from the gate set, it can be seen that the noise affects the Identity gate about twice as much compared to the X and Y gates. In the figure below, the Calculated Error and the GST reported Error are given. It can be seen that for all of the gates, GST does a relatively good job reporting the different errors for each of the gates.

Figure 15: Markovian noise: Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

Non-Markovian noise

When looking at the Non-Markovian case, it can be seen that even though the calculated errors are the same, GST seems to be influenced by the correlation of the noise in a significant manner. The reported error for the Identity gate comes out much higher, and the X and Y gates are steadily under-reported. The consistency of this is shown in the third column, in which you can see the standard deviations for each of the error generators. It can be seen that, on the whole, GST is quite confident in this estimation.

Figure 16: Non-Markovian case: Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

Fidelity comparison

To further illustrate the differences, the table below presents the fidelities for the calculated error generators, as well as for both the Markovian and non-Markovian cases. The effects of the Non-Markovianity of the gates clearly influence the results.

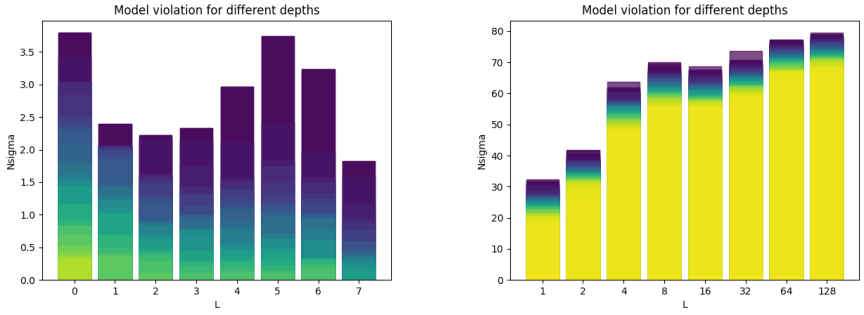
Table 1: Comparison of Ideal and GST Reported Entanglement Infidelities

Gate	Ideal Infidelity	GST Markovian	GST Non-Markovian
I	3.77%	3.76%	9.97%
X	3.16%	3.11%	1.02%
Y	3.14%	3.06%	1.10%

5.1.3 Model Violations

Model violations are assessed over 100 different runs of GST to determine the consistency and reliability of GST under both Markovian and Non-Markovian noise conditions:

Figure 17: Model violations for 100 GST runs



(a) Markovian case

(b) Non-Markovian case

As expected, the model violations for the Markovian case are pretty low. This backs up the idea that significant model violations mainly come from non-Markovian noise. We see violations ranging from 0 to 4, which is generally considered good, and shows minimal deviation from what we expect under purely Markovian conditions. This tells us that GST works well when Non-Markovian influences are out of the picture. The variability we see is likely due to the inherent randomness in quantum measurements. With more experimental runs or "shots," these minor model violations should decrease even further.

On the other hand, the non-Markovian case shows much higher model violations, ranging from 30 to 80, and these values increase with circuit depth. This big jump in violations compared to the Markovian case clearly signals non-Markovianity. These higher violations indicate the presence of non-Markovian noise and point out the limitations of GST when such noise is involved.

5.1.4 Discussion of Baseline Experiment

The data illustrates several key points regarding the performance and accuracy of GST in quantifying gate errors under both Markovian and non-Markovian noise conditions:

- **Reporting accuracy:**
  - **Markovian Case:** The results for all the gates are reported with quite a high level of accuracy, closely matching the calculated baseline. This is especially seen in the table of fidelities, which shows the fidelity estimates are within 2% of the calculated result for each of

the gates. This indicates that GST effectively captures the behavior of the system under Markovian noise conditions.

  - **Non-Markovian Case:** There is a notable over-reporting of errors for the I gate, as well as a corresponding under-reporting for the X and Y gates. This suggests that GST is influenced by the Non-Markovian dynamics of the gates, leading to a transfer of erroneous dynamics from the X and Y gates over onto the I gate.

- **Model Violations:**

  - **Markovian Case:** The model violations are low, supporting the notion that significant model violations are primarily a consequence of non-Markovian noise. Violations ranging from 0 to 4 are considered acceptable, indicating minimal deviation from expected results under purely Markovian conditions.

  - **Non-Markovian Case:** The non-Markovian case shows significantly higher model violations, ranging from 30 to 80, with the value increasing as circuit depth increases. These higher violations clearly signal that non-Markovianity was detected, indicating that GST is affected by the presence of non-Markovian noise. It can also be seen that the model-violations go up for longer sequence lengths, this is an indication of the circuit not being fully dephased yet even at L=128, in which case model violations would go down.

- **Consistency:** The standard deviations seen for the error generators in the Non-Markovian case clearly show that GST is very confident in its predictions. However, the consistent overreporting for the I gate and underreporting for the X and Y gates suggest that the error in the results is fundamental and consistent, arising from non-Markovian effects rather than from randomness.

These findings raise concerns about GST's ability to accurately predict errors for all types of gates, particularly under Non-Markovian noise conditions. While it effectively identifies errors in the I gate under Markovian conditions, its performance in detecting errors in the X and Y gates, especially under non-Markovian conditions, is less reliable.

5.2 Comparing Noise Strength

As a first parameter to look at, we can look at the effects of how varying strengths of noise impact the results of GST under Non-Markovian conditions. Using the noise power spectral density $S(f) = \frac{10^A}{f}$, where A varies to demonstrate different noise intensities. This experimental setup is designed to explore the impact of pink noise amplitude on overall GST accuracy.

5.2.1 Experimental Setup

The experiment employs the same XYI gate set as in the baseline study, averaging the results over a 100 runs of GST. Each circuit within the GST experiment is executed 1,000 times, testing three different noise strengths: from low-intensity (A = -6) to high-intensity (A = -4) noise, with one run of intermediate-intensity (A = -5) noise in between.

5.2.2 GST Output Comparison by Noise Strength

The results for different strengths of noise are shown below. Each plot displays the calculated errors, the GST reported errors, as well as the standard deviations of these error generators.

Noise Level A = -6
The results for the lowest strength noise in this experiment are shown below.

Figure 18: A = -6, Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

At this lower noise strength, there is a systematic over-reporting of errors on the I gate and under-reporting on the X and Y gates, just like in the baseline experiment. Furthermore, it can be seen that the reported Standard Deviations for each of the error generators are quite low compared to these error generators themselves.

Noise Level A = -5

The results for the intermediate strength noise in this experiment are shown

below.

Figure 19: A = -5, Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

At this intermediate noise level, the discrepancy in over- and under-reporting between the gates diminishes, and GST predictions become more accurate. The error measurements for the X and Y gates are more reflective of the actual calculated errors, indicating improved accuracy in GST's predictive capability at this noise strength.

Noise Level A = -4

The results for the high strength noise in this experiment are shown below.

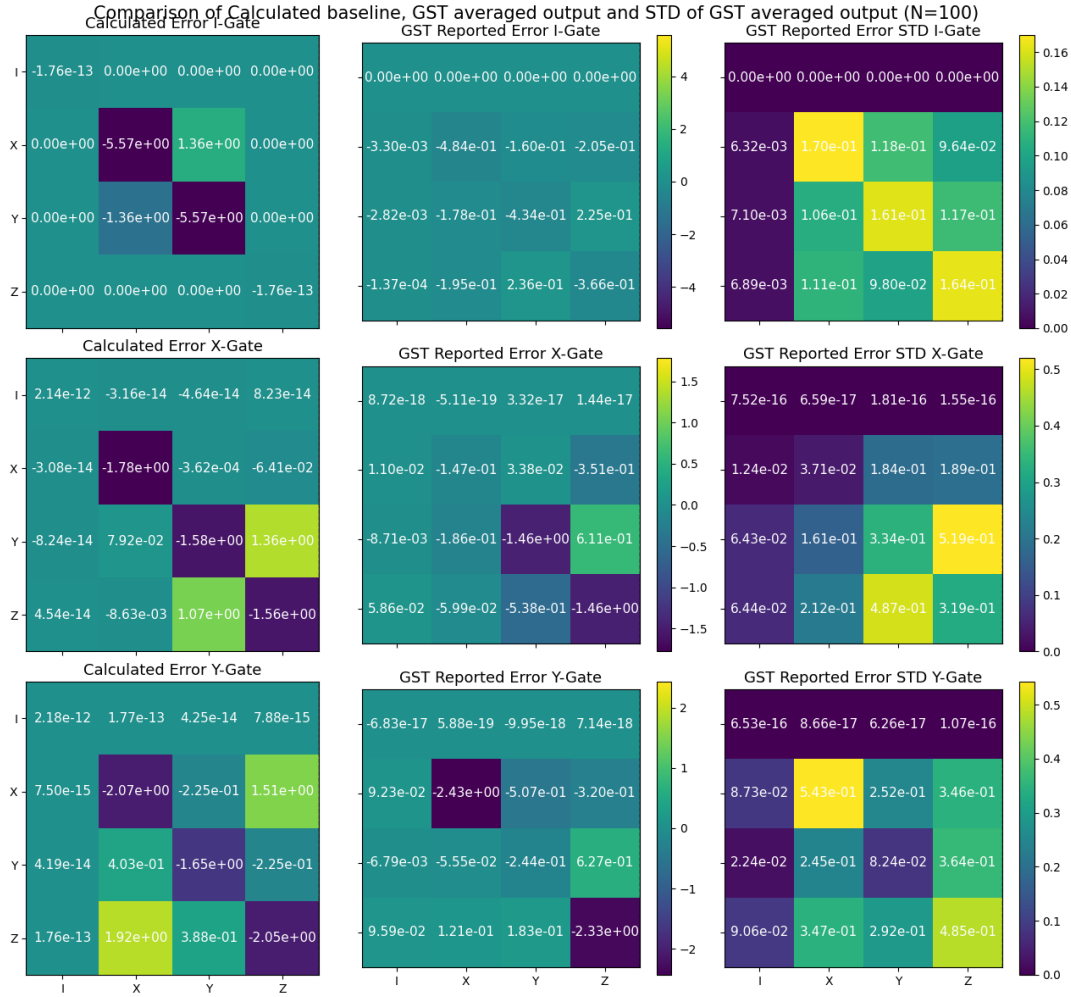
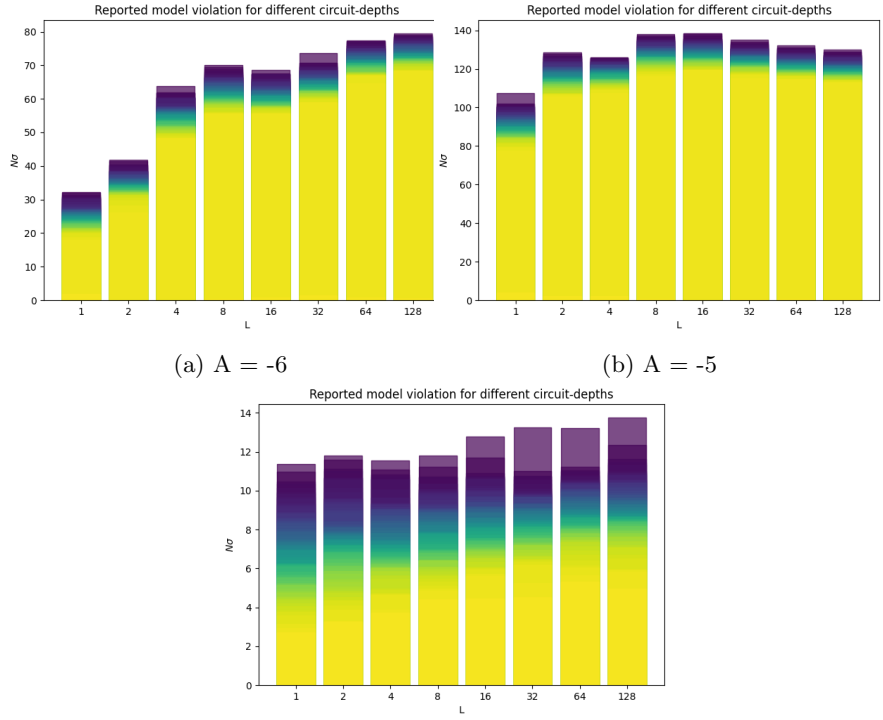


Figure 20: A = -4, Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

As noise strength increases further, the earlier over-reporting of the I gate vanishes, and the error measurements for both the X and Y gates become quite accurate. This suggests a potential optimal noise intensity for GST performance under Non-Markovian conditions, indicating that there is a sweet spot where GST predictions are most reliable.

5.2.3 Model Violations

Model violations are shown for each noise level and are briefly discussed in this section.



(a) A = -6

(b) A = -5

(c) A = -4

Figure 21: Model violations for different noise levels

As can be seen above, at the lowest noise level (A = -6), model violations are relatively low, especially for shallow circuit depths, indicating that GST performs well under these conditions. As the noise strength increases to an intermediate level (A = -5), the model violations increase significantly, reflecting the moderate increase in noise strength and suggesting a more challenging environment for GST accuracy. Interestingly, at the highest noise level tested (A = -4), model violations decrease again. This reduction is likely due to the overwhelming noise at this level, causing the circuits to become decohered, especially over longer sequences.

5.2.4 Discussion of Noise Strength Experiment

This experiment reveals a notable transition from low-strength noise, where the Identity gate errors are over-reported and the X and Y gates are under-reported, to high-strength noise, where the errors for the X and Y gates are over-reported

and the Identity gate is under-reported. The intermediate noise level of A = -5 yields the most accurate predictions across all gates, suggesting an optimal noise regime for GST. This "sweet spot" indicates a noise strength where GST is still effective under Non-Markovian conditions, with deviations observed at both higher and lower noise strengths affecting the accuracy of gate error predictions in an adverse way.

Particularly for the highest noise strength tested, A = -4, we observe a decrease in model violations. This phenomenon can likely be attributed to the overwhelming strength of the noise, which causes the circuits to become highly dephased over longer sequences, this hypothesis is supported by the results of the model violations for A = -5, in which, for larger depths of the circuits, the violations are already going down. In such conditions, the circuits' results become quite random; one can imagine that for a very long circuit with very noisy gates, the result of measuring at the end is essentially a 50/50 chance if the gates are noisy enough. This randomness suggests that while the model violations decrease, the actual usefulness of GST becomes limited under these extreme noise conditions.

5.3 Comparing Batch Sizes

Typically, GST circuits are executed multiple times to gather sufficient data for statistical analysis. The standard approach of running all required shots for one circuit before moving to the next can unintentionally amplify noise correlations for each one of the evaluated circuits, leading to skewed results. Therefore, the choice of batch size—how many times each circuit is consecutively run—plays a crucial role in the accuracy and reliability of the results. This experiment explores how different batch sizes impact GST outcomes, especially considering the potential correlation of noise between runs.

This investigation assesses whether an alternative interleaving method, where each circuit is run once per cycle before repeating the sequence, can effectively reduce these noise correlations. While this method is more complex to implement experimentally due to the challenges of frequent circuit initialization, it could potentially offer a more accurate depiction of gate behaviors by minimizing correlated errors.

5.3.1 Experimental Setup

The experiments use the XYI gate set, consistent with the previous experiments, with each GST run consisting of 1000 shots. The gates are generated with a given PSD of $S(f) = \frac{10^{-6}}{f}$. On the two extremes we have:

- Batch Size = 1: Represents a scenario where each circuit is run once per iteration, cycling through all circuits over 1000 iterations.
- Batch Size = 1000: Each circuit is run 1000 times consecutively before moving on to the next circuit in the experiment.

While the results were obtained for intermediate batch sizes as well, this section will only show the results of these two extremes, since this gives enough insight on its own.

5.3.2 GST Output Comparison by Batch Size

The GST outputs for both batch sizes are analyzed to observe the effects on error reporting and consistency:

Batch Size 1
The figure below, again shows overreporting of errors on the I gate and underreporting on the X and Y gates. The standard deviation of reported errors is notably low, indicating that GST is quite consistent in its assessments across different runs.

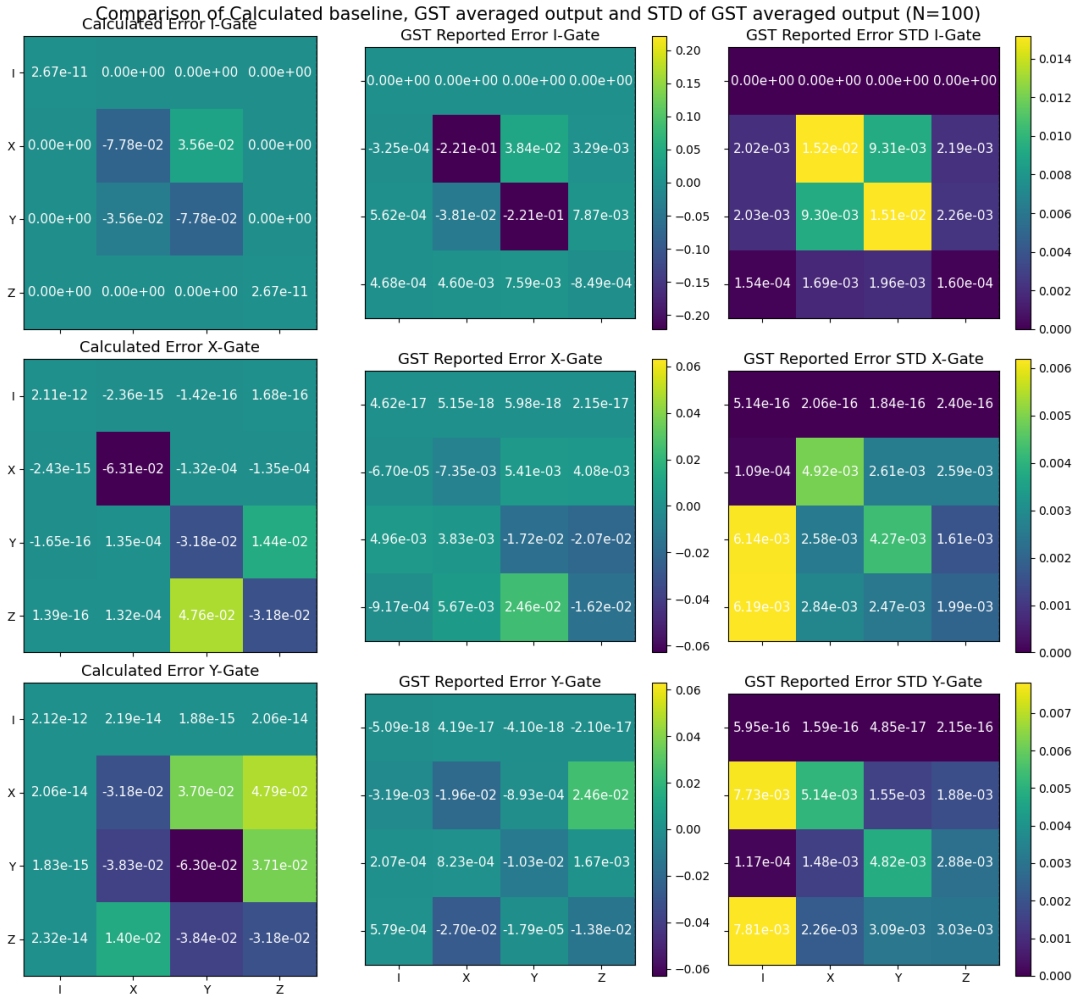


Figure 22: Batch size = 1, Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

Batch Size 1000

Similar trends in error reporting are observed as with Batch Size = 1, but with significantly higher standard deviations.

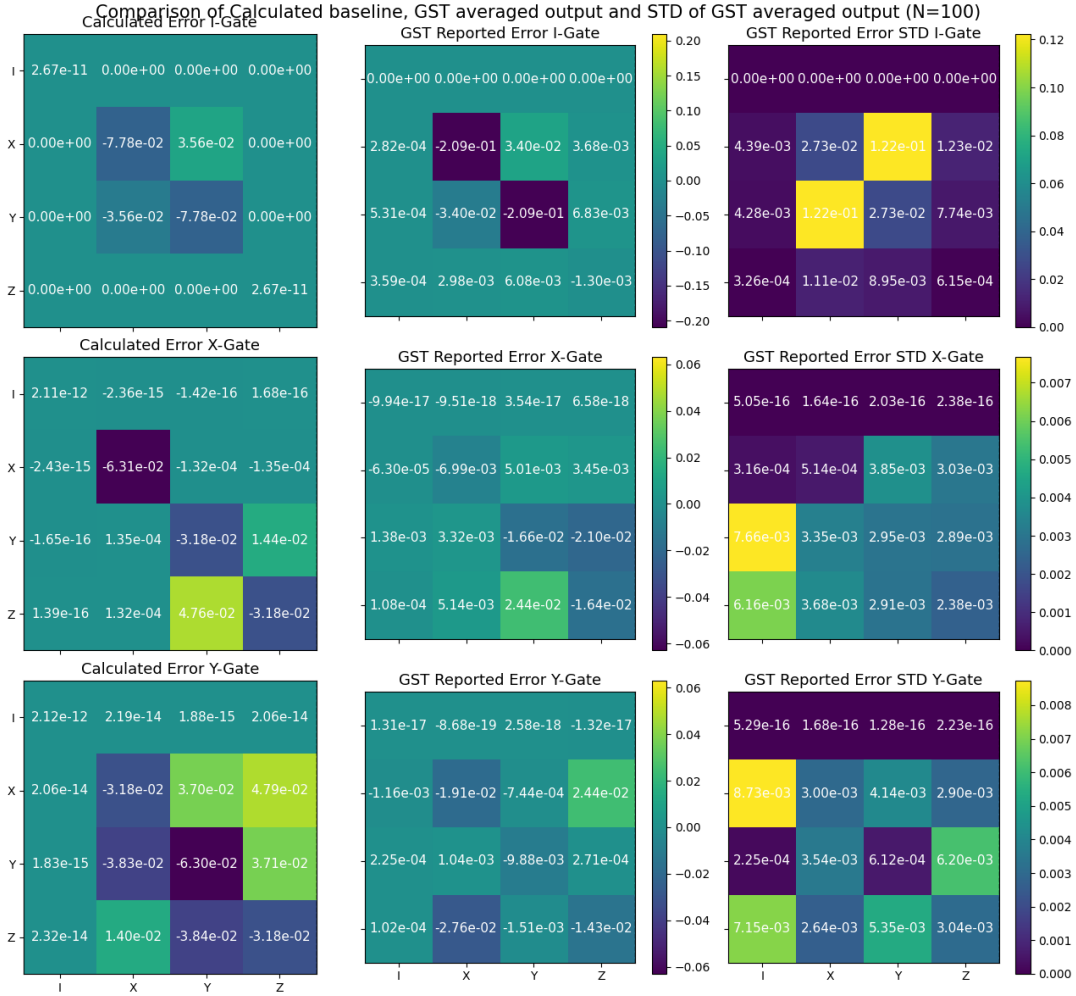
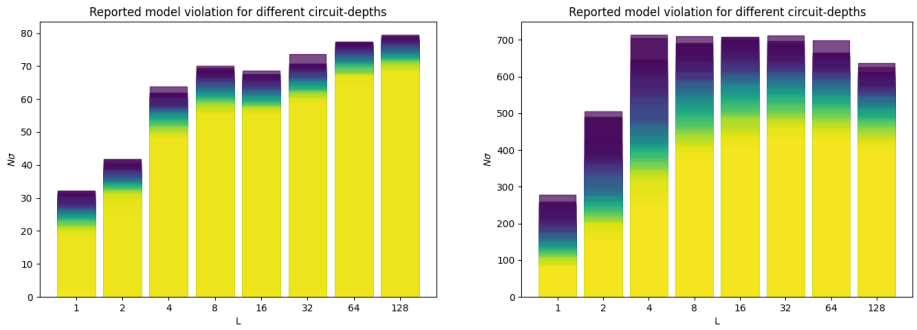


Figure 23: Batch size = 1000, Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

5.3.3 Model Violations

Model violations are also analyzed to assess how well GST fits the experimental data under different batch sizes.



(a) Batch Size = 1 (b) Batch Size = 1000

Figure 24: Model violations for different batch sizes

When comparing both runs, it becomes evident that model violations are lower in the interleaved (Batch Size 1) approach, with values ranging from 30-80. In contrast, model violations are significantly higher in the batched approach (Batch Size 1000), with values ranging from 300-800. This suggests that GST has a decreased ability to detect non-Markovianity in the interleaved run, while the higher violations in the batched approach indicate significant issues with noise correlation adversely affecting the GST fitting process. Interestingly, there is a small decrease in model violations for longer circuits, which may indicate that some dephasing effects which come into play over longer sequences.

5.3.4 Discussion of Batch Size Experiment

The data from the experiment reveals an intriguing scenario: although smaller batch sizes are theoretically supposed to enhance the quality of results by reducing noise correlations between different evaluations of a circuit, our findings suggest that the actual impact on GST output is minimal. The table below summarizes the reported infidelities for both batch sizes against a calculated baseline, providing a clear comparison of outcomes:

Table 2: Comparison of Ideal and GST Reported Entanglement Infidelities for Different Batch Sizes

Gate	Ideal Infidelity	Batch Size = 1	Batch Size = 1000
I	3.77%	9.69%	10.04%
X	3.16%	1.02%	1.00%
Y	3.14%	1.01%	1.08%

The results highlight a minor variance in the average outcomes of the GST experiments for the two strategies, suggesting that switching to a smaller batch size might not necessarily produce better data. This finding challenges the common notion that lower batch sizes inherently lead to more accurate assessments

by mitigating the effects of Non-Markovian noise through the decorrelation of successive experimental runs.

However, the experiment also underscores a significant difference in the consistency of the results produced. The standard deviations observed for larger batch sizes were notably higher than those for smaller batches. This increased variability implies that while the mean outcomes might be similar, the reliability and predictability of results are compromised when larger batch sizes are used.

The increase in model violations observed for larger batch sizes—a sign often interpreted as a negative impact on the GST's ability to fit data accurately—supports the notion that smaller batch sizes could be more desirable. Despite this, the actual discrepancies between the averages of GST results under both batch configurations were not as pronounced as one might expect, raising questions about the cost-effectiveness and practicality of implementing smaller batch sizes in actual experiments.

In summary, while the choice of batch size does not drastically alter the average reported data, it significantly affects the consistency of these results. Experimenters must weigh the benefits of potentially more stable results against the logistical and operational complexities introduced by smaller batch sizes.

5.4 Comparing Max Depth

GST generates circuits up to a maximum depth. This maximum depth can be specified, allowing germs to be repeated more often for better results. Lower maximum depths should provide less insightful results due to less data and less pronounced error amplification from repeated germs. However, lower depths are easier to run experimentally since the overall size of a GST experiment increases with its maximum depth.

5.4.1 Experimental Setup

In this set of experiments, we maintained consistency with our previous experimental conditions. The gates were generated with a noise PSD described by the function PSD: $S(f) = \frac{10^{-6}}{f}$. Furthermore, like in previous experiments, we used the XYI gate set for these experiments, as well as having each run consist of 1000 shots. Lastly, like before, each of the results was obtained by averaging the results of a 100 separate GST runs.

We tested the GST protocol across a range of depths to observe how the maximum depth impacts the results. The specific depths tested were 8, 16, 32, 64, and 128. By varying the maximum depth, we aimed to understand if indeed, deeper circuits, which repeat the germs more often, could potentially provide better estimates. This variation helps us explore the trade-off between the depth of the circuits and the quality of the GST results.

5.4.2 GST Output Comparison by Max Depth

Below, a comparison is seen of the GST reported results from L=8 up to L=128.

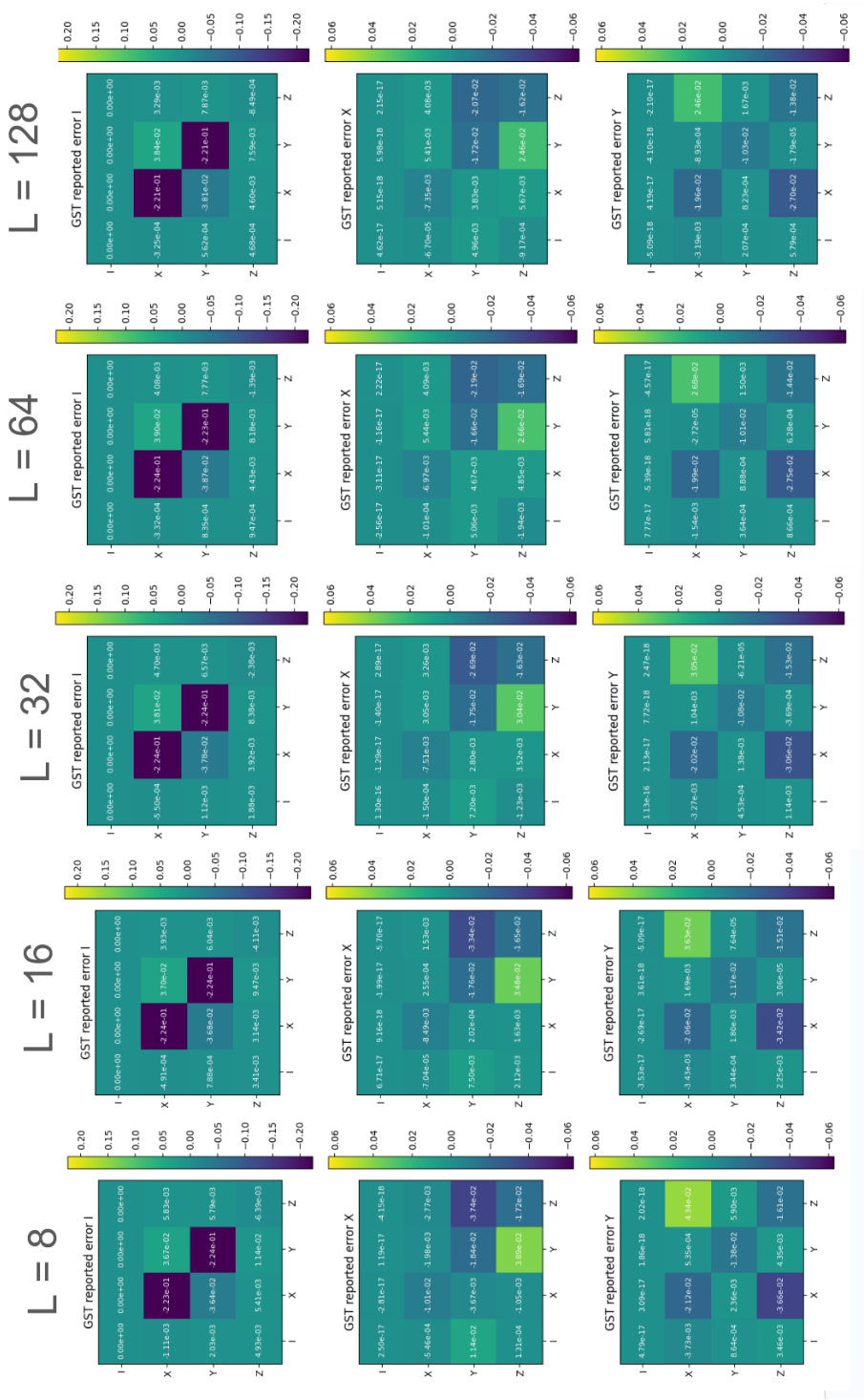


Figure 25: Compared GST estimated Error generators for different settings of Max Depth (L)

The errors seen for lower depths are gradually reduced in the larger depth experiments. Intermediate runs also show intermediate values for these error generators. In order to see if indeed, higher maximum depth runs yield better error generators, they must be compared to the reference calculated baseline. Let's isolate the runs for L=8 and L=128.

Max Depth = 8

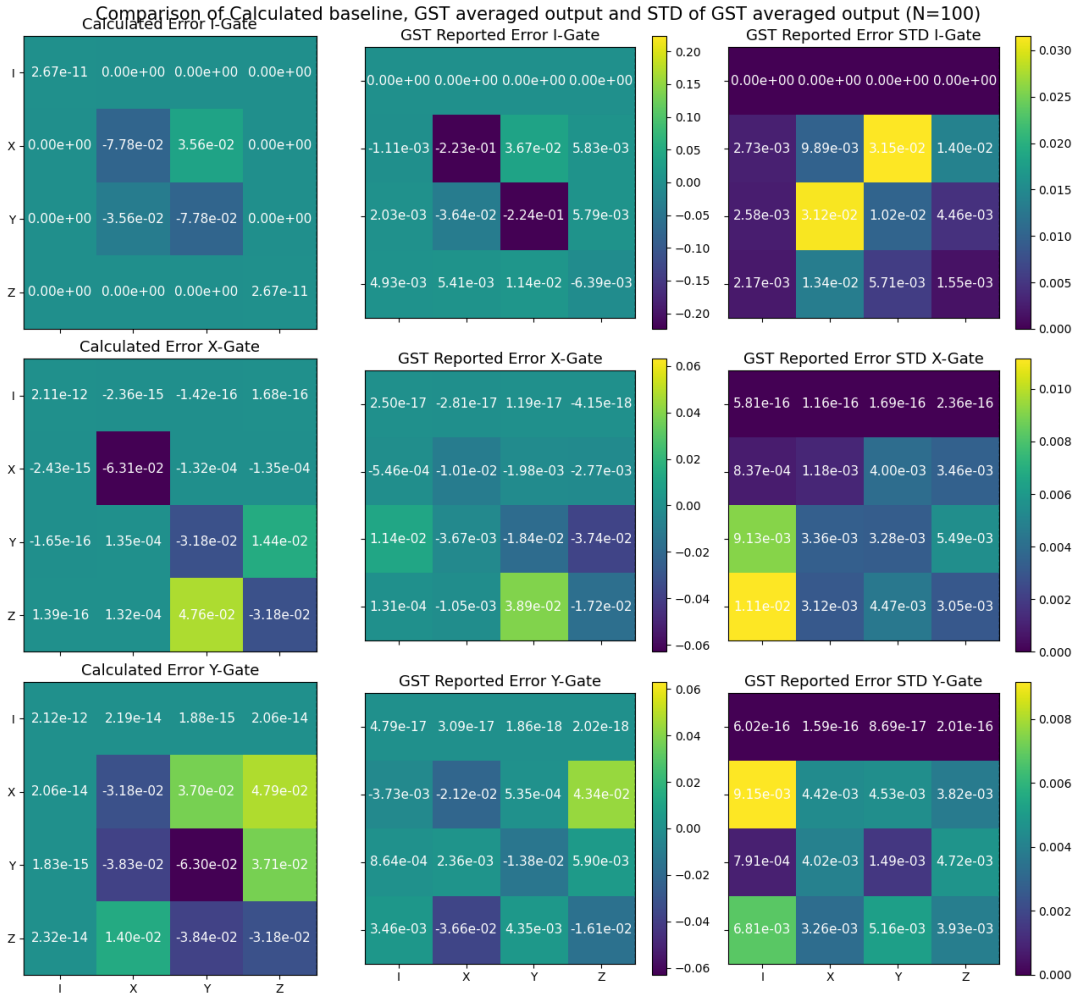


Figure 26: L = 8, Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

For L=8, there is, as can be expected at this point, systemic over-reporting of errors in the Identity gate and under-reporting errors in X and Y gates, with standard deviations around 1e-2.

Max Depth = 128

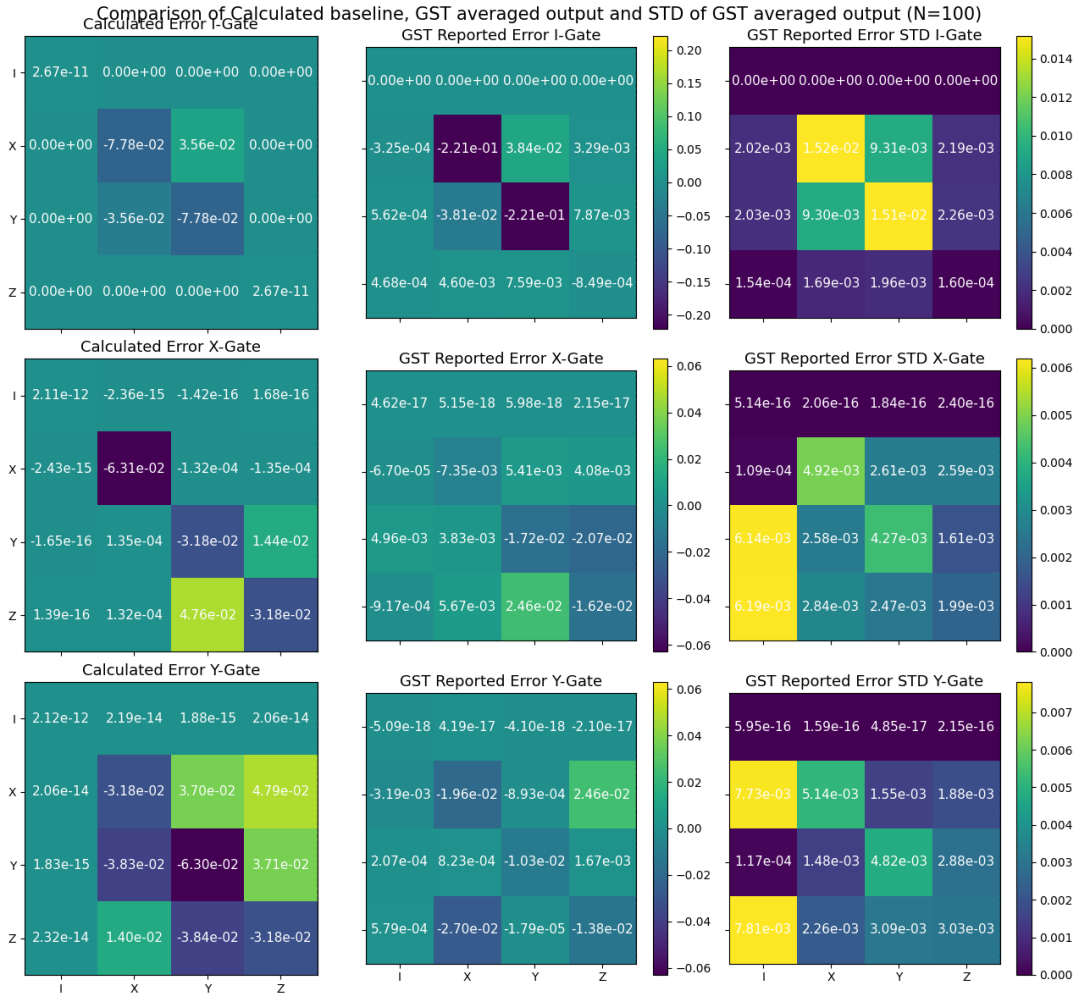


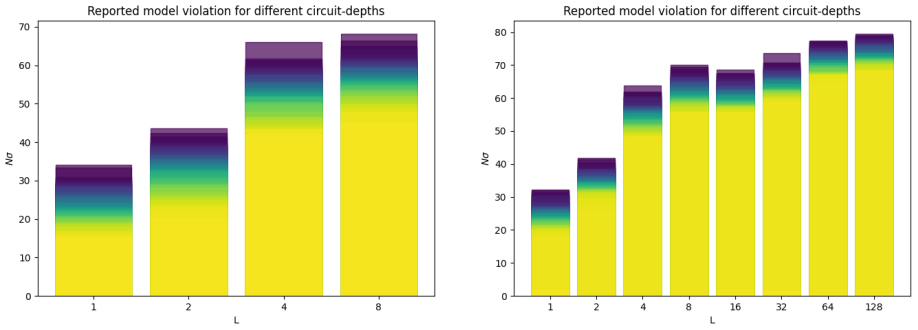
Figure 27: L = 128 Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

Similar error generators seen in the extremely shallow experiment (L=8) appear in the larger experiment (L=128). There seems to be a gradual fading of errors from L=8 to L=128. These errors seen do not consistently correlate with errors in the calculated baseline. While it is good that these phantom errors disappear, the end result does not look much better compared to the actual baseline for L=128. One last difference is that the standard deviations

tend to get smaller the larger the experiment grows, this makes sense, since more data should lead to more consistent results.

5.4.3 Model Violations

Model violations are also analyzed to assess how well GST detects non-Markovianity for different circuit depths.



(a) L = 8 (b) L = 128

Figure 28: Model violations for different maximum depths

The model violations graph for L=8 only shows 4 bars, as these bars in the graph correspond to the model violations for different depths. It makes sense, therefore, that the model violations for these specific depths are the same for both runs. Furthermore, we observe a gradual increase in model violations as depth increases, indicating that larger maximum depths have more potential for spotting non-Markovianity in the simulation.

5.4.4 Discussion of Max Depth Experiment

The results across different maximum depths are largely similar. This can be observed in the table below, which lists the infidelities for the baseline as well as for the L=8, 32, and 128 runs.

Table 3: Comparison of Ideal and GST Reported Entanglement Infidelities for Different Depths

Operation	Ideal Infidelity	GST (L = 8)	GST (L = 32)	GST (L = 128)
I	3.92%	10.19%	10.11%	9.97%
X	3.10%	1.16%	1.05%	1.02%
Y	3.19%	1.30%	1.17%	1.10%

The reported infidelity barely changes from the very shallow runs to the very deep runs. The L=8 run is a very simple, small experiment, so we should not

expect very good results. The idea is that the results would get significantly better for longer runs; however, we are not seeing that either. For L=128, the results are still very much off. This may be due to the assumption of Markovianity, causing the error amplification, which normally greatly improves the estimation, to fall short. Instead, it gives a washed-out version of our short experiment, where none of the actual germs are amplified consistently enough to improve the estimation in the end.

The results we see are similar to those in the Batch Size Experiment, in the sense that longer runs do not necessarily improve the results significantly but do lead to more consistent GST experiments. This can be seen from the standard deviations of both runs, where the shallow run (L=8) has standard deviations about a factor of 2 higher than those of the L=128 run. This may indicate that there is a trade-off between consistency and computational complexity.

5.5 Comparing Number of Shots

When running a GST experiment, every circuit in the experiment is run multiple times. This number is known as the amount of shots and can be freely chosen for a given GST run. More shots generally lead to more consistent and accurate results, as the measured outcomes will better approximate the actual underlying probabilities. However, this comes at the cost of having to perform more circuits on the quantum device, which can be expensive. This experiment aims to determine how many shots are sufficient to obtain representative results.

5.5.1 Experimental Setup

The gates were (once again) generated with a noise Power Spectral Density described by the function $S(f) = \frac{10^{-6}}{f}$. For these experiments, we used the XYI gate set and a maximum depth of 128. The number of shots tested were: 10, 50, 100, 200, 500, and 1000.

5.5.2 GST Output Comparison by Number of Shots

We start by comparing the GST output for different settings of the number of shots (N).

Error Generators

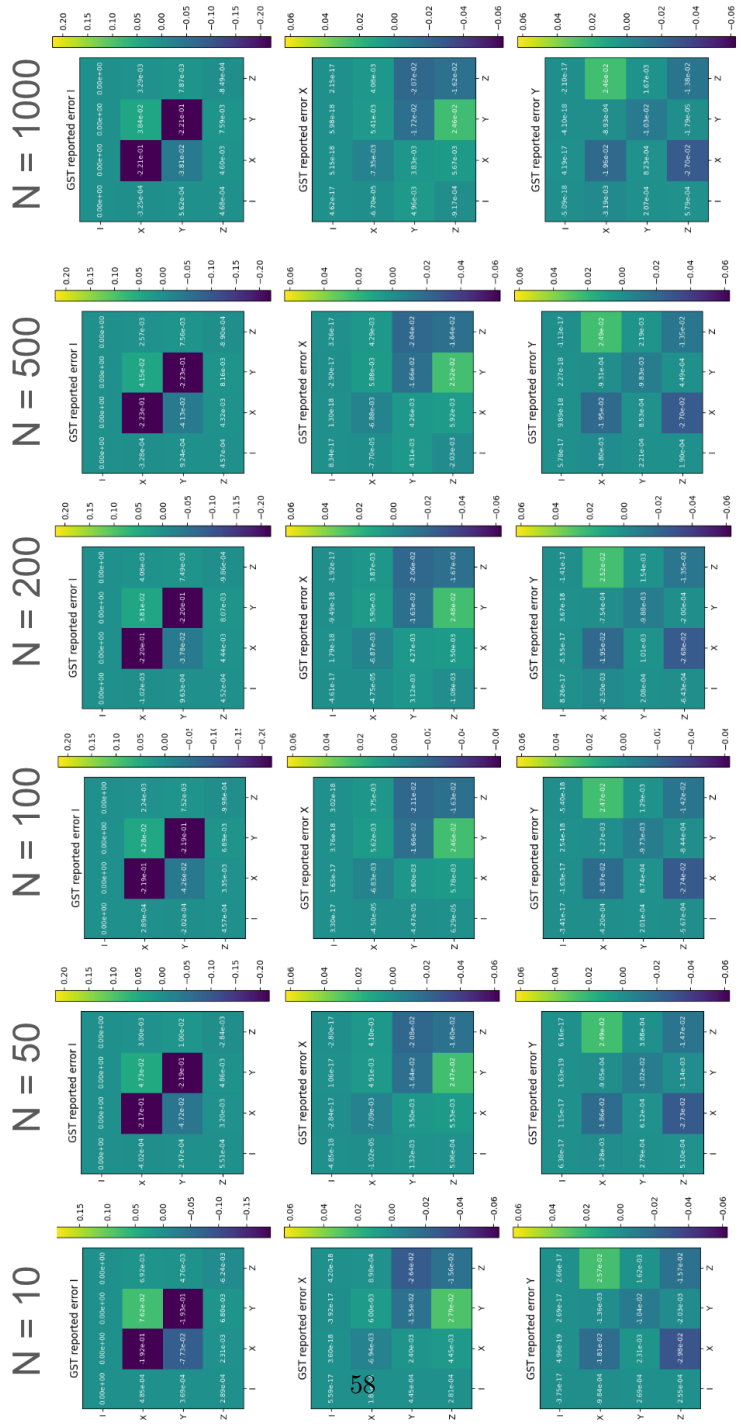


Figure 29: Comparison of the reported GST Error Generators for different settings of N

As shown in the figure above, the results are largely of the same order, especially for the X and Y gates. Not much difference is seen with increasing the number of shots. This consistency is expected since the underlying probabilities remain the same, and running enough iterations of GST should, in principle, yield similar results. However, the average for the Identity gate takes some time to settle, with lower values for N=10 and N=50, and only stabilizing for runs with N$\geq$100.

Standard deviations

Looking at the standard deviations, we observe the following:

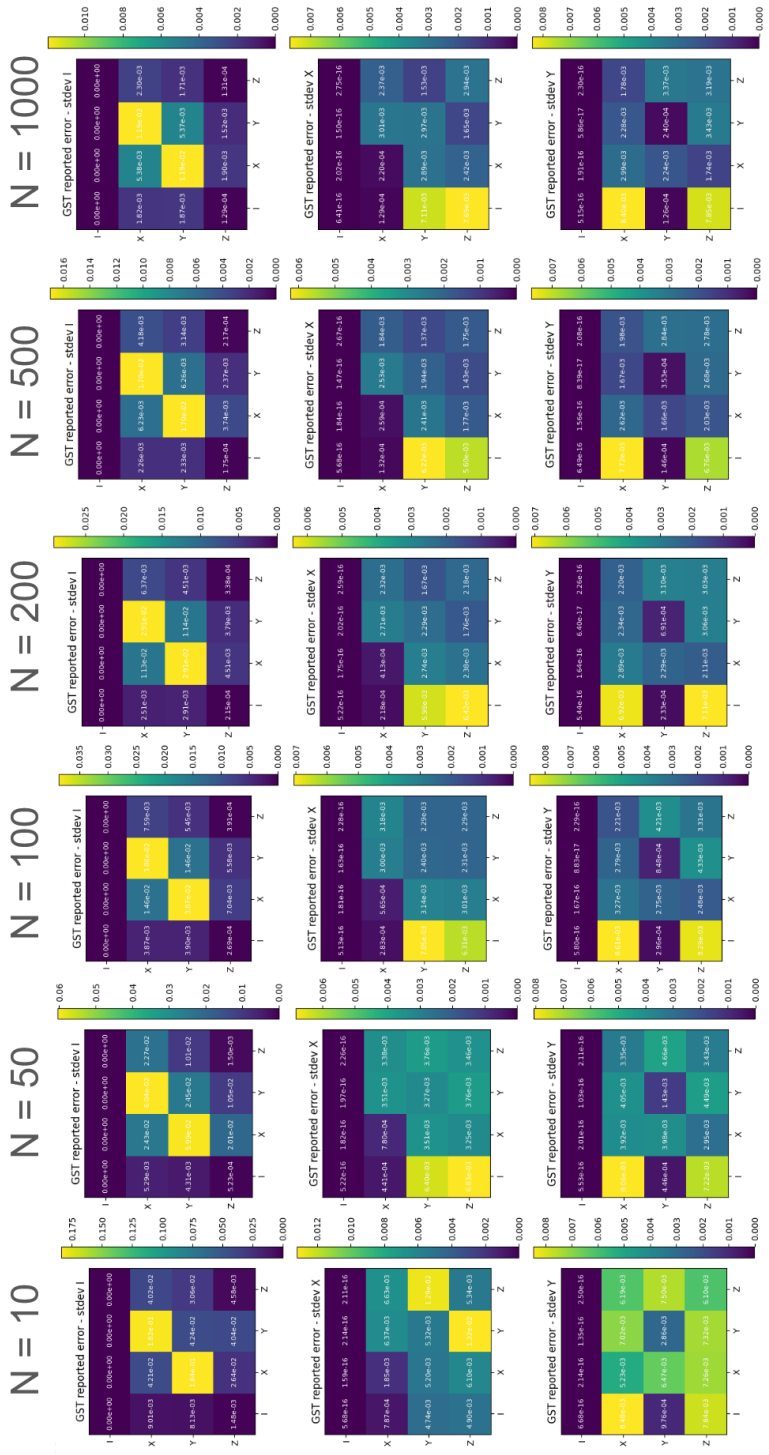
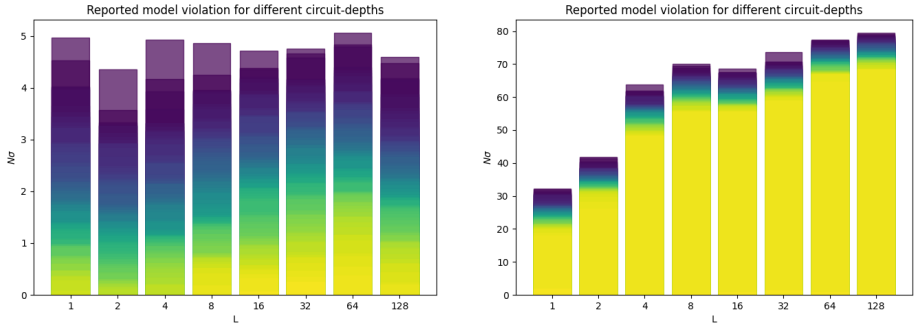


Figure 30: Comparison of the Standard Deviation of the Error generators for different settings of N

As can be expected, the standard deviations are much higher for the runs involving fewer shots. For the lower shot runs, particularly for the X and Y gates, the results show a much broader distribution over the different entries in the error generator. In contrast, higher shot counts make for more reliable runs. The difference in standard deviations between the runs with N=10 and N=1000 is as much as a factor of 20. For the X and Y gates, the reported errors are more consistently placed on specific entries in the error generator with higher shot counts. However, this effect is less pronounced for the I gate. Instead for the I gate, GST appears to become increasingly uncertain about which entries of the Error Generator to blame as the number of shots increases.

5.5.3 Model Violations

Model violations are analyzed to assess how well GST detects non-Markovianity for different numbers of shots.



(a) N = 10 (b) N = 1000

Figure 31: Model violations for different numbers of shots

For a low number of shots (N = 10), low amounts of model violation are seen, all in the range of approximately 3-5. This usually indicates a very good fit, suggesting that there wasn't a significant amount of non-Markovianity detected.

In contrast, for a higher number of shots (N = 1000), the model violations are not only much higher but also more consistent, indicating a significant model violation has been detected. The model violations for N = 1000 are about a factor of 10 higher than those for N = 10.

5.5.4 Discussion of Number of Shots Experiment

Fewer shots do not significantly change the average results of GST. However, the consistency with which GST returns these results is much higher for runs with a higher number of samples This is seen especially in the standard deviations for the Identity-gate, which are about a factor 20 lower for the N=1000 run than they are for the N=10 run. This indicates a trade-off between complexity and consistency. In real-life experimental settings, it is important not to set

the number of shots too high, as this can lead to very lengthy and difficult to execute experiments. Conversely, setting the number of shots too low makes the results very unpredictable. For example, the results for N=10 show a standard deviation in the error generator that is higher than the actual entries in the error generator. This means that, even though on average the results might be good for N=10, in practice, the validity of these results would need to be verified by running more iterations of GST. In practice, experimenters must select a number of shots which strikes a balance between managing this inconsistency on the one hand, and the complexity of the overall GST experiment.

5.6 Comparing Gate Sets

Since not every quantum device supports the same operations, when running GST, there is the choice of which gate set to use for the experiment. One might also imagine a scenario in which a given device has a large gate set, and it might make sense to evaluate only parts of it. In this section, we compare the results of running the XYI gate set with a gate set in which the Identity gate has been removed. Since we observed a consistent over-reporting of the error in the I gate, the hypothesis is that removing it from the gate set might improve the estimates in the error of the X and Y gates.

5.6.1 Experimental Setup

In this set of experiments, we maintained consistency with our previous experimental conditions. The gates were generated with a noise PSD described by the function: $S(f) = \frac{10^{-6}}{f}$, and the maximum circuit depth was set to L=128 We used the same XYI gate set as in the baseline study, averaging the results over a 100 GST runs. Each circuit within the GST experiment was executed 1,000 times. We tested both the XYI gate set and the XY gate set, which are identical except that in the latter the Identity gate has been removed. This leads to an overall reduction of the GST-experiment, from a total of 1022 unique circuits, down to only 834 unique circuits.

5.6.2 GST Output Comparison by Gate Set

XYI-Gateset
The results for the XYI gate set are shown below. As expected, the familiar over-reporting of the I gate error is seen, and the errors in the X and Y gates are under-reported.

Figure 32: XYI-gateset, Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

XY-Gateset

The results for the XY gate set are shown below. Despite having no Identity gate, the same underreporting of errors in the X and Y gates is seen.

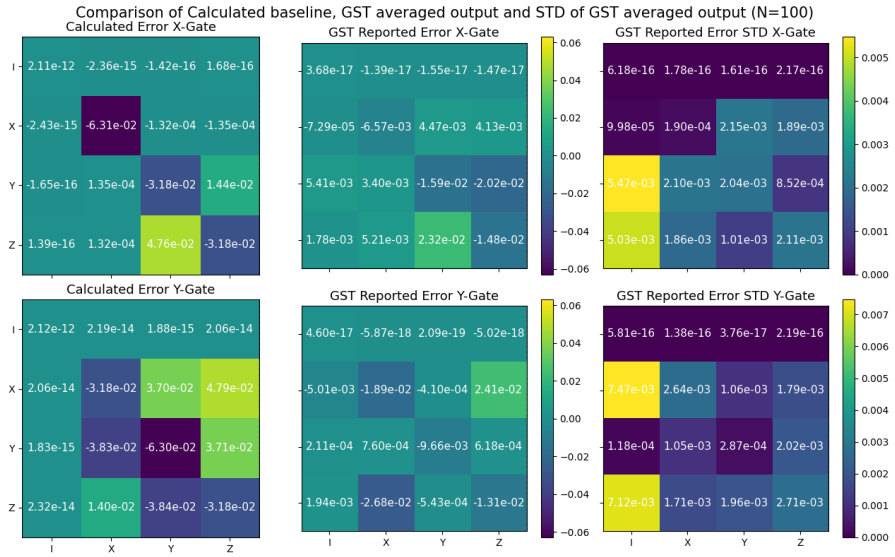


Figure 33: XY-gateset, Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

Infidelities

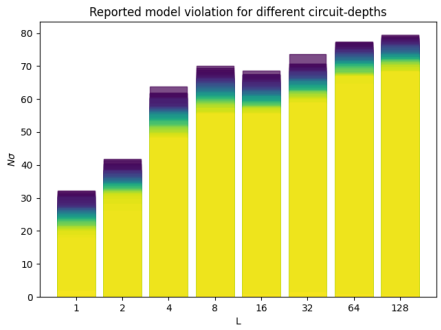
To further illustrate the similarity of the results, the Entanglement Infidelities are given below.

Infidelities	Calculated	GST XYI-gateset	GST XY-gateset
I	3.77%	9.97%	-
X	3.16%	1.02%	0.93%
Y	3.14%	1.10%	1.01%

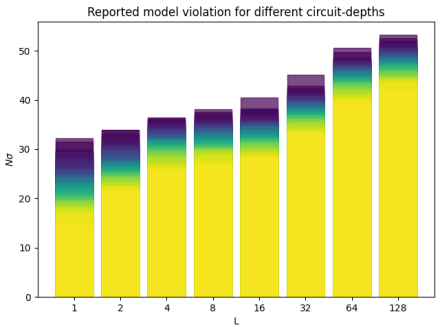
It can be seen that by excluding the Identity gate from the gateset, the results stay largely the same for the X and Y gates.

5.6.3 Model Violations

Model violations are analyzed for each noise level and discussed briefly in this section.



(a) Reported Model Violations for the XYI-gateset



(b) Reported Model Violations for the XY-gateset

Figure 34: Reported Model Violations for different gatesets

For the XYI gate set, reported model violations are in the range of approximately 30-80, with increasing model violations for increasing circuit depths. For the XY gate set, reported model violations are lower, and show up in the range of approximately 20-50, with increasing model violations for increasing circuit depths. The reported model violation for the XY gateset is consistently lower than that of the XYI gateset.

5.6.4 Discussion of Gate Set Experiment

Examining the error generators and reported infidelities, there seems to be an almost complete match of results between the XY and XYI gate sets. The differences in results are likely due to the random nature of the measurements, introducing some variability to the GST runs.

The standard deviation of the results matches quite closely as well, which is surprising given that the model violation is noticeably larger for the XYI gate experiment compared to the XY gate experiment. In previous experiments, there was a clear correlation between standard deviation and model violation, but this correlation is not observed here. Although the difference in model violation is relatively small, which might partially explain the lack of correlation, it does not fully account for it.

In general, excluding the Identity-gate from the experiment tends to lower the model violation. Interestingly, the slight decrease in model violation suggests that the Identity-gate may not significantly affect the model violation. The expected improvement of error estimation in the X and Y gates is not observed, indicating that gauge-fixing is not the primary source of the consistent over-reporting on the Identity-gate and under-reporting on the X and Y gates as was also seen in previous experiments.

5.7 Comparing Fiducial pair reduction

The purpose of Fiducial Pair Reduction (FPR) is to significantly reduce the overall complexity of the experiment by selecting circuits that do not significantly contribute to the overall analysis. By smartly picking a set of circuits that is amplificationally complete, the goal is to retain the power of GST while reducing the complexity. Ideally, FPR should produce results similar to those without FPR but in a fraction of the time.

5.7.1 Experimental Setup

In this set of experiments, we maintained consistency with our previous experimental conditions. The gates were generated with a noise PSD described by the function PSD: $S(f) = \frac{10^{-6}}{f}$. Furthermore, like in previous experiments, we used the XYI gate set for these experiments, as well as having each experimental run consist of 1000 shots. Lastly, like before, each of the results was obtained over 100 GST runs.

We conducted the GST protocol in two configurations: one with FPR enabled and the other with FPR disabled. By enabling FPR, we aimed to test its effectiveness in reducing the experimental complexity while retaining the accuracy of GST.

5.7.2 GST Output Comparison

Below, a comparison of the results from the runs with FPR enabled and FPR disabled is presented.

FPR disabled The runs with FPR disabled should be familiar by now; they show over-reporting on the I gate and under-reporting on the X and Y gates, with standard deviations around 0.014.

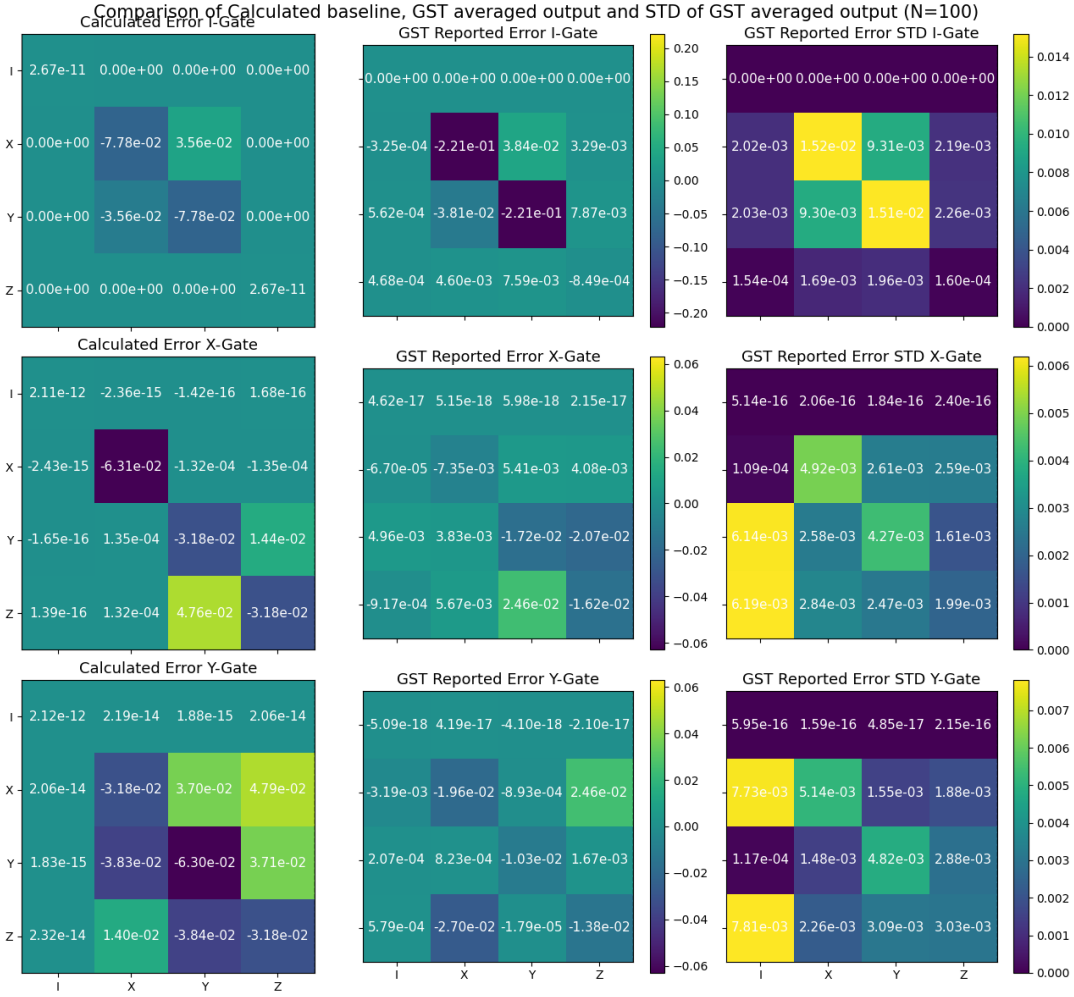


Figure 35: FPR disabled, Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

FPR enabled The runs for FPR enabled interestingly show a more chaotic result for the I gate, with standard deviations much higher, about a factor of 10.

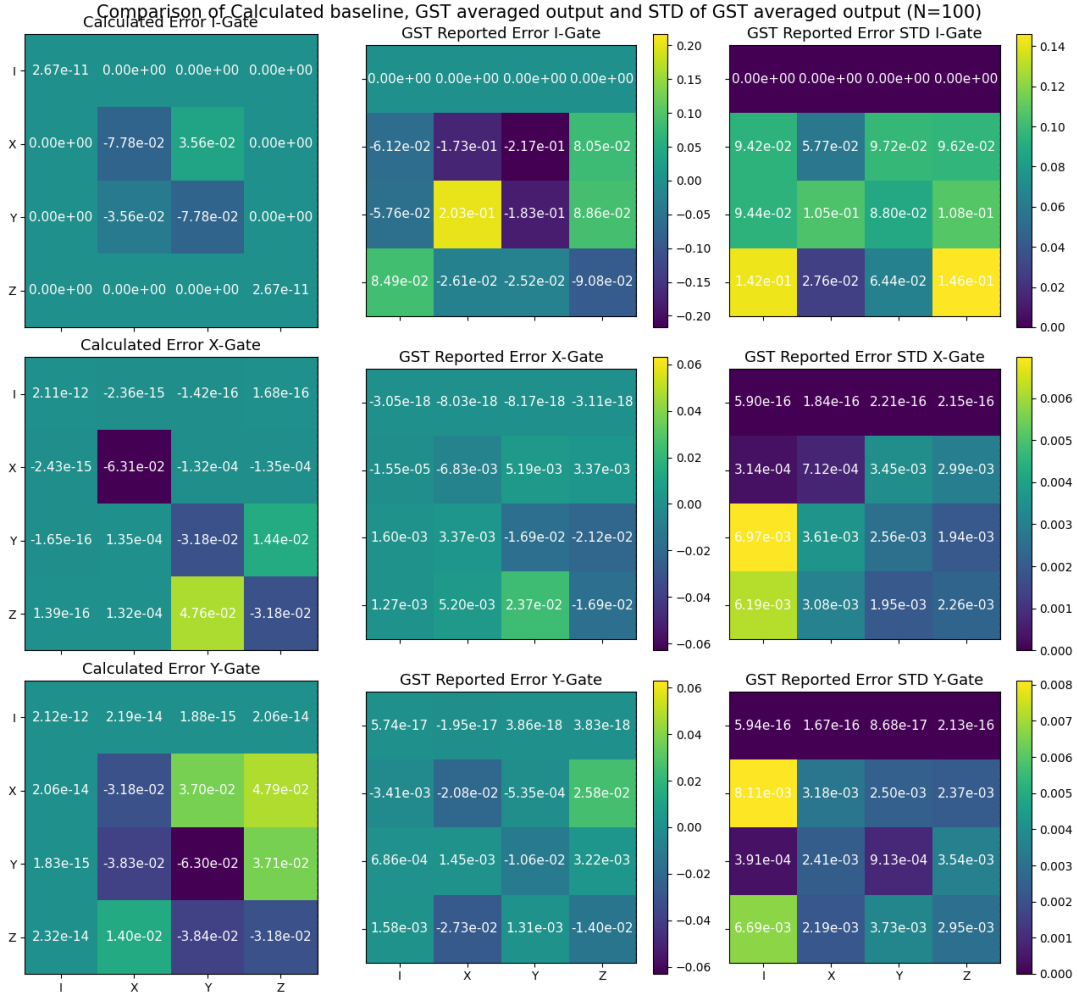


Figure 36: FPR enabled, Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

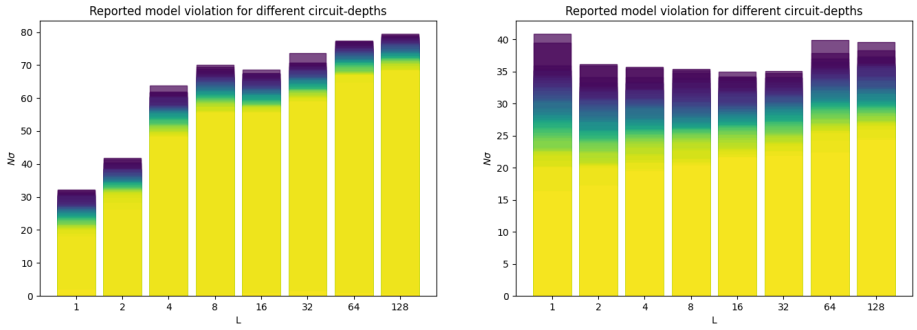
Infidelities

Lastly, the entanglement infidelities reported by GST are listed in the table below. These results show that FPR makes the over-reporting on the I gate slightly worse, while not significantly influencing the results on the X and Y gates.

Gate	Calculated	GST - FPR Disabled	GST - FPR Enabled
I	3.77%	9.97%	11.33%
X	3.16%	1.02%	1.01%
Y	3.14%	1.10%	1.14%

5.7.3 Model Violations

Below, a comparison of the results from the runs with FPR enabled and FPR disabled is presented.



(a) FPR Disabled

(b) FPR Enabled

Figure 37: Model violations of GST with and without Fiducial Pair Reduction

For the run with FPR disabled, the model violations are in the range of 30-80 and show a steady climb for larger depths. In contrast, the model violations for the FPR enabled run are very consistent across all depths, staying within the range of 30-40. It is seen that the FPR run has slightly higher model violations for smaller depths, while for higher depths, the non-FPR run starts to show more model violations.

5.7.4 Discussion of FPR experiment

FPR had the largest effect on the estimation of the I gate, with the X and Y gates being left relatively untouched by it. The overall experiment size for the FPR versus non-FPR runs is about a factor of 5, with the FPR scheme producing much smaller experiments.

This reduction in experiment size comes at a cost: the results of FPR are worse than the results of the non-FPR run. The standard deviations become very high, meaning that with FPR enabled, the results become very unstable. In this case, for the I gate, these standard deviations are so high that the results of GST are largely uninterpretable.

While this seems like a significant drawback for FPR, it should be remembered that FPR is the only reason GST is scalable enough to run on 3 qubits. As mentioned previously, the experiment size tends to grow exponentially with

the number of qubits. Thus, despite its limitations, FPR is essential for making GST feasible for larger quantum systems.

6 Discussion

In this section, we will first recall the main takeaways from each of the experiments conducted. After reviewing these key points, we will combine the lessons learned to provide a comprehensive understanding of the implications for GST performance and accuracy under various conditions. Lastly, the limitations of the study will be discussed.

6.1 Effects of different parameters

We will first go over each of the experiments, following the same order as in the report.

6.1.1 Impact of Noise Strength

The noise strength experiment suggests that the consistency of over- and under-reporting depends on the noise strength. With higher noise strengths, the balance between over-reporting the I gate and under-reporting the X and Y gates changes, suggesting the existence of a sweet spot where GST can balance errors correctly despite Non-Markovianity. This indicates that GST's performance greatly depends on the noise strength in the case of Non-Markovian noise.

6.1.2 Effect of Batch Sizes

Batch size does impact the results, but not in a very clear way. The averages over many runs of GST showed largely the same results regardless of batch size. The primary difference was in the consistency with which GST produced these results. Lower batch sizes are beneficial primarily in increasing the consistency of each run of GST. The reported model violations were clearly correlated with this shift, indicating that while the average results remained stable, the reliability of each individual run improved with smaller batch sizes. Since experiments with smaller batch sizes are usually more difficult to run in real-life experiments, this once again indicates a trade-off between complexity and consistency.

6.1.3 Impact of Maximum Depth

Experiments varying maximum depth reveal surprisingly little effect on estimation accuracy, even when considering it within the light of the aforementioned trade-off between complexity and consistency. While lower depths did provide less consistent results, the relative increase in consistency that higher maximum depths provided was not very significant, making a case for GST runs with lower maximum depths.

6.1.4 Gate Set Configuration

Experiments with different gate sets indicate that removing "troubling" gates does not improve error estimation for other gates. GST, through its method of choosing circuits, appears to quite successfully isolate errors for each gate, minimizing cross-gate influences. Removing the Identity gate from the gate set did not have a significant effect on the other gates.

6.1.5 Fiducial Pair Reduction (FPR)

FPR significantly reduces experiment size but introduces consistency issues similar to those seen in previous experiments. For the one-qubit run, FPR detrimentally affected overall consistency and accuracy. While FPR reduces experimental complexity, it comes at the cost of higher standard deviations and less stable results. While this might suggest FPR is best avoided, it is essential for scaling GST to larger quantum systems, as experiment size grows exponentially with the number of qubits.

6.2 Implications

6.2.1 Accuracy of GST under Non-Markovianity

In the baseline experiment, it can be seen that for the Markovian case, GST is very good at accurately predicting the error. However, the accuracy in all experiments with Non-Markovianity has been quite unsatisfactory. The error in the I gate is consistently over-reported, while GST consistently under-reports the error in the X and Y gates. However, this is not the full story. The noise strength experiment indicates that this balance depends on the noise strength used. It suggests there is a sweet spot for the noise strength where, despite Non-Markovianity, GST properly balances the estimated errors between the I and X/Y gates. This could explain why the results for other experiments are consistently off, as they were all generated with a noise strength of A = -6. According to the Noise Strength Experiment, this level of noise strength falls in the range where the I gate's error is indeed over-reported.

6.2.2 Consistency vs Complexity

Most of the results highlight a trade-off between consistency and complexity. This is evident in experiments involving circuit depths, the number of shots, batch sizes, and the Fiducial Pair Reduction experiment. The more complex options consistently show a lower amount of variation in the reported results. This suggests that, on average, the results GST provides are surprisingly consistent across different settings of maximum depth, number of shots, and batch sizes.

6.2.3 Interpretation of Model Violation

In many of the performed experiments which showed this trade-off between consistency and complexity, there appears to be a strong correlation between the model violations reported and the standard deviations of the returned results. This suggests that model violations should not be interpreted as a measure of the average accuracy of GST results but rather as an indication of their consistency.

Model violations could indicate that the results are imprecise, rather than inaccurate. In extreme cases, it might be more effective to run repeated GST experiments on a smaller scale (with low circuit shots and max depth) and then average these results to obtain a better indication of the underlying data.

It is important to recognize that the presence of non-Markovian noise in GST results is, in fact, a valuable diagnostic feature. Efforts to minimize model violations should not be solely for the purpose of obtaining ostensibly 'better' results. Instead, model violations should be interpreted as inherent properties of the system under investigation. They provide crucial insights into the noise characteristics and their impact on the quantum gates. Consequently, model violations should be considered as an integral part of the diagnostic toolkit rather than a parameter to be minimized indiscriminately.

6.3 Limitations

While this study provides valuable insights into the performance of GST under various conditions, several limitations should be acknowledged to contextualize the findings. These limitations primarily relate to the noise model, gate sets used, and the effectiveness of the comparison between Markovian and Non-Markovian noise effects.

6.3.1 Noise Model

The noise model used in this study only captured noise acting in the Z direction (fluctuations in the energy levels of the qubit):

$$H(t) = H_{ideal}(t) + \sigma_z \cdot S(t)$$

It did not account for other types of noise, such as fluctuations in the driving signals of the qubits, which might have led to significantly different results. Furthermore, the specific noise-generation scheme was limited to noise generated from a power spectral density (PSD), which may not capture all possible effects of noise in real experimental conditions, such as burst errors or noise with higher-order correlations. Lastly, since gate generation was the bottleneck, a decision was made to conduct numerous GST experiments using a few sets of pre-rendered gates. Had this been less of a bottleneck, it would have been interesting to mix Markovian and Non-Markovian noise in different proportions to clearly show when Non-Markovian effects begin to dominate.

6.3.2 Gate Sets

Almost all of the experiments were conducted using the XYI one-qubit gateset. Other types of gates, such as Hadamard or Z gates, might have different effects. This focus gives a somewhat one-sided perspective, where most results show significant over-reporting on the I gate. Additionally, two-qubit gates, while supported by the software, were not extensively studied due to the increased complexity and longer run times. This report includes some results for two-qubit gates but does not delve deeply into them.

6.3.3 Markovian vs Non-Markovian Comparison

While some attempts, such as the baseline experiment, were made to explicitly show the effects of Markovian versus Non-Markovian noise, this was not systematically done for all experiments, since many of these experiments only show the Non-Markovian situation. It would have been interesting to isolate the effects of Non-Markovian noise more clearly and show its contribution to the overall results. However, due to the abundance of data, choices had to be made about which results to present. Some parameters that seemed detrimental in certain experiments (e.g., enabling FPR in the FPR experiment) might actually be beneficial in experimental conditions where the noise is less significant or more of it is Markovian.

7 Conclusion

In this study, the primary goal was to develop a code-base in Python to study the effects of non-Markovianity on Gate Set Tomography (GST). We have explored the performance and accuracy of GST under various conditions, focusing particularly on the impact of noise models, gate sets, and the differences between Markovian and Non-Markovian noise. Our experiments have provided valuable insights into the robustness and limitations of GST in capturing and analyzing quantum gate errors.

7.1 Key Findings

- **Noise Strength:** The noise strength experiment revealed that GST's ability to balance the errors between different gates is highly dependent on the noise strength. We identified a trade-off point, where GST performs optimally, despite the presence of Non-Markovian noise.
- **Batch Sizes:** The impact of batch sizes on GST results was not straightforward. While the average results remained largely the same across different batch sizes, lower batch sizes improved the consistency of GST outputs, highlighting a trade-off between complexity and consistency.
- **Maximum Depth:** Varying the maximum depth of GST circuits showed minimal effects on estimation accuracy. Although higher depths provided

slightly more consistent results, the improvement was not significant, suggesting that lower maximum depths may be sufficient for practical GST applications.

- **Gate Set Configuration:** Removing specific gates, such as the Identity gate, did not significantly impact the error estimation of other gates. GST appears to isolate errors effectively, minimizing cross-gate influences.
- **Fiducial Pair Reduction (FPR):** While FPR significantly reduced experiment size, it introduced consistency issues, particularly for one-qubit runs. This represents the same trade-off between complexity and consistency, but FPR is a rather extreme choice for a reduction in complexity. Therefore, FPR should only be used in situations where the alternative (no-FPR) is infeasible, such as in larger simulations and experiments involving more qubits.

7.2 Implications

Our findings underscore the importance of carefully selecting experimental parameters to balance GST's interpretability with its complexity. Noise strength, batch sizes, and circuit depths all play crucial roles in the accuracy and consistency of GST results.

7.3 Limitations

Several limitations were identified in this study. The noise model used only captured noise in the Z direction, potentially overlooking other significant noise types. Most experiments were conducted using the XYI one-qubit gateset, limiting the generalizability of the findings. Moreover, while attempts were made to compare Markovian and Non-Markovian noise effects, this was not systematically done across all experiments.

7.4 Outlook

Future research should aim to address these limitations by incorporating more comprehensive noise models and exploring a wider variety of gate sets. Systematically isolating and analyzing the effects of Non-Markovian noise across different experimental conditions would provide deeper insights. Additionally, extending this study to include multi-qubit systems using FPR and other techniques to manage complexity would provide an even more interesting perspective on GST. A preliminary investigation into this topic was conducted and is outlined below.

7.4.1 Outlook: Two-qubit case

By generating 2-qubit gates in the following fashion:

$$G_{2\_\text{qubit}} = G_{\text{ideal}} \otimes G_{\text{noisy}}$$

we can formulate operators for a 2-qubit system, in which the noise is only acting on one of the qubits. A central question in this case is whether GST can differentiate these two qubits and consistently place the error on the noisy qubit. Since GST might not be able to differentiate where the noise is coming from, a possibility would be that the result of the ideal qubit is somehow affected by the noise in the other qubit.

The software supports this kind of analysis, and preliminary investigation shows that GST is quite effective in this scenario, however, looking at the raw error generators like before, does not provide a very intuitive understanding:

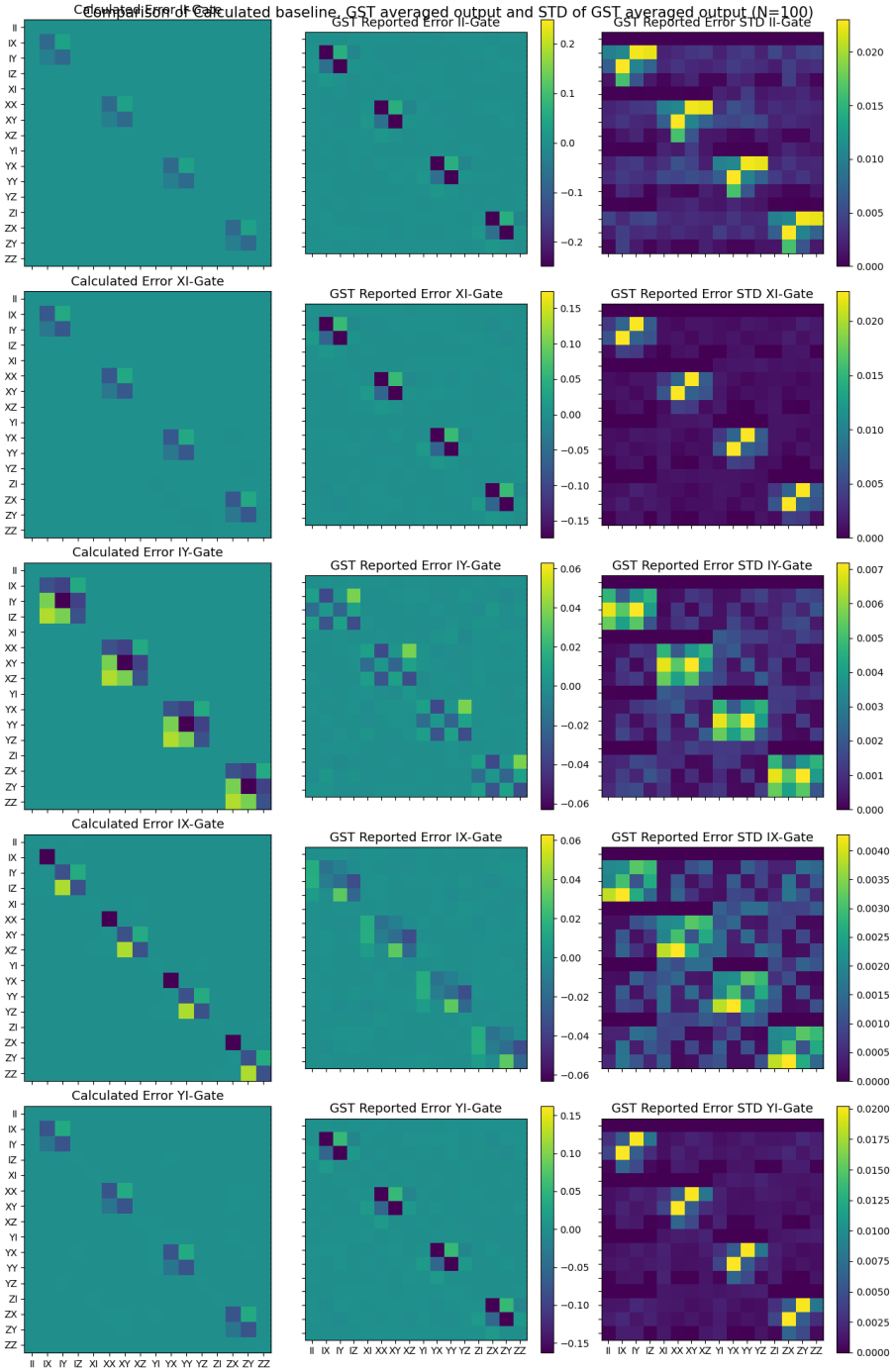


Figure 38: 2-Qubit XYI gateset with one noisy qubit, and one ideal qubit, Comparison of calculated Error generators (Left), GST reported Error generators (Middle), as well as the standard deviation for each of the GST reported Error generators (Right) for each of the gates in the Gate set

Additionally, we can look at the reported infidelities below:

Table 4: Comparison of Entanglement Infidelities for Different Gates

Gate	Entanglement Infidelity
II	10.60%
IX	1.35%
XI	7.81%
IY	2.50%
YI	7.50%

This preliminary data shows that in the case of the second qubit being noisy, there is a distinct impact on the infidelities. The table of infidelities seems to show that whenever the noisy qubit is running idle, as is the case for gates II, XI, and YI, the infidelity of the gate is heavily affected. This observation seems to highlight this recurring issue, where the identity gate tends to be blamed for most of the errors in the system. This behavior is consistent with our earlier findings where the identity gate often absorbs the error attribution in the presence of noise. The fact that this happens gives an indication that the 2-qubit GST experiment is able to correctly identify the noisy qubit, however, no conclusive evidence is presented here.

The exact question of whether the noise is transferred in GST from one qubit to another thus remains unresolved. This leads to an important area for further research: conducting a more detailed analysis to isolate and verify if GST can distinguish between the sources of noise for each qubit independently. Future experiments could include varying the type and level of noise applied to each qubit and examining the resultant GST outputs to determine the precise impact on both qubits.

7.5 Final Remarks

In conclusion, this study has highlighted both the strengths and weaknesses of GST under various conditions, providing a foundation for further research and practical applications. By understanding and addressing the identified limitations, future studies can enhance the reliability and scalability of GST, contributing to more robust quantum computing technologies. By developing a codebase in Python to study the effects of non-Markovianity on GST, this research has also laid the groundwork for future explorations and improvements in quantum gate error diagnostics.

References

[1] Daniele Binosi et al. *Threshold Quantum State Tomography*. 2024. arXiv: 2401.12864 [quant-ph].

[2] Sergio Boixo et al. “Characterizing quantum supremacy in near-term devices”. In: *Nature Physics* 14.6 (Apr. 2018), pp. 595–600. DOI: 10.1038/s41567-018-0124-x.

[3] D. Leibfried et al. “Experimental Determination of the Motional Quantum State of a Trapped Atom”. In: *Physical Review Letters* 77.21 (1996), pp. 4281–4285. DOI: 10.1103/physrevlett.77.4281.

[4] Erik Nielsen et al. “Gate Set Tomography”. In: *Quantum* 5 (Oct. 2021), p. 557. ISSN: 2521-327X. DOI: 10.22331/q-2021-10-05-557. URL: http://dx.doi.org/10.22331/q-2021-10-05-557.

[5] Erik Nielsen et al. “Probing quantum processor performance with pyGSTi”. In: *Quantum Science and Technology* 5.4 (July 2020), p. 044002. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ab8aa4. URL: http://dx.doi.org/10.1088/2058-9565/ab8aa4.

[6] Jeremy L O’Brien et al. “Quantum process tomography of a controlled-NOT gate”. In: *Physical review letters* 93.8 (2004), p. 080502.

[7] Corey Ostrove et al. “Near-Minimal Gate Set Tomography Experiment Designs”. In: *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 1. IEEE. 2023, pp. 1422–1432.

[8] E. Paladino et al. “Decoherence and $1/f$ Noise in Josephson Qubits”. In: *Phys. Rev. Lett.* 88 (22 May 2002), p. 228304. DOI: 10.1103/PhysRevLett.88.228304. URL: https://link.aps.org/doi/10.1103/PhysRevLett.88.228304.

[9] John Preskill. “Quantum Computing in the NISQ era and beyond”. In: *Quantum* 2 (Aug. 2018), p. 79. ISSN: 2521-327X. DOI: 10.22331/q-2018-08-06-79. URL: http://dx.doi.org/10.22331/q-2018-08-06-79.

[10] Kenneth M. Rudinger et al. *Two-Qubit Gate Set Tomography with Fewer Circuits*. 2023. arXiv: 2307.15767 [quant-ph].

[11] Dan J Shepherd. “On the role of Hadamard gates in quantum circuits”. In: *Quantum Information Processing* 5 (2006), pp. 161–177.

[12] J. Timmer and M. König. “On generating power law noise.” In: 300 (Aug. 1995), p. 707.

[13] HeeBong Yang and Na Young Kim. “Material-Inherent Noise Sources in Quantum Information Architecture”. In: *Materials* 16.7 (2023). ISSN: 1996-1944. DOI: 10.3390/ma16072561. URL: https://www.mdpi.com/1996-1944/16/7/2561.

[14] Richard N Youngworth, Benjamin B Gallagher, and Brian L Stamper. “An overview of power spectral density (PSD) calculations”. In: *Optical manufacturing and testing VI* 5869 (2005), pp. 206–216.

[15] Mário Ziman, Martin Plesch, and Vladimír Buž Zek. “Reconstruction of superoperators from incomplete measurements”. In: *Foundations of Physics* 36 (2006), pp. 127–156.