**ML4GST**

Author 1, Author 2, Author 3

**Abstract**This is the abstract of the paper. It summarizes the main points and findings of the research.

# 1. Introduction

Content for Introduction

# 2. Problem Setting and Definitions

The goal of tomography in any general setting is to learn a latent space that maximally matches with the observed data . The nature of this latent spacediffers from application to application. For instance, in medical imaging, by aggregating information from multiple 2D sinograms of CT scan, one can reconstruct a full 2D/3D CT image, whereas the dimension of (reconstructed 2D/3D CT image) is the same or higher than the observed data (2D sinograms). Notably, both and reside in the pixel space, that requires little to no transformation when passed to typical neural network models like convolutional neural network (CNN) or diffusion model.

This is vastly different from quantum tomography, where and do not reside in the same space. In the case of quantum state(process) tomography, latent space will be the density(process) matrix that takes the form of a square matrix, where the observed data refers to the probabilities measured from the quantum device, conditioned on a certain measurement and/or preparation operators.

The same mismatch between latent space and observed space continues in gate set tomography, which prevents direct implementation of typical image deep generative models. Instead, an intermediate function has to be used in order to map the latent space to the observed space, an analytical function that maps the neural network output to the expected probabilities. In contrary to most existing publications that directly use deep neural network to reconstruct the full density or process matrix in quantum state or process tomography settings, we aim to predict physical error parameters instead in gate set tomography, and then use analytical function to reconstruct the full process matrices afterwards, as a way to ensure completely positive and trace preserving (CPTP) condition is met and remove the necessity of gauge degree of freedom fixing.

**Background Theory of Gate Set Tomography.** Different from traditional quantum process tomography, which implicitly assumes near zero state preparation and measurement (SPAM) errors [insert a figure], gate set tomography relaxes this assumption by directly incorporating gates as both preparation and measurement operators, or formally as preparation and measurement fiducials. In quantum computer characterization setting, rather than probing each individual gate using traditional process tomography, gate set tomography aims to simultaneously reconstruct the full gate set using maximum likelihood method [ref]. By measuring the outcomes prescribed by a list of gate sequences that acts to amplify errors of each gate, one can run an optimization algorithm to find out all the parameterized process matrices within the gate set [insert a figure]. It is precisely due to this ‘all in one’ tomographic method, that gate set tomography has the highest reconstruction accuracy versus traditional state tomography and process tomography that are largely plagued by the problem of SPAM errors [insert a table][ref]. However, the trade-off of gate set tomography is immediately obvious, where way more computational resources have to be used to solve for this ‘simultaneous maximum likelihood’.

**Proposal.** We try to alleviate this issue by proposing transformer model based deep neural network, which excels in encoding and processing sequences [ref][ref], that has also shown promising results in quantum state tomography setting recently [ref][ref].

# 3. Related Work

**Neural Network Based Quantum Tomography.** The advancement of neural network algorithms and computer hardware in the last decade have brought forth numerous novel applications in the industry, such as autonomous robotics via reinforcement learning [ref], image generation [ref] via diffusion model, text generation via large language model [ref] and so much more. Riding on this trend, quantum physics community has borrowed these techniques from the industry for quantum tomography. Earlier work mainly focuses on using restricted Boltzmann machine for simple quantum state tomography tasks that can be represented by pure states. Later works employ deep neural networks for more difficult tasks such as general density matrix reconstruction [ref] and process matrix construction [ref]. The methods being used ranging from simple feedforward network to more advanced models such as conditional generative adversarial network and transformer model.

**Gate Set Tomography.** There are few works that focus on efficient scaling based on the original of gate set tomography paper. Notably, one paper [ref] proposes compressive gate set tomography via Riemannian optimization. However, no known works to date study gate set tomography from a deep neural network perspective.

# 4. Method

Our ML4GST neural network consists of the following main components: 1) separate embedding layers for both integer-encoded gate sequences and measured probabilities, 2) separate positional encoding for both gate sequences and measured probabilities, 3) cross attention layer to aggregate information from two branches, 4) transformer block to encode the aggregated information, 5) fully connected layer to output physical error parameters. A schematic of the neural network architecture is shown in Fig.1

# 4.1 Separate branch for embedding gate sequences and measured probabilities

In gate set tomography, each gate sequence outputs a set of measured probabilities. By aggregating the information of multiple pairs of (gate sequence, measured probabilities), one can extract the information of the process matrix of each gate used. The gate sequences are first preprocessed by integer encoding and zero padding, where each gate is mapped to a unique integer and the gate sequences are zero padded to match the longest gate sequence in the dataset. The encoded gate sequences are then passed to an embedding layer that is used in typical transformer setting. On the other hand, the measured probabilities are passed to a fully connected layer and subsequent reshaping that emulates the effect of an embedding layer.

This way, both branches will have an extra learnable feature dimension that are ready to be processed by a transformer block later on.

# 4.2 Separate branch for positional encoding

We use the standard sine and cosine positional encoding for both embedded gate sequences and measured probabilities, that are flattened beforehand. As each individual pair of (gate sequence, measured probabilities) yields little tomographic information, we instead group multiple pairs together to increase the receptive field. The positional encoding then runs through element wise for this flattened embedded gate sequences and measured probabilities at each branch.

# 4.3 Cross attention layer

After embedding and positional encoding at each branch, a cross attention layer is used to attend to the relationship between the grouped gate sequences and measured probabilities. We choose cross attention instead of simple concatenation to avoid the vast data shape mismatch between gate sequences and measured probabilities that can possibly drown out the training signal.

# 4.4 Fully connected layers

Finally, multiple fully connected layers are used after the transformer block, to output predicted physical error parameters.

# 5. Experiments

We test out ML4GST using the open-source python package pyGSTi [ref], with the capability of: 1) customizing process matrix of each individual gate within a gate set, 2) selecting appropriate fiducials for the customized gate set, 3) generating appropriate gate sequences for GST experiment, 4) simulating measured counts for the gate sequences.

# 5.1 pyGSTi simulation settings

**1-Qubit XYI model.** pyGSTi python package uses Pauli Transfer Matrix (PTM) as default process matrix throughout GST implementation. Here we replaced the built-in single qubit XYI model pack with our custom PTM, specifically the X and Y rotational gates. These custom gates are parametrized with physical error parameters and in our case, the over-rotational angles and depolarizing errors. We then use the built-in function to find suitable fiducials based on the custom X and Y rotational gates. After that, we run the built-in single qubit XYI GST experiment function to generate gate sequences and simulated measured counts. Number of shots is set to 10k, maximum sequence length to , and sampling error to binomial.

# 5.2 Training details

**Grouping Data.** As mentioned briefly in the method, the gate sequences generated by pyGSTi are converted from strings to unique integers, and subsequently zero padded to match the maximum length of a sequence in the dataset, whereas simulated measured counts are normalized into probabilities. After that, both gate sequences and probabilities datasets are divided into groups, specified by a hyperparameter ‘group\_size’. Typically, the last group will not be divisible and will have a remainder. We simply repeat the elements inside the last group instead of zero padding to preserve overall data quality.

**Curriculum Learning.** We make use of curriculum learning to further divide the whole dataset into parts, again specified by a hyperparameter ‘part\_size’. The dataset is sorted in ascending order based on the non-zero length of gate sequence. This ensures the model learns global features from shorter gate sequences in the beginning and then progressively fine-tune predictions in later stages when it sees longer gate sequences. This learning methodology is similar to the algorithm implemented in the original GST paper [1], where the authors iteratively adding longer sequences accumulatively during optimization. We instead opt for non-accumulative approach in the curriculum learning.

**Analytical PTM Reconstruction.** Based on the predicted physical error parameters, namely the over-rotational angles and depolarizing errors, we analytically reconstruct PTMs corresponding to the gates within the gate set.

**Computing Loss.** As each grouped data outputs one set of physical error parameters prediction, it also has its own set of reconstructed PTMs. We compute probabilities analytically for all the gate sequences within a group, using the same set of reconstructed PTMs. This procedure is performed iteratively group by group within a particular stage set by curriculum learning. Finally, we compute mean squared error loss between the ground-truth probabilities and the reconstructed probabilities. Fig 2. shows the overall data pipeline from input to MSE loss.

# 6. Results

**Choice of Loss Function.**

The original GST paper [1] uses two loss functions for long-sequence GST optimization, the log likelihood function, and the estimator.

where denotes the index of a circuit, and let be the number of outcomes of , the total number of times circuit was repeated, the number of times outcome was observed, the probability predicted by the model of getting outcome from circuit , and is the corresponding observed frequency. The authors made use of estimator as a proxy of during optimization except the last phase, as it is more computationally efficient. Whereas is used in the final phase to steer the estimate to comply with the true statistical derivation. Here, we further simply the estimator to mean-squared error (MSE) loss, that has also been used in simpler linear GST setting. Alternatively, MSE loss can be seen from the perspective of reducing the likelihood function to a normal distribution by invoking central limit theorem [2].

where = is the sampling variance in the measurement.

**Convergence Analysis.**

For depolarizing errors, because we use the tanh activation function at the neural network output layer and subsequently taking absolute value in the custom training loop, the plots shown below will generally have the predicted values jumping between positive and negative. This is intended, as we want the predicted depolarizing error values to be close to and centered at zero, where tanh activation function is the prime candidate.

For both predicted depolarizing errors and over rotational angles, the predicted values exhibit oscillatory behavior at the beginning of each stage of curriculum learning, where an entirely new set of data was fed into the neural network for further training. This is indicated at epoch 90, 190, 263, corresponding to the start of stage 2, 3 and 4. Fig. shows the convergence behavior for depolarizing errors and over rotational angles.

Additionally, we showed that without curriculum learning, the model fails to within the normalized number of epoch, which is equal to the number of epoch for each stage in the curriculum learning. Fig. shows the convergence trajectories without curriculum learning.

**Benchmarking.**

To show that our predicted values are in good agreement with the ground-truth values from simulation, we choose KL divergence**,**  estimator and full Log L function as benchmark. We compare the benchmark results among three cases, ground-truth values, predicted values with curriculum learning and predicted values without curriculum learning.

**KL divergence.**

estimator

Log L

# 7. Conclusion

Content for Conclusion

# A. Appendices

Content for Appendices

[1] E. Nielsen, J. K. Gamble, K. Rudinger, T. Scholten, K. Young, and R. Blume-Kohout, “Gate Set Tomography,” *Quantum*, vol. 5, p. 557, Oct. 2021, doi: 10.22331/q-2021-10-05-557.

[2] D. Greenbaum, “Introduction to Quantum Gate Set Tomography.” arXiv, Sep. 09, 2015. doi: 10.48550/arXiv.1509.02921.