Machine-Learning Enhanced Reconstruction on Quantum States

機器學習增強之量子態重建

專題領域：光電所

組　　別：A222

指導教授：李瑞光

組員姓名：王立愷、許智凱

研究期間： 110年9月 至 111年4月 底止，計 8 個月

Abstract

Quantum state tomography (QST) aims to unveil complete information of a quantum system. However, traditional QST is time-consuming, and the rapidly growth of parameters describing a large Hilbert space causes a great demand on computing resources. In this study, we rebuild a model with convolutional neural network architecture according to the one demonstrated in the paper [1]. Besides, we further modify the architecture. Finally, our model is not only good in the viewpoint of average fidelity, but also have a higher performance in individual inferences. Its needs fewer times of inferences to reconstruct a quantum state, which makes the QST system closer to the edge end.

1. Introduction

Quantum state tomography (QST) enables us to characterize the complete information of a quantum state. It aims to exploit some properties and information in a quantum system. Traditional QST is time-consuming, and the rapidly growth of parameters describing a large Hilbert space boosts a great demand on computing resources. To tackle this problem, recently several machine learning (ML)-based methods are proposed.

In the paper [1], they chose convolution neural network (CNN) to perform the density matrix reconstruction. Our model inherits the architecture of it. We found out with the same inference object, which is the object we study here, squeezed states, the model in paper [1] has average pooling, while the model in paper [3] doesn't have any pooling layers. This motivated us to identify the difference of them and improve the reliability of our model in this way.

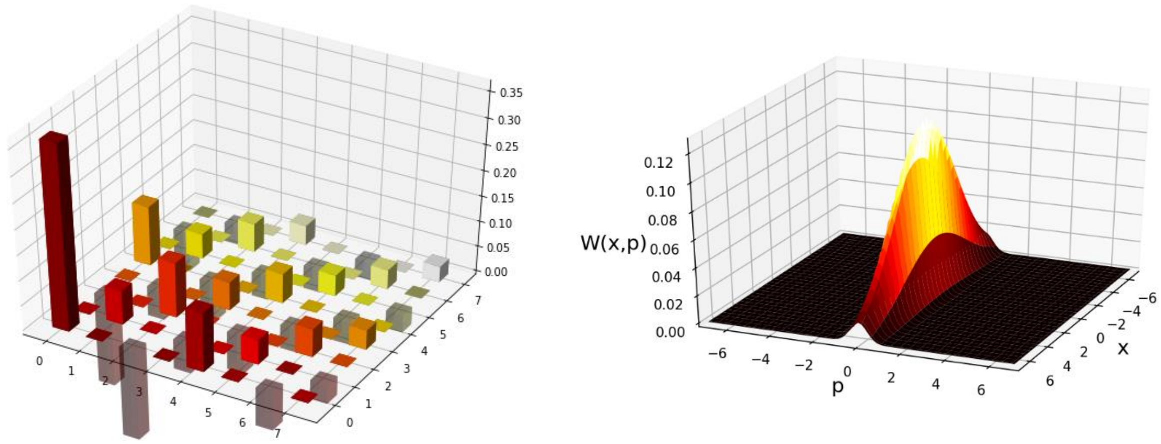

Fig. 1. Visualization of the density matrix (left) and the Wigner function (right) of a 9dB squeezed thermal state.

There are two types of models in this study. One is to construct the density matrix of a state, which we call DTX model. The other one directly infers the parameter of a squeezed state, which we call parameter model.

This study can be divided into two parts. First, we discuss the effects of different hyperparameters in the training process. Second, the crucial part for the improvement, we add in pooling layers in different manners to identify the influence of pooling layers to the performance of our model. At the same time, this testing is performed on both the low squeezing data and high squeezing data to test the impacts of different datasets on the inference results with respect to low and high squeezing levels.

The experiment result shows that the inference fluctuation of the model with pooling is smaller than that without pooling. In high SQ level, such effect is more obvious. In the last, we verify the correctness of the models by SQ/ASQ level reconstruction. The correctness of the two models is very close.

2. Method

2.1 Parameter models and density matrix models

Parameter models are used to acquire the parameters of the input squeezed thermal state, i.e. r, θ, and $\bar{n}$, and density matrix models (DTX models) aim to acquire 35X35 density matrices. In the training process, we feed in 500k to 1200k simulated quadrature sequence data into the model. Each one has the length 4096. After well training, we feed 4096 sampling points from the experimental quadrature sequence data into the models to inference.

2.2 Three datasets

We have two types of simulated data: one is low squeezing and the other one is high squeezing. We name the low squeezing type data “Dataset 1”, high squeezing type data “Dataset 2” and the whole data “Dataset 3”. We want to clarify the performance of the model with certain squeezing interval, so we prepared these datasets to make comparisons.

	Property	Pump power
Dataset 1 (D1)	Low squeezing level data	0.25mW to 5mW
Dataset 2 (D2)	high squeezing level data	30mW to 90mW
Dataset 3 (D3)	Low and high squeezing level data mixed	Consist the above two

Table 1. The dataset types

2.3 Transform the parameters to SQ/ASQ

In the last stage of verification, we transform the squeezed state parameter (r, θ, $\bar{n}$) into the corresponding squeezing (SQ) level and anti-squeezing level (ASQ). For the ideal case, the SQ and ASQ levels should be the same. However, due to the phase noise and loss mechanism, the system exhibits unavoidable degradation [1, 3]. Considering these effects in the reality, we can compute the SQ and ASQ level of a squeezed state.

3. Results

3.1 Hyperparameter testing

This part is to test the effect of each factors in the model training and determine the best hyperparameters for the afterward experiments. We divided the tests into three classes: dataset size, optimizer and with/without pooling layer.

3.1.1 Optimizer

For both the parameter model and DTX model, we tested three types of optimizers, SGD, Adam, and RMSprop. We see that optimizers have a great influence on the loss and the mean square error. Among them, Adam works the best in both the parameter model and the DTX model. Thus we use Adam in the following experiments.

3.1.2 dataset size

We feed 300k~1200k data into the model. The parameter model gets saturated when the data size reaches 500k~ 600k, so we take a conservative estimate that the data size of 720k for the parameter model is enough. In the later study for the parameter model, the quantity of data we used is 720k.

3.1.3 Different pooling architectures

We observed that in the model demonstrated in the paper [1], they use convolution layers with strides in the transition layers (which is composed with batch normalization, activation layers, and a convolution layer.) We thought that such design might cause some information loss. Hence, we added pooling layers in the transition layers with strides in charge of the function of reducing channel width. We choose average pooling as the pooling layers to add in our model, and tried three architectures for both the parameter model and the DTX model. The architectures are mainly different at the transition layers. According to their performance on training set, we choose arc01 to perform the following experiments.

arc01		**arc02**		**arc03**	
Layer Name	parameter	Layer Name	parameter	Layer Name	parameter
BatchNormalization		BatchNormalization		BatchNormalization	
Activation		Activation		Activation	
Conv1D	stride = 1	AveragePooling1D	stride = 4	AveragePooling1D	stride = 1
AveragePooling1D	stride = 4	Conv1D	stride = 1	Conv1D	stride = 4
Activation		Activation		Activation	

Table 2. The transition layer of arc01, arc02, and arc03

3.2 With pooling layers vs without pooling layers

Here we show the experiment result for the dataset 3. Dataset 3 comprises both low and high SQ data, that is, the combination of D1 and D2. Its anti-squeezing level ranges from 0dB to 25dB, namely pump power is from 0.25mW to 90mW. We build a model that can reconstruct both low SQ and high SQ state.

Then, we want to see if the model with pooling had smaller fluctuation. We monitored the inference performance point by point with 2.5mW and 60mW pumping power data in the following graphs. Consider σ, the standard deviation of $\bar{r}$, for every pump power. Note that we focus on $\bar{r}$ since the squeezing parameter $\bar{r}$ is subject to the squeezing level.

For 2.5mW pump power, $\sigma = 0.344$ for the model with pooling.

For 2.5mW pump power, $\sigma = 0.349$ for the model without pooling.

For 60mW pump power, $\sigma = 0.535$ for the model with pooling.

For 60mW pump power, $\sigma = 0.621$ for the model without pooling.

From the results, we conclude that in high SQ level, adding pooling does make the fluctuation of inference smaller, and it also has slight effect at low SQ level.

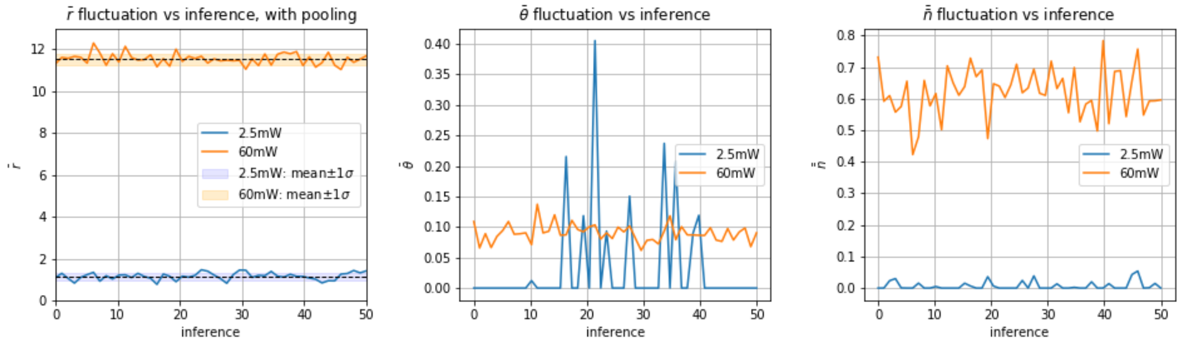

Fig. 3. The inference result of the model with pooling layers. We show the inference values point by point. $\sigma = 0.344$ for 2.5mW pump power, and $\sigma = 0.535$ for 60mW pump power.

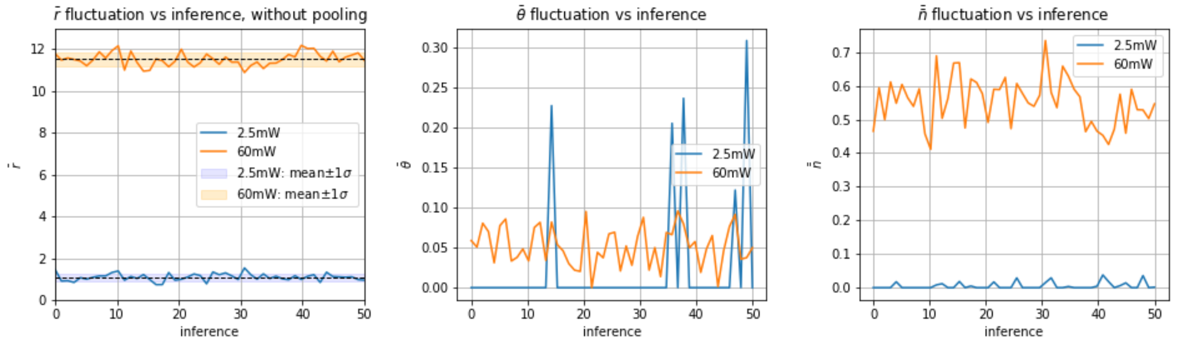

Fig. 4. The inference result of the model without pooling layers. We show the inference values point by point. $\sigma = 0.349$ for 2.5mW pump power, and $\sigma = 0.621$ for 60mW pump power.

3.3 SQ/ASQ reconstruction

We applied validation by SQ/ASQ reconstruction to see which model is better. We transformed the three parameters into SQ/ASQ level in dB by the method mentioned in chapter 2.3 and compared them with the original experiment data (ground truth) to see which is closer to the ground truth. Then we fitted the data points with exponential curves.

We can see that the two fitting curves are nearly the same. We calculated the mean square error (mse) between the inference data points and the experiment data curve. The mse of the model without pooling is 0.631 while the one with pooling is 0.790. The difference between these two errors is really small.

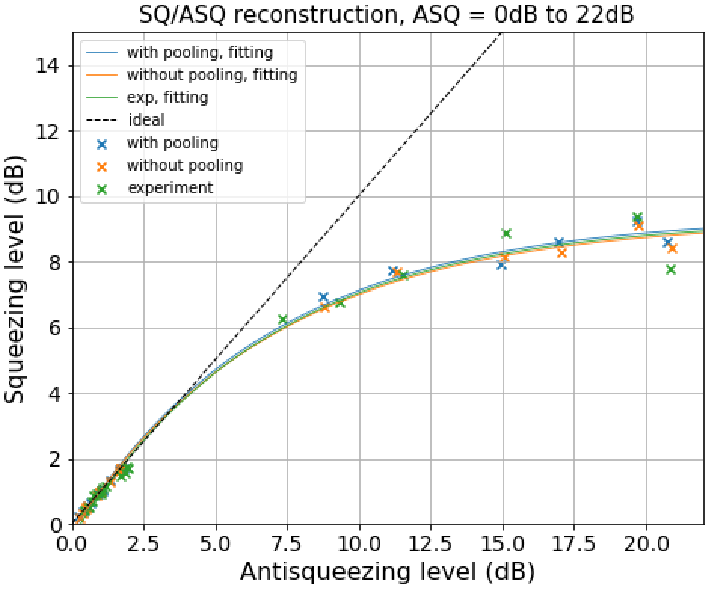

Fig. 5. Compare the models using SQ/ASQ level. There are 21 data fed into both models. The pump power ranges from 0.25mW to 90mW.

We assume there is a tradeoff. Pooling decreases the fluctuation of inference values, letting us get an effective result by fewer inference times. Meanwhile, it discards some information and deviates the inferred parameters from the real experiment value, leading to a larger error in the SQ/ASQ computation. However, this increment of error is small and acceptable.

4. Conclusion

We target to reconstruct quantum states in real time by means of machine learning. While the high fidelity of the original model is on average-sense, we focus on reducing the number of inference times needed for getting a reliable result. Our study makes it closer to fulfilling the demand of real-time QST. First, we introduced the background physics of squeezed thermal state. They can be represented by 3 parameters r, θ, $\bar{n}$ or density matrices. Second, we built our parameter and density matrix models and tuned the hyperparameters. We also introduced our 3 types of dataset: D1 (low SQ), D2 (high SQ) and D3 (low + high SQ). Then, we tried adding pooling layers to the models and trained them with the datasets. We compared the $\bar{r}$ fluctuations over individual inferences and found that adding pooling didn't affect low SQ much but obviously reduce the inference fluctuation of high SQ. Moreover, we transformed the parameters into SQ/ASQ level to see how pooling affects the model performance. We observed that pooling subtly shifts the average inference results away from the real experiment values, and that is a tradeoff. Finally, we successfully built our models that only require 4096 samples of the quadrature sequence data and only few inferences to reconstruct the quantum state, achieving the “real-time” goal.

5. References

[1] H.-Y. Hsieh, Y.-R. Chen, H.-C. Wu, H. L. Chen, J. Ning, Y.-C. Huang, C.-M. Wu, and R.-K. Lee, “Extract the Degradation Information in Squeezed States with Machine Learning”, Phys. Rev. Lett. **128**, 073604 (2022)

[2] Y.-R. Chen, H.-Y. Hsieh, J. Ning, H.-C. Wu, H. L. Chen, Y.-L. Chuang, P. Yang, O. Steuernagel, C.-M. Wu, and R.-K. Lee, “Experimental reconstruction of Wigner distribution currents in quantum phase space,” arXiv:2111.08285 (2021).

[3] H.-Y. Hsieh, J. Ning, Y.-R. Chen, H.-C. Wu, H.- L Chen, C.-M. Wu, and R.-K. Lee, “Direct parameter estimations from machine-learning enhanced quantum state tomography”, Symmetry 14, 874 (2022)

[4] Akira Furusawa, “Quantum States of Light”, Volume 10 (2015)

[5] H. Seifoory, S. Doutre, M. M. Dignam, and J. E. Sipe, "Squeezed thermal states: the result of parametric down conversion in lossy cavities," J. Opt. Soc. Am. B 34, 1587-1596 (2017)

[6] A. I. Lvovsky and M. G. Raymer, “Continuous-variable optical quantum-state tomography”, Rev. Mod. Phys. 81, 299 (2009)

6. 心得感想

起初，覺得這個量子物理加機器學習的題目很酷，便選擇了這份專題。開始了後還真的是什麼都不懂，為此兩人都花了一個多學期去修課來學習相關知識，這一學期才開始有真正的研究進度。在過程中，我們做了許多測試，藉由瞭解模型的細節表現，來尋找可以改進的空間。也探討了模型在不同資料集與不同的小範圍結構更改下，面對不同壓縮態的表現。但其實到後期，我們一直都還覺得自己沒有做出實際貢獻，因為模型的大架構是基於前人的研究結果，我們大多只是調參數做測試。直到非常後期我們嘗試了新的物理假設，讓推論的參數變成五個，此時我們才認為比較有創新的東西。惟可惜此部分在訓練過程遇到難關，研究還未能有初步的成果。

我們由衷感謝中間過程不斷給予我們協助、指導我們的博班謝憲毅學長。即使他在口試要到了論文也還沒趕出來的階段，他還是很在乎我們的研究，給予我們建議。在此祝福他博士口試順利通過。