Extending Quantum State Tomography for Superconducting Quantum Processors

Francisca Vasconcelos, Morten Kjaergaard, Tim Menke, Simon Gustavsson, Terry P. Orlando, and William D. Oliver

Massachusetts Institute of Technology, Engineering Quantum Systems (EQuS)

Quantum State Tomography (QST), or the reconstruction of the density matrix of a quantum state via measurements, is critical to ensure the proper functionality of qubits and quantum operations in a quantum computer. In this work, we extend existing QST code based on Maximum Likelihood Estimation from two to an arbitrary number of qubits and from one to arbitrarily many energy levels. A 100x algorithmic speedup is achieved over the original implementation. However, the exponential scaling of the density matrix makes this MLE-based algorithm infeasible for analysis over 6-qubits on a standard computer. To mitigate this limitation, we propose a novel deep-learning based approach to QST. Utilizing the CycleGAN architecture from the field of computer vision, we aim to address the issues of scalability and bias that plague current QST implementations.

Introduction & Related Work

Quantum computing has the potential to revolutionize computer science. Scientists are working to achieve quantum supremacy (the potential ability to solve problems classical computers cannot) and quantum advantage (solving problems faster), through the development of large-scale quantum computing devices. This, for example, could result in an exponential algorithmic runtime speedup of problems such as optimization and prime factorization, that cannot be solved with the most advanced classical supercomputers for even moderately sized problem. At the root of the quantum computer's computational power is the qubit. In addition to entanglement, qubits store information as a quantum superposition of the two classical bit states, $\alpha|0\rangle + \beta|1\rangle$ (where $\alpha, \beta \in \mathbb{C}$). A qubit measurement only reveals the classical bit-value of 0 or 1, corresponding to qubit state $|0\rangle$ or $|1\rangle$, respectively. However, the state to which a qubit collapses is not random. By performing a statistical analysis of several measurements, it is possible to infer the original superposition state and reconstruct the density matrix. This process is known as Quantum State Tomography (QST) and is critical to ensure the proper functionality of qubits and quantum operations in a quantum computer.

QST was introduced long before the first quantum computers came into existence, with related work dating as far back as the 1980s **[9]**. While QST has been used to assess actual quantum computers since the early 2000s **[10, 14]**, only recently it has become a highly active area of research. This is in part due to recent advances in machine learning and artificial intelligence as well as the realization of larger, more capable, quantum systems. Historically, researchers were mostly concerned with the actual fabrication of qubits, using QST simply as a metric for qubit-quality factors, such as coherence time and fidelity. As scalable quantum systems became more feasible, research groups felt the need for accurate metrics of algorithm and qubit performance on their processors. This is evident from the increasing complexity of recent work, and the superiority of the newer over traditional methods. Even with recent progress, QST has become a limiting factor in the ability to assess quantum computers. In 2019, the authors of a 12-qubit experiment claimed that full state tomography was impractical and instead proved a bound on state fidelity **[7]**.

That being said, there currently exist a variety of quantum computers, varying in type and size. These computers can range from single-qubit laboratory systems to the 50-qubit IBM machine and use physics ranging from superconducting circuits to trapped ions, photonics, or majorana particles **[17]**. One of the main challenges in the field is to make QST algorithms broad and scalable enough to handle all this diversity of systems, as well as those still to come in the future. The reported work addresses this difficulty by extending the capacity of the MIT Engineering Quantum Systems (EQuS) group 1- and 2-qubit QST method[1] to n-qubit systems. While the mathematical and physical intuitions used to develop the algorithm are broadly applicable, the analysis will be tailored specifically to the superconducting artificial atoms fabricated and tested by the EQuS group. This will enable targeting and correction of the specific properties of this particular system.

The main challenges in QST lie in the inherent noise and imperfections of quantum systems, as well as the exponential scaling of the density matrix in the number of qubits. Error is frequently introduced by energy leakage during measurements, as well as decay of the qubits themselves. Several constraints on the density matrix, such as normalization and positive semi-definiteness, must be met to ensure that the final prediction represents an actual physical state. However, since the problem is one of statistical inference, it is impossible to determine if the outcome of QST is correct, even when the desired state is known. While this difficulty can be addressed with recourse to machine learning, it cre-

[1] Implemented by EQuS postdoc Morten Kjaergaard based on the PhD thesis of Jerry Chow, Yale University.

ates a challenging learning problem, since the labels used to train models might not be fully accurate. Furthermore, the process of collecting data is resource and time intensive, making it difficult to leverage big-data techniques.

Common mathematical techniques used in single-qubit QST (which extend to multi-qubit QST) include Direct Inversion Tomography, Minimum P-Distance of Bloch vectors, Maximum Fidelity, Fisher Information Distance, Maximum Likelihhod Estimation (MLE), and Bayesian Mean Estimation **[12]**. While MLE is one of the most commonly used techniques, due to its simplicity and good performance, it has received strong criticism, due to the existence of slightly obscure but critical states for which it fails **[5, 13]**. Since this technique is currently used in the EQuS QST implementation, an analysis of its reliability will be performed. Furthermore, the recent trend in QST seems to be moving away from highly mathematical and physics-based modeling to the space of machine learning, namely deep learning. A number of groups have been using neural networks to learn the mappings from quantum processor output data to quantum state representation **[1, 15, 18]**.The current EQuS algorithm does not make use of these state-of-the-art machine learning techniques, which will be an area of potential exploration.

In order to implement n-qubit state tomography, we generalize the preexisting 1- and 2-qubit process. This QST process consists of training a Support Vector Machine (SVM) to count probabilities, curve-fitting Rabi oscillations to learn 'beta' parameters, solving for the measurement expectations, and using MLE to construct the density matrix. The expanded QST code is tested on actual experimental data from a 3-level, 2-qubit waveguide QED system. The n-qubit expansion of the QST algorithm poses an interesting algorithmic challenge, since the size of the density matrix scales exponentially with the number of qubits (as a result of the properties of the qubit Hilbert space). We achieve a 100x speedup on the original QST implementation, but the exponential scaling dominates. 5-qubit tomography takes 15 minutes, while 6-qubit tomography takes 13 hours (on a standard desktop) and any larger system is deemed computationally infeasible. To address this, we propose a novel approach to QST, based on the CycleGAN deep learning model from the field of computer vision.

QstGAN is a new architecture, aimed at addressing two of the main difficulties faced by QST. Since the CycleGAN network was developed to process image data, which is large and arguably the most complex domain currently considered in machine learning, we theorize that it will work well for the large but simpler measurements of quantum tomography. Although there has been recent work in the field utilizing deep learning architectures, these are primarily based on two types of approaches. One possibility is to rely on fully-supervised architectures **[6, 11, 19]**. However, these networks suffer from the limitation that they can only be trained with simulated data. Hence, they tend to underperform when tested on real data.

The second possibility, which we pursue in this work, is to use real data for training and rely on unsupervised machine learning techniques. Most works in this area have used restricted Boltzmann machines **[2, 16]**. While lending

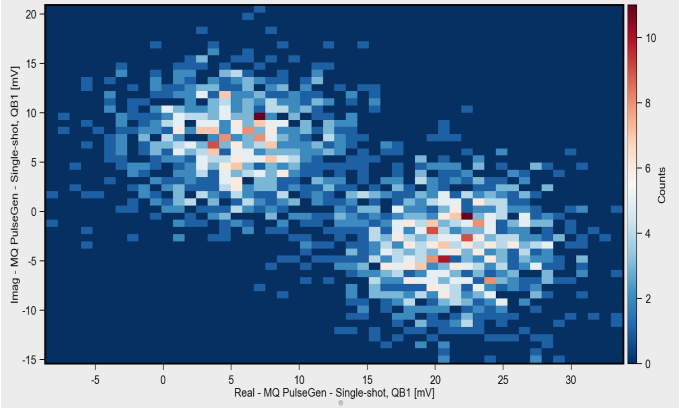

Figure 1: *Example single-qubit measurement output mapped onto the I-Q plane.*

themselves to a compelling physical interpretation, these networks are known to underperform a more recent class of models known as generative adversarial learning (GAN) networks **[8]**. A particular class of these models, known as the CycleGAN **[20]** has recently become popular in computer vision, where it is considered state-of-the-art for problems such as image synthesis. We will compare the performance of QstGAN with other deep-learning and statistical QST approaches.[2]

Methods

In this work, we aim to address two key challenges in QST: 1) inability to generate true density matrix labels for measurement data and 2) exponential growth of parameters with the number of qubits. We start by providing a brief overview of the pre-existing 1 and 2-qubit QST implemented by EQuS postdoc, Morten Kjaergaard, based on the methodology developed in the MS and PhD theses of Julia Cramer **[4]** and Jerry Chow **[3]**. We also briefly discuss work to improve and extend this implementation to n-qubits, emphasizing the limitations of this commonly used technique. Finally, we propose a novel QST method, which utilizes the CycleGAN network.

Traditional MLE Based QST

To perform QST, a series of single shot data (in the $|0\rangle$ and $|1\rangle$ state), with respective labels, is first loaded into the code suite. The measurement process consists of transmitting a microwave with a specific frequency to the qubit and comparing it to the altered microwave returned by it. Using the phase-shift between the two signals, the measurement can be mapped onto the complex (I-Q) plane, as shown in [Fig 1]. An SVM is then trained to discern between the ground and excited states, as shown in [Fig 2]. Next, a 0 to π Rabi oscillation is performed on the quantum computer and the Rabi voltages are converted to Rabi probabilities, using the SVM. These probabilities are then fitted with cosines, and

[2]Note to reader: The QstGAN portion of the work was not completed in the SuperUROP timeframe and there are no results to report. However, the proposed high-level idea will be discussed in the *Future Work* section.

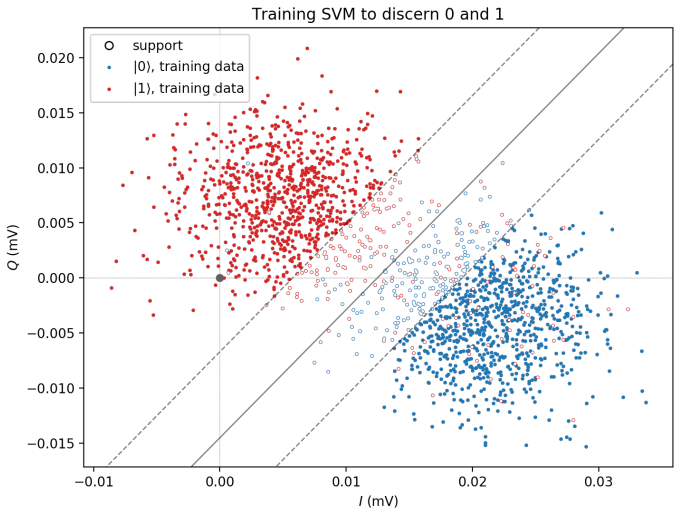

Figure 2: *An SVM trained to discern ground states from excited states for a single-shot qubit experiment.*

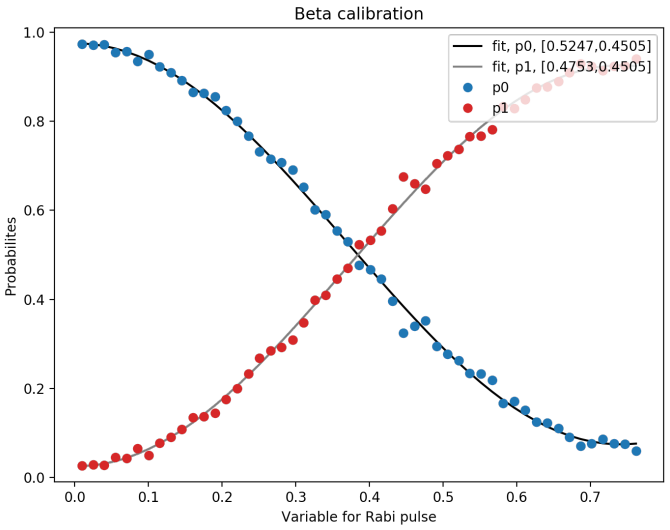

Figure 3: *Beta parameters found by fitting cosines to Rabi oscillation probability curves, for a single-qubit experiment.*

the parameters of the fit are used to find the 'beta' parameters of the quantum system, as shown in [Fig 3]. These 'beta' parameters indicate if the system has a tendency to measure more or less frequently in the excited or ground state. For example, when there is energy leakage, qubits that should be in the excited state might occasionally decay to the ground state before measurement.

Given β parameters, QST aims to reconstruct the density matrix. To determine the density matrix of a given state, the qubit is excited to that state within the quantum processor, and a series of measurements (anywhere from hundreds to thousands) is performed, separately along the x, y, and z axes (different microwave pulses must be applied for each axis). Using the SVM, the measured values are binned into the $|0\rangle$ or $|1\rangle$ state, and the number of values in each bin is denoted as p_0 and p_1 respectively. Using the previously calculated β parameters, the system of equations

$$\begin{bmatrix} p_0 \\ p_1 \end{bmatrix} = \begin{bmatrix} \beta_I^{|0\rangle} & \beta_{\sigma_A}^{|0\rangle} \\ \beta_I^{|1\rangle} & \beta_{\sigma_A}^{|1\rangle} \end{bmatrix} \begin{bmatrix} \langle I \rangle_M \\ \langle \sigma_A \rangle_M \end{bmatrix}, A \in \{x, y, z\}. \quad (1)$$

is solved for the measurement expectations $\langle I \rangle_M$ and $\langle \sigma_x \rangle_M$, $\langle \sigma_y \rangle_M$, and $\langle \sigma_z \rangle_M$.

These measurement expectations are then used to estimate the true expectations, using MLE. The true expectations are defined as

$$\langle \sigma_A \rangle = Tr(\rho_{1QB} \sigma_A), \quad (2)$$

where

$$\rho_{1QB} = \frac{1}{2}(\langle I \rangle I + \langle \sigma_x \rangle \sigma_x + \langle \sigma_y \rangle \sigma_y + \langle \sigma_z \rangle \sigma_z) \quad (3)$$

is the density matrix, I the identity, and σ_x, σ_y, and σ_z correspond to the three Pauli matrices. MLE minimizes the cost function

$$L = \sum_{P \in \{\sigma_x, \sigma_y, \sigma_z\}} (\langle P \rangle_M - Tr(P \rho_t))^2 \quad (4)$$

to find the density matrix ρ_t whose expectations of (2) best match the measurement expectations $\langle \sigma_x \rangle_M$, $\langle \sigma_y \rangle_M$, and $\langle \sigma_z \rangle_M$.

To ensure that the density matrix represents a true physical system, it must satisfy two constraints: it must be positive semi-definite and normalizable. The MLE problem of (4) enforces positive semi-definiteness by using the Cholesky decomposition of the density matrix

$$\rho_t = \frac{T^\dagger T}{Tr(T^\dagger T)}, \quad T = \begin{bmatrix} t_0 & 0 \\ t_2 + i t_3 & t_1 \end{bmatrix}. \quad (5)$$

Normalization is enforced by adding the constraint

$$t_0^2 + t_1^2 + t_2^2 + t_3^2 = 1 \quad (6)$$

to the optimization. When the MLE optimization is complete, the density matrix is determined by using the t_i values to construct ρ_t. Sample visualizations produced by the 1- and 2-qubit tomography code on experimental measurement data are shown in [Fig 4] and [Fig 5].

MLE QST Extension and Limitations

The extension of this algorithm to larger qubit systems is one of the primary focuses of this work. To enable this, we performed work along two directions: theoretical and algorithmic.

Theory

To better understand the theory behind the QST, we derived an alternative form of the cost function of equation (4). This was done with recourse to mathematical manipulation of the Cholesky decomposition of the density matrix and the mathematical properties of the trace operation and eigenvalue decomposition of matrices. The derivation is as follows.

We begin by replacing the density matrix with its Cholesky decomposition,

$$Tr(P\rho_t) = Tr(P\frac{T^\dagger T}{Tr(T^\dagger T)}) = \frac{Tr(PT^\dagger T)}{Tr(T^\dagger T)} = \frac{Tr(TPT^\dagger)}{Tr(TT^\dagger)}.$$

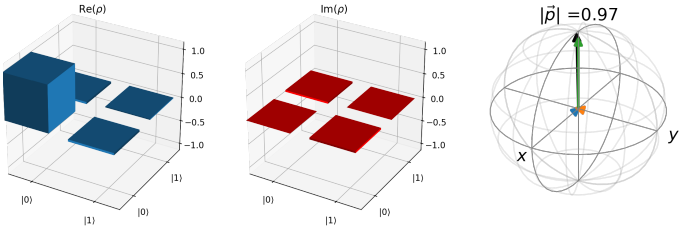

Figure 4: *Sample visualization of 1-qubit density matrix and corresponding Bloch-sphere representation, from previous QST implementation.*

We then replace the Pauli matrices with their respective eigenvalue decompositions $P = X\Lambda_P X^{-1}$. Since the Pauli matrices are unitary, $X^{-1} = X^\dagger$, and

$$\frac{Tr(TPT^\dagger)}{Tr(TT^\dagger)} = \frac{Tr(TX\Lambda_P X^{-1}T^\dagger)}{Tr(TXX^\dagger T^\dagger)} = \frac{Tr((TX)\Lambda_P(TX)^\dagger)}{Tr(TX(TX)^\dagger)}.$$

Defining $Q = TX$, it follows that

$$Tr(P\rho_t) = \frac{Tr(Q\Lambda_P Q^\dagger)}{Tr(QQ^\dagger)}. \tag{7}$$

Using this in the MLE expression of (4) leads to

$$\begin{aligned}
\langle P\rangle_M - Tr(P\rho_t) &= \langle P\rangle_M - \frac{Tr(Q\Lambda_P Q^\dagger)}{Tr(QQ^\dagger)} \\
&= \frac{\langle P\rangle_M Tr(QQ^\dagger) - Tr(Q\Lambda_P Q^\dagger)}{Tr(QQ^\dagger)} \\
&= \frac{Tr(Q\langle P\rangle_M Q^\dagger) - Tr(Q\Lambda_P Q^\dagger)}{Tr(QQ^\dagger)} \\
&= \frac{Tr(Q\langle P\rangle_M Q^\dagger - Q\Lambda_P Q^\dagger)}{Tr(QQ^\dagger)} \\
&= \frac{Tr(Q(\langle P\rangle_M I - \Lambda_P)Q^\dagger)}{Tr(QQ^\dagger)}.
\end{aligned}$$

Defining $\Lambda'_P = \langle P\rangle_M I - \Lambda_P$, we obtain the matrix P' of eigenvalue decomposition $P' = X\Lambda'_P X^\dagger$, and can rewite the expression above as

$$\begin{aligned}
\langle P\rangle_M - Tr(P\rho_t) &= \frac{Tr(T(X\Lambda'_P X^\dagger)T^\dagger)}{Tr(TXX^\dagger T^\dagger)} \\
&= \frac{Tr(TP'T^\dagger)}{Tr(TT^\dagger)} \\
&= Tr(P'\frac{TT^\dagger}{Tr(TT^\dagger)}) \\
&= Tr(P'\rho_t)
\end{aligned}$$

leading to a simplified expression for the cost function

$$L_{new} = \sum_{P'} (Tr(P'\rho_t))^2, \tag{8}$$

where the P' matrices are the same as the P matrices, with eigenvalues modified to

$$\lambda_{P'} = \langle P\rangle_M - \lambda_P. \tag{9}$$

Implementation of this cost function achieved the same speeds as the original form, but allowed us to further analyze the MLE optimization. For example, it becomes clear that in order to minimize the cost, the columns of ρ_t should lie in the nullspace of all the P' matrices. Furthermore, the process of implementing this new cost function resulted in the discovery of a flaw in our original Python implementation: an unnecessary, costly for-loop call in our cost function generator method. Fixing this mistake resulted in a 100x speed-up of our algorithm.

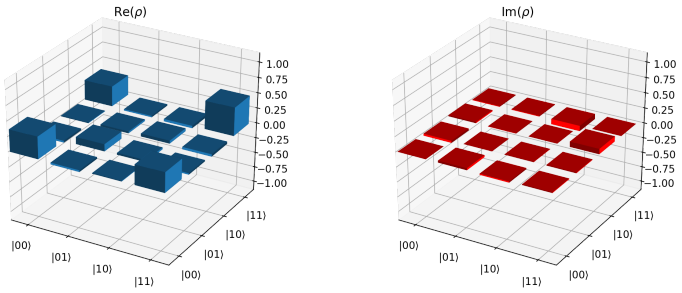

Figure 5: *Sample visualization of 2-qubit density matrix, from previous QST implementation.*

Algorithms

To enable the extension of the MLE algorithm to larger qubit systems, the functions used in the code were generalized to account for multi-qubit entanglement. This posed a computational complexity challenge, since the size of the density matrix is $4^{(\#\text{ qubits})}$. While the basic structure was established with the development of the 2-qubit code, some tricks were used in the n-qubit extension to minimize redundancies. For example, when performing 3-qubit QST, if all three qubits are measured along the x axis (an $'XXX'$ pulse), the values that would be found by applying an $'III'$, $'IIX'$, $'IXI'$, $'IXX'$, $'XII'$, $'XIX'$, or $'XXI'$ pulse fall out of the system of equations relating binned values to expectation measurements via 'beta' parameters, by extending (1). Furthermore, from a software engineering perspective, the QST was previously implemented as two separate scripts, one for 1-qubit and one for 2-qubit tomography. The overall code structure has been overhauled to allow the user to simply select the number of qubits for the tomography, by typing a value into an all-purpose tomography function/class.

Although n-qubit tomography was theoretically achieved in this work, in reality this approach is constrained by computational runtime, as is discussed in the *Results Section*. Furthermore, as discussed in the *Introduction* section, a number of studies have found that MLE has implicit biases in how it maps unphysical states to physical ones. Finally, the entire process of training the SVM and calculating the 'beta' matrix to prepare the system for MLE, relies on assumptions about how the system works, which are likely to result in error propagation. We propose a new approach to QST that aims to address these issues in the *Future Work* section.

Results & Evaluation

Our experiments show that the more efficient implementation of the cost function, using (8), reduces the overall runtime of QST by two orders of magnitude. This is shown in [Fig 6]. This complexity reduction enables 5-qubit tomography in ∼10 minutes, instead of ∼17 hours [Fig 7] of the original approach. We tested our implementation with simulated noisy GHZ (maximally-entangled) states

$$|\text{GHZ}\rangle_N = \frac{|0\rangle^{\otimes N} + |1\rangle^{\otimes N}}{\sqrt{2}}, \tag{10}$$

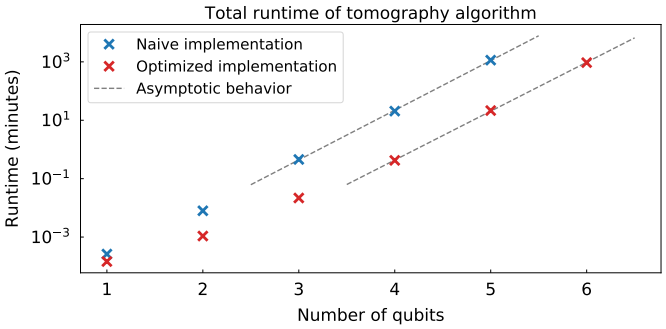

Figure 6: *Comparison of runtime for naive and optimized MLE cost function implementations for multi-qubit systems. Timing measurements were performed on Late 2014 iMac (4GHz Intel Core i7). Note that the optimized cost function allows for the tomography of 6-qubits, which was not feasible with the naive implementation.*

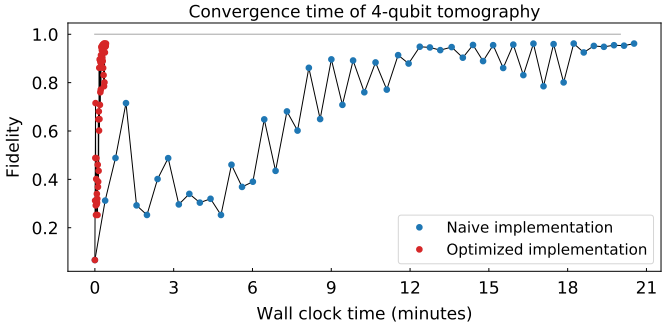

Figure 7: *Convergence of MLE for naive and optimized cost function implementations of a 4-qubit system. Timing measurements were performed on Late 2014 iMac (4GHz Intel Core i7). Fidelity is calculated using the* $Tr(\rho_{estimated}\rho_{ideal})$ *formalism.*

where N is the number of qubits, using the convergence metric

$$\text{Fidelity} = \sqrt{\rho_{\text{ideal}}\rho_{\text{est}}}, \tag{11}$$

where ρ_{ideal} is the density matrix used to generate simulated expectation values and ρ_{est} is the predicted density matrix during each iteration of the MLE convergence process. It should be noted that higher fidelities can be achieved by reducing the MLE tolerance, at the cost of increased runtime.

While, in theory, this approach can work for any number of qubits, the exponentially scaling algorithmic complexity makes it practically infeasible for more than 6-qubits. Nevertheless, for the near term benchmarking goals of EQuS, 3- and 5-qubit packages, the approach is sufficient. Although EQuS has not yet begun measuring 3-qubit or larger systems, we experimentally demonstrate the generalization of the QST implementation on data from a state-of-the-art waveguide QED experiment.[3] Here, the author uses our implementation to generate a density matrix from measurements of a 3-level, 2-photon system [Fig 8]. This would not have been possible with the previous 1- or 2-qubit tomography implementations, which only worked for 2-level systems. The author demonstrated an 85% fidelity using the tomography suite (in this fidelity calculation, ρ_{ideal} is the target state of the experiment and ρ_{est} is the QST-predicted state).

[3]Bharath Kannan, *Generating Non-Classical and Spatially-Correlated Photons in a Waveguide QED Architecture* (unpublished)

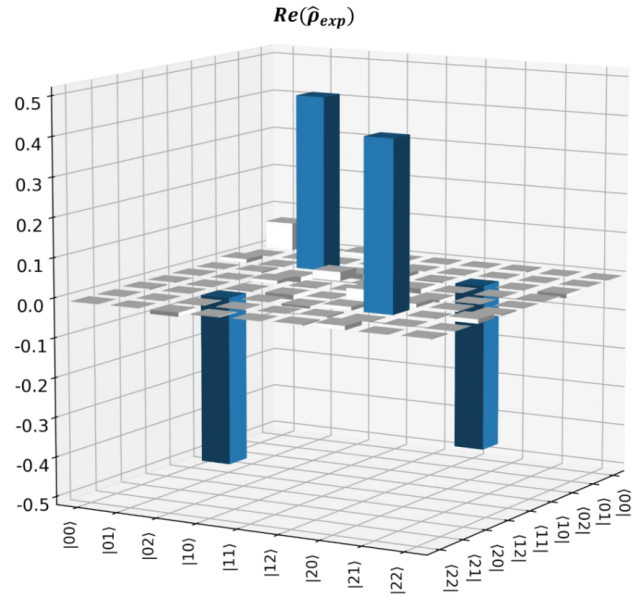

Figure 8: *MLE QST implementation output from 3-level 2-qubit waveguide QED experimental data. Only the real portion of the matrix is depicted. The state was calculated to have 85% fidelity.*

Future Work

Although our MLE implementation will suffice for the near-term small-scale tomography purposes of the EQuS group, corporate superconducting processors with anywhere from 19- (Rigetti Acorn) to 72-qubits (Google Bristlecone) already exist. Thus, it is necessary to develop a more scalable form of QST, in order to assess these large systems. We introduce a novel QST framework, which draws inspiration from the field of computer vision.

QstGAN: CycleGANs for QST

CycleGANs (Cycle-Consistent Generative Adversarial Networks) are frequently used for style transfer and image generation in computer vision. We propose to adapt this network architecture to learn the mapping from measurement data to density matrix and vice versa. Since CycleGANs networks were designed to find patterns in very high-dimensional image data, they are well suited to address the exponential growth issue faced by QST. Furthermore, because they rely on unsupervised learning, they are well suited to deal with the small amounts of training data available and the lack of groundtruth density matrix label for any given measurement training data.

The proposed high-level architecture is shown in [Fig 9]. The network will consist of two generator networks and a discriminator network. One generator will learn the mapping from density matrix (plus random noise) to I-Q measurement values, while the second generator will learn the mapping from I-Q values to density matrix. The generator that takes the density matrix as input will map a high-dimensional gaussian vector input to the IQ-measurement values, conditioned on the given density matrix. When training the network, the discriminator will try to discriminate between these synthetic measurements and real measurement data. The process of training the network will consist

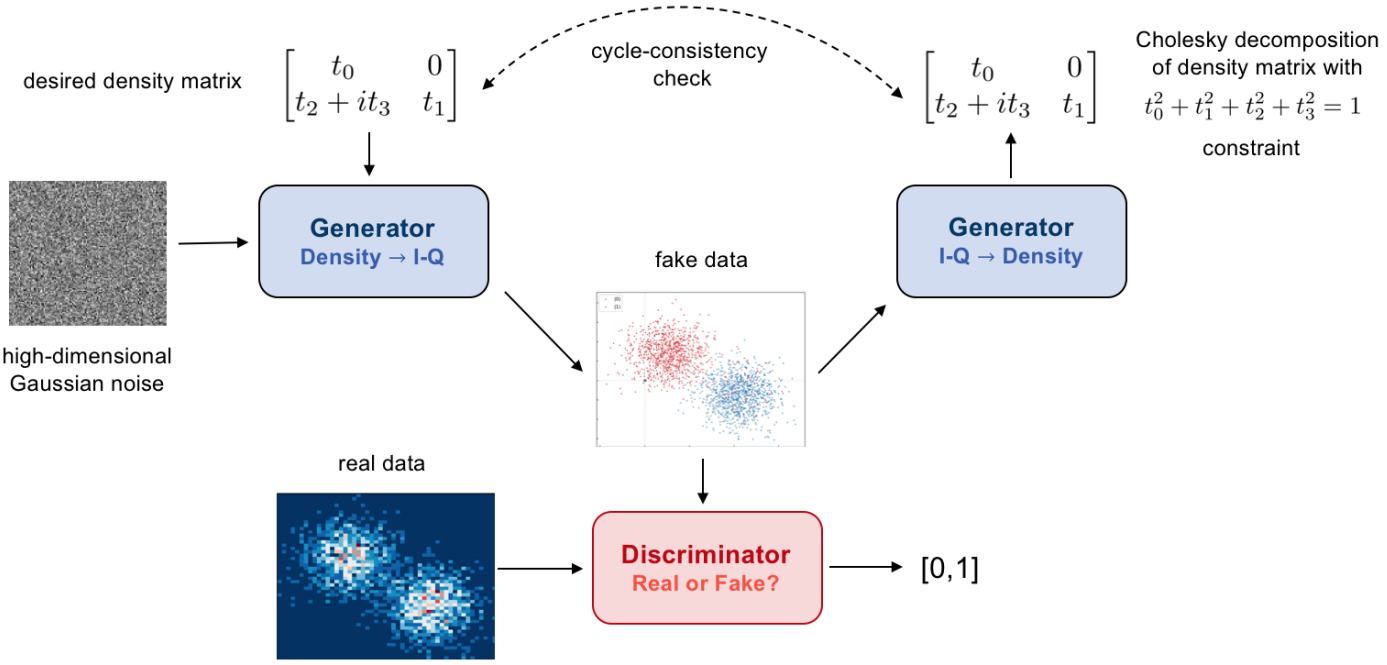

Figure 9: *Proposed QST CycleGANs network architecture.*

of several iterations of measurement data generation and discrimination, with a cycle-consistency check, to ensure that the network is learning the proper transformation (by checking that the inverse transform of the generated data matches the original input).

One of the key challenges is ensuring that the network is truly learning a mapping from measurement to density matrix and back. Thus, to help guide the network, we propose to pre-train the individual networks (prior to the overall CycleGANs training) with simulated measurement values, corresponding to ideal density matrices. This can be done by calculating the probability counts from the density matrix and creating a corresponding set of point clusters with a distinct separation, along each of the measurement axes. We will try training the network both with and without this training step and see if it affects the outcome or performance.

QstGAN Proposed Implementation

Although the ultimate architecture is a CycleGAN (to learn the mapping from measurement data to I-Q values and vice versa), it is necessary to create a framework to optimize the network parameters and serve as a performance benchmark. Thus, we will begin by implementing a fully-supervised network, followed by a GAN (just the mapping from measurement data to density matrix). Once this works reasonably well we train the CycleGAN. We will begin by focusing on a 1-qubit system and characterize the performance of our network. Once this produces good results, we will extend our implementation to higher qubit systems.

To train the initial fully-supervised network, mapping I-Q measurement values to density matrices, we will use simulated data. We do not expect this to perform very well on real data. We will add the GAN generator that maps I-Q values to density matrices. We expect this to perform worse than the ultimate CycleGAN network, which will have a cycle-consistency check (meaning it will learn both the function mapping and its inverse). We will try training the network to generate three different forms of density matrices (to determine which works best). The first will be the entire density matrix, the second will be the Cholesky decomposition of the density matrix, and the final will be the expectation values along each of the measurement axes. All represent the same information, but one structure could be easier for the network to learn than the others.

Finally, we will train the full CycleGAN network with a structure similar to that of the GAN. Our main experiment for this architecture will be a comparison to pre-training with simulated data, to see if we can boost network performance.

We will compare all of these networks by testing performance on a common set of simulated data, for which we have the groundtruth density matrix. We will also generate density matrices for experimental data using MLE and compare the output to those of the three networks. We will need to find a metric for experimental density matrix comparisions and are currently considering Von Neumann entanglement entropy.

To conclude our analysis, we will do a comparison with other major tomography techniques and papers. Specifically, we plan to compare our CycleGAN architecture with the generally accepted MLE approach, restricted Boltzmann machine architectures (which have become the standard deep learning approach in the field), and recent work using tensor networks and Recursive Neural Networks (RNNs) to characterize entanglement of large qubit systems. We will use three main criteria in our assessment: algorithm input (in our case this is the raw I-Q data, but all others use the unphysical expectation measurements), complexity (for all full-QST techniques this is $O(4^n)$), and QST performance (we hope that our network will achieve provably better results).

Conclusion

While QST is critical for analyzing the performance of quantum processors, it proves to be a difficult problem, with challenges ranging from measurement noise to exponential runtimes. In this work, we have improved the qubit assessment capabilities of the EQuS superconducting quantum processor by creating a more general implementation of tomography and reducing measurement times. We also propose a novel approach to QST, based on state-of-the-art deep learning frameworks, which we hope will eliminate the inherent critical challenges of quantum state tomography: bias and scalability.

Bibliography

[1] Eliot Bolduc et al. "Projected gradient descent algorithms for quantum state tomography". In: *npj Quantum Information* 3.1 (2017), p. 44.

[2] Juan Carrasquilla et al. "Reconstructing quantum states with generative models". In: *Nature Machine Intelligence* 1.3 (2019), p. 155.

[3] Jerry Moy Chow. "Quantum information processing with superconducting qubits". PhD thesis. 2010.

[4] Julia Cramer. "Algorithmic speedup and multiplexed readout in scalable circuit QED". MA thesis. 2012.

[5] Christopher Ferrie and Robin Blume-Kohout. "Maximum likelihood quantum state tomography is inadmissible". In: *arXiv preprint arXiv:1808.01072* (2018).

[6] Jun Gao et al. "Experimental Machine Learning of Quantum States". In: *Physical review letters* 120.24 (2018), p. 240501.

[7] Ming Gong et al. "Genuine 12-qubit entanglement on a superconducting quantum processor". In: *Physical Review Letters* 122.11 (2019), p. 110501.

[8] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[9] ID Ivonovic. "Geometrical description of quantal state determination". In: *Journal of Physics A: Mathematical and General* 14.12 (1981), p. 3241.

[10] Matteo Paris and Jaroslav Rehacek. *Quantum state estimation*. Vol. 649. Springer Science & Business Media, 2004.

[11] Yihui Quek, Stanislav Fort, and Hui Khoon Ng. "Adaptive Quantum State Tomography with Neural Networks". In: *arXiv preprint arXiv:1812.06693* (2018).

[12] Roman Schmied. "Quantum state tomography of a single qubit: comparison of methods". In: *Journal of Modern Optics* 63.18 (2016), pp. 1744–1758.

[13] Travis L Scholten and Robin Blume-Kohout. "Behavior of the maximum likelihood in quantum state tomography". In: *New Journal of Physics* 20.2 (2018), p. 023050.

[14] RT Thew et al. "Qudit quantum-state tomography". In: *Physical Review A* 66.1 (2002), p. 012303.

[15] Giacomo Torlai et al. "Many-body quantum state tomography with neural networks". In: *arXiv preprint arXiv:1703.05334* (2017).

[16] Giacomo Torlai et al. "Neural-network quantum state tomography". In: *Nature Physics* 14.5 (2018), p. 447.

[17] Rodney Van Meter and Clare Horsman. "A blueprint for building a quantum computer". In: *Communications of the ACM* 56.10 (2013), pp. 84–93.

[18] Tao Xin et al. "Local-measurement-based quantum state tomography via neural networks". In: *arXiv preprint arXiv:1807.07445* (2018).

[19] Qian Xu and Shuqi Xu. "Neural network state estimation for full quantum state tomography". In: *arXiv preprint arXiv:1811.06654* (2018).

[20] Jun-Yan Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.