CPEN400Q W2023

Companion Report on Identifying Overparameterization in Quantum Circuit Born Machines

sonozaki@ubc.ca

sgonugun@ubc.ca

haria01@ubc.ca

oreade16@ubc.ca

Table of Contents

Theory

The theoretical framework of the paper "Identifying overparameterization in Quantum Circuit Born Machines" intricately weaves together the domains of quantum computing, machine learning, and statistical mechanics to explore the phenomenon of overparameterization in quantum circuits. Here, the core concept of overparameterization is expanded from its classical roots to a quantum context, specifically within the realm of Quantum Circuit Born Machines (QCBMs).

- Overparameterization in Classical and Quantum Contexts

Classical Machine Learning: Overparameterization in classical machine learning refers to a model having more parameters than necessary to fit the training data. Traditionally, this should lead to overfitting; however, paradoxically, it often results in better generalization and a smoother loss landscape, facilitating easier optimization. Papers such as by Zhang et al. (2021) and Neyshabur et al. (2018) have explored these phenomena, demonstrating that overparameterized networks tend to converge to solutions that generalize well despite their capacity to fit the training data perfectly [1,2].

Extension to Quantum Computing: In quantum computing, particularly in the training of QCBMs, the paper hypothesizes that a similar phenomenon occurs. The study references foundational works on QCBMs, such as those by Benedetti et al. (2019) and Hamilton et al. (2019), which establish the basis for using quantum circuits as generative models [3,4]. These QCBMs are trained to mimic complex probability distributions, which can be highly beneficial for tasks in quantum simulation and optimization.

- Theoretical Analysis of Overparameterization in QCBMs

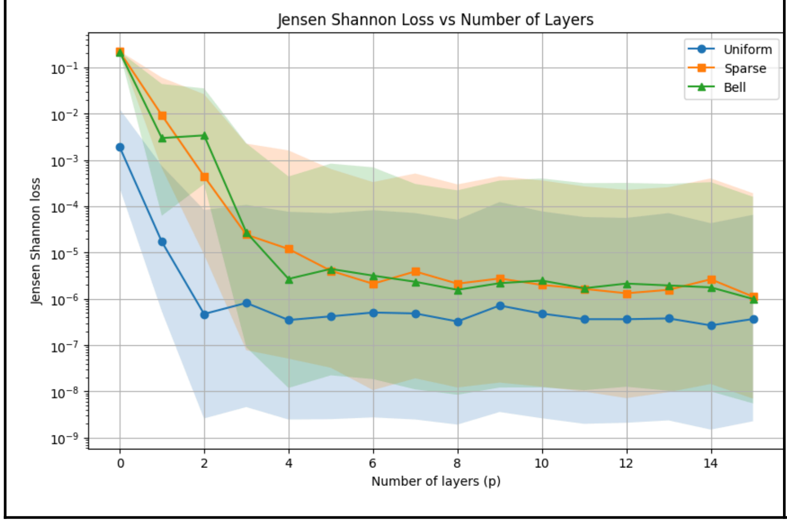
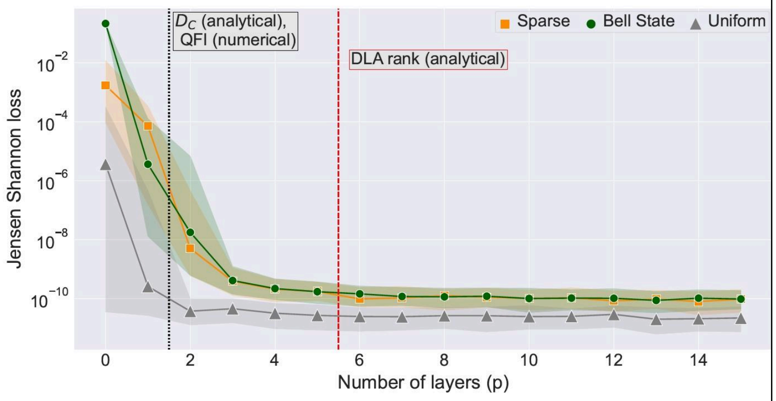
Quantum Circuit Complexity: The paper delves into how the complexity of a quantum circuit, measured in terms of the number of gates or depth, affects its ability to model probability distributions. The reference to works by Harrow and Napp (2021) illustrates the point that lower circuit complexity can limit a model's capacity, while increased complexity, up to a point, enhances it [5].

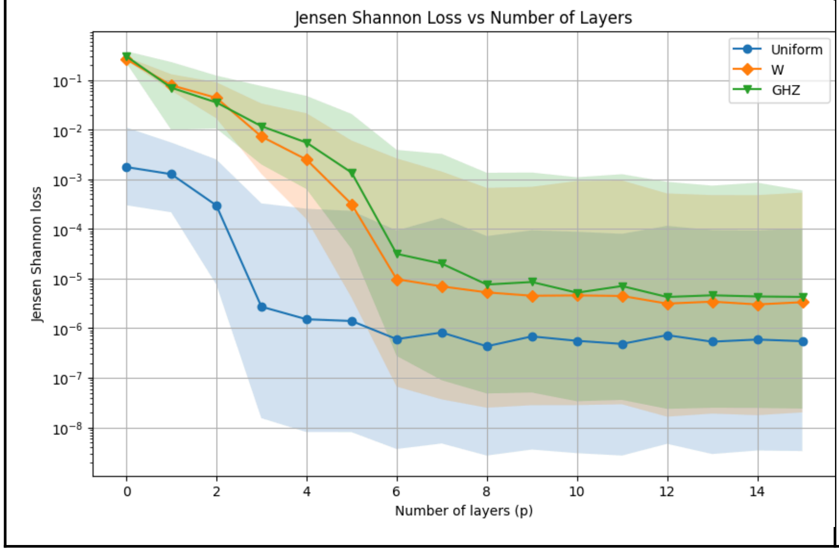
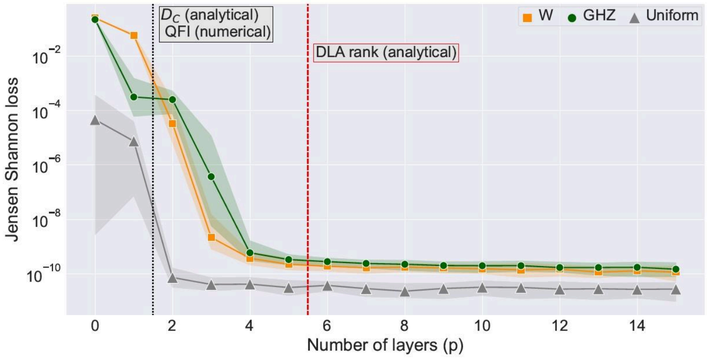
Jensen-Shannon Divergence: To measure the effectiveness of the QCBMs, the study employs the Jensen-Shannon Divergence (JSD), a symmetric measure of the similarity between two probability distributions. This divergence has been widely used in the analysis of classical generative models but is particularly apt for quantum models due to its bounded nature, which offers computational stability. Goodfellow et al. (2014) discuss the advantages of using JSD in training generative adversarial networks, which parallels the training of QCBMs [6].

Empirical Risk Minimization: In discussing the empirical risk minimization, the paper cites significant quantum computing studies that analyze the loss landscapes of quantum models. References include works by McClean et al. (2018) and Cerezo et al. (2021), which explore the occurrence of barren plateaus in quantum neural networks [7,8]. These studies are critical as they suggest that overparameterization could potentially mitigate the issues posed by barren plateaus by smoothing the loss landscape, thus enhancing the trainability of QCBMs.

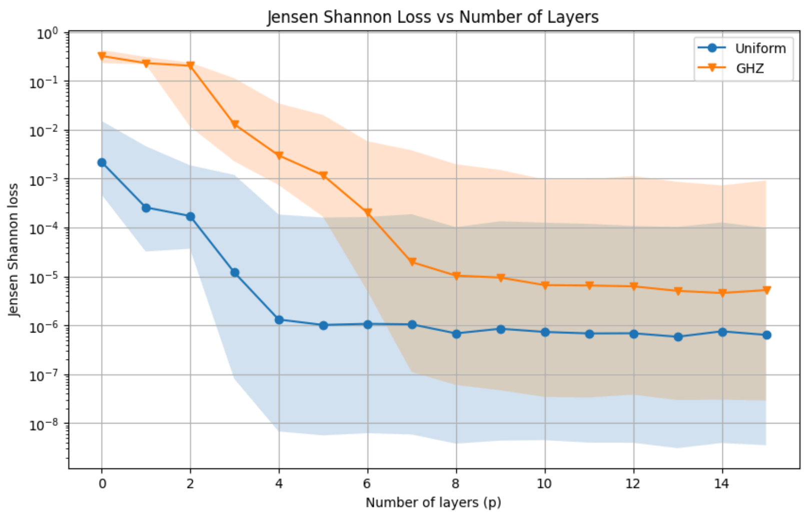
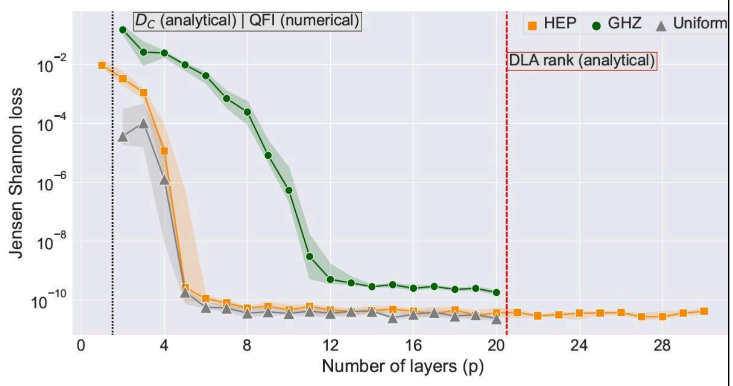
Statistical Mechanics Analyses: Drawing from statistical mechanics, the paper theorizes that the overparameterization in QCBMs may lead to a phase transition in the model's behavior. References to studies on the energy landscapes in physics, such as those by Biroli et al. (2018), provide a theoretical underpinning for how increasing parameter space could transition the system into a new state where optimization becomes more feasible [9].

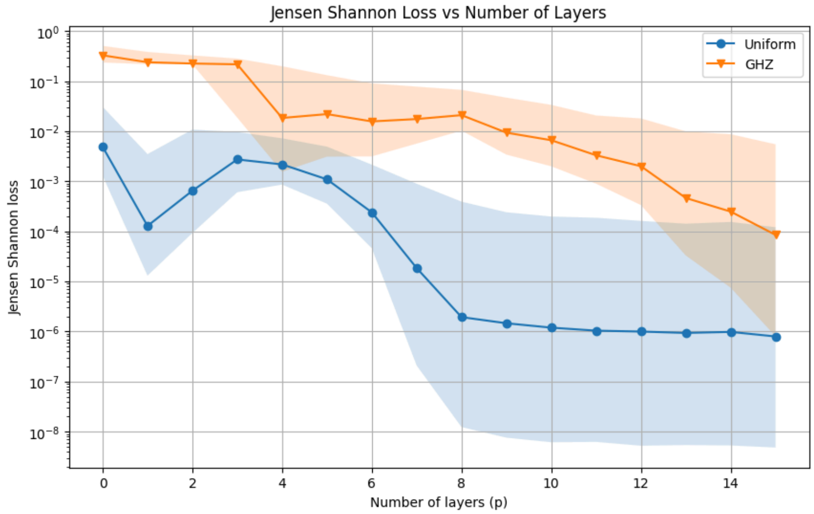
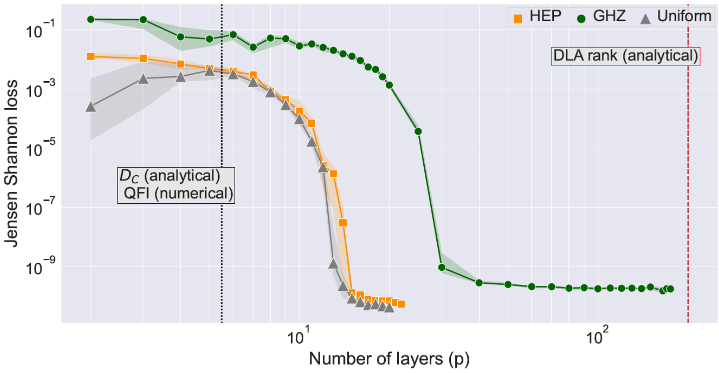
Result

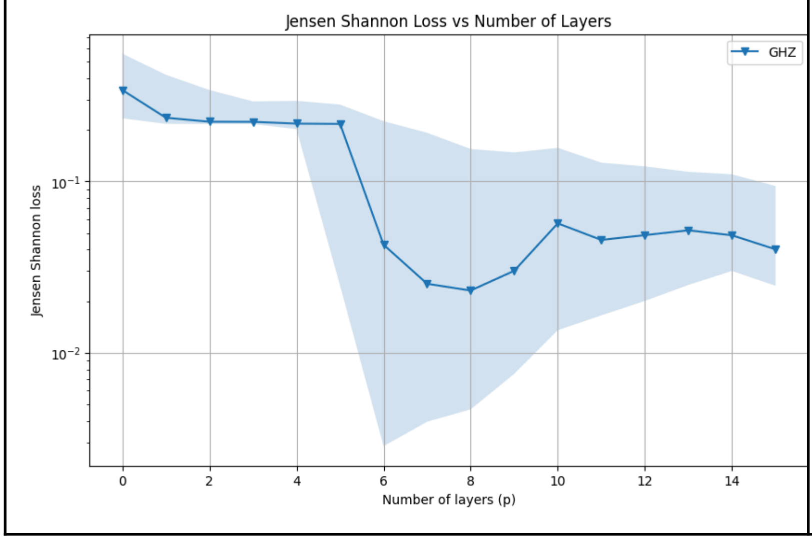
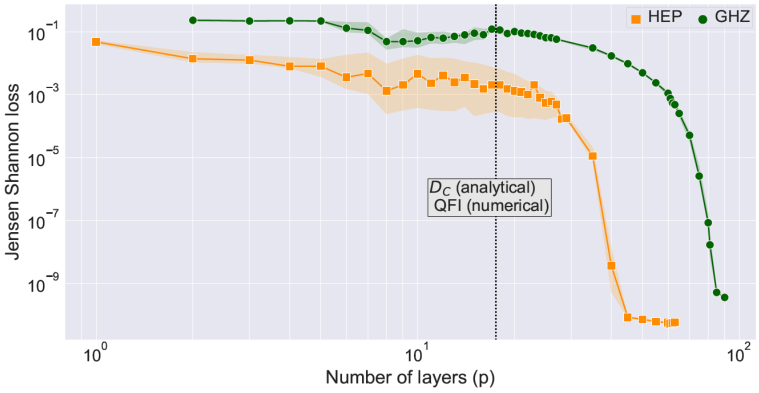
Our result	Paper result (Figure 2 on paper)
	

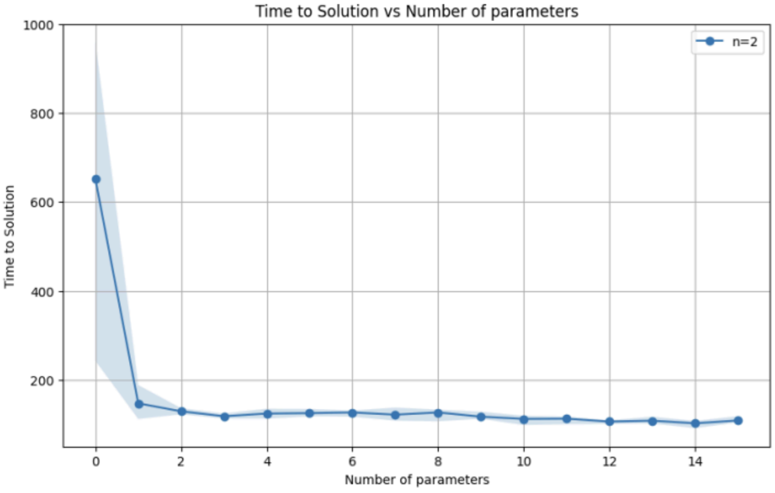
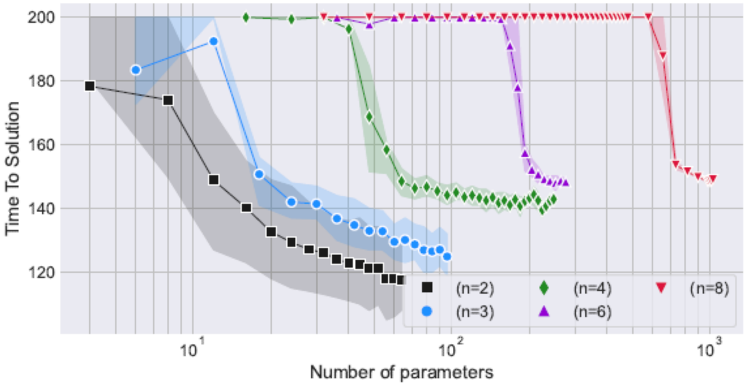
Our result	Paper result (Figure 3 on paper)
	

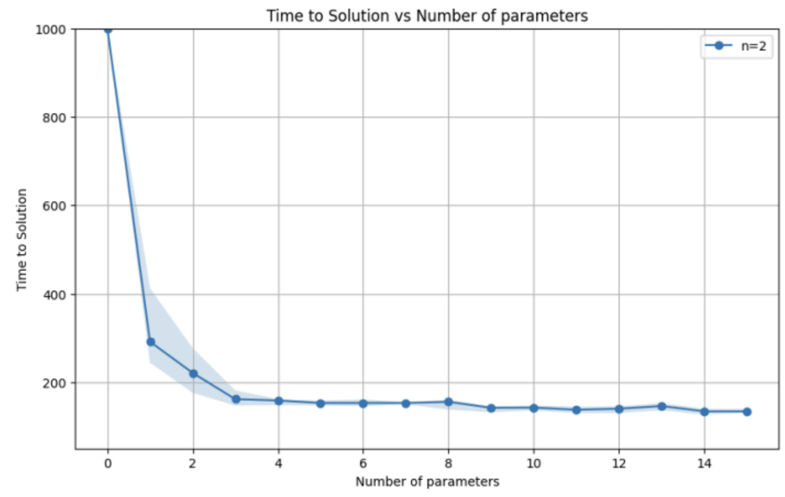
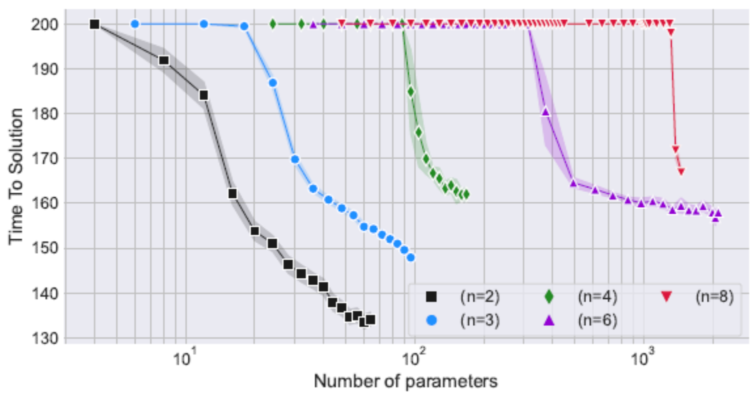
We had to make adjustments to the experiment parameters, where QCBM training runs were reduced to 10 instead of 100 on the paper due to time and computing resource limitations in running the experiment. We have kept the same number of layers for Figure 2-3.

Our result	Paper result (Figure 4 on paper)
	

Our result	Paper result (Figure 5 on paper)
	

Our result	Paper result (Figure 6 on paper)
	

Our result	Paper result (Figure 7 on paper)
	

Our result	Paper result (Figure 8 on paper)
	

The experiments for Figure 4-6, we had to limit the number of layers to 15 as it was not possible to run QCBM for a high number of layers like 100, so our result would only show portions of the results from the paper. Additionally, HEP target distribution is not included in our result because we did not have access to the sample data from the LHC that was large enough and close to the experiment the paper mentions.

The overall trend of the Jensen-Shannon (JS) loss decreasing as the number of layers increases is consistent with the paper's findings. The plateau layer, where the JS loss saturates to a low value, is also consistent with the paper for the 4-qubit case. However, there are some differences observed in the specific JS loss values compared to the paper's plots. For example, to consistently achieve a JS loss below 10^-8 across multiple training runs, around 300 runs were required, which is higher than the paper's results. These differences could potentially arise from variations in the implementation details of the ansatz circuit used, as the paper only explicitly mentions the hardware-efficient ansatz for the 4-qubit case. The ansatz implementation for other qubit counts may differ between the paper and the experiments conducted here.

We were unable to produce results for Figures 7 and 8. This underlines the importance of sufficient computational resources and time in conducting comprehensive and robust quantum computing experiments.

Framework

The framework of the Quantum Circuit Born Machines is going to be expanded upon in this chapter. We use the following libraries and for the following reasons:

1. Pennylane: This is used for creating the quantum circuits as well as quantum machine learning

2. Numpy: The Numpy library is used for numerical computing which is imported as np from pennylane.
3. Matplotlib: The Matplotlib library for data visualization, which is imported as matplotlib.pyplot as plt
4. Tqdm: The tqdm library is used for tracking progress in the Jupyter notebooks
5. JSON: The json library for data serialization and deserialization for saving our data as JSON.
6. Pandas: The pandas library for showing data in tables format.

After all installations and imports have been done, then the data distributions are replicated in python. These include a uniform distribution, a sparse distribution, a Bell state, a GHZ state, a W state and a HEP.

1. Uniform Distribution: A uniform distribution is generated over 2 raised to the power of n qubits bitstrings. The uniform distribution is statistically a distribution with equal likely outcomes for each outcome. These probabilities sum up to 1. The size is 2 raised to the power n_qubits, and each probability is 1 divided by 2 raised to the power n_qubits
2. Sparse Distribution: A sparse distribution is a distribution where the sparsity is the number of non-zero probabilities. The non-zero probabilities are all equal. The choice of the zero probabilities is random, and as for the uniform distribution, these probabilities sum up to 1.
3. Bell State Distribution: According to Wikipedia, the Bell State distribution are four specific maximally entangled states of two qubits. This was built by setting the probabilities of the |00> and |11> states to 0.5 and 0.5 respectively. Like both the uniform and spare distributions, the probabilities must sum up to 1. The Bell Distribution could be explained as a sparse distribution where only specific bit strings have non-zero probability amplitudes.
4. GHZ state: The Greenberger–Horne–Zeilinger state is a three qubit state which can be defined as follows:

$$|\mathrm{GHZ}_3\rangle = \frac{1}{\sqrt{2}}\left(|000\rangle + |111\rangle\right)$$

We set the first and last bits to be of 0.5 probability.

5. W state: The W state is also a three qubit state. For this can be defined as:

$$|W_3\rangle = \frac{1}{\sqrt{3}}\left(|100\rangle + |010\rangle + |001\rangle\right)$$

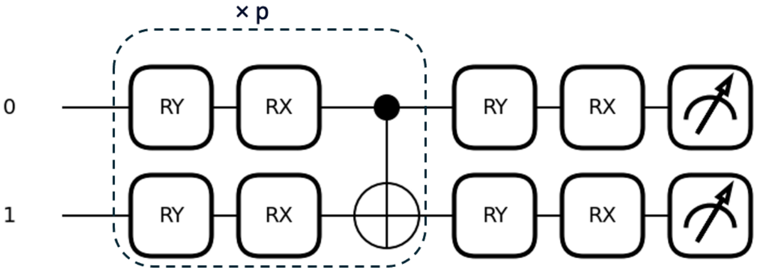
The bits 1,2 and 4 have a one-third probability added to each, which sum up to 1. These bits are |001>, |010> and |100>.

6. HEP distribution: The High Energy Physics distribution is a distribution that models scenarios relevant to particle physics, such as collisions and delay processes. A simplified function of a HEP distribution is done by creating a random distribution and then normalizing it.

After this, the ansatze are drawn up for 2 bits, 3 bits, 4 bits, 6 bits and 8 bits. Below will be a description of the ansatze of all the 2 bits.

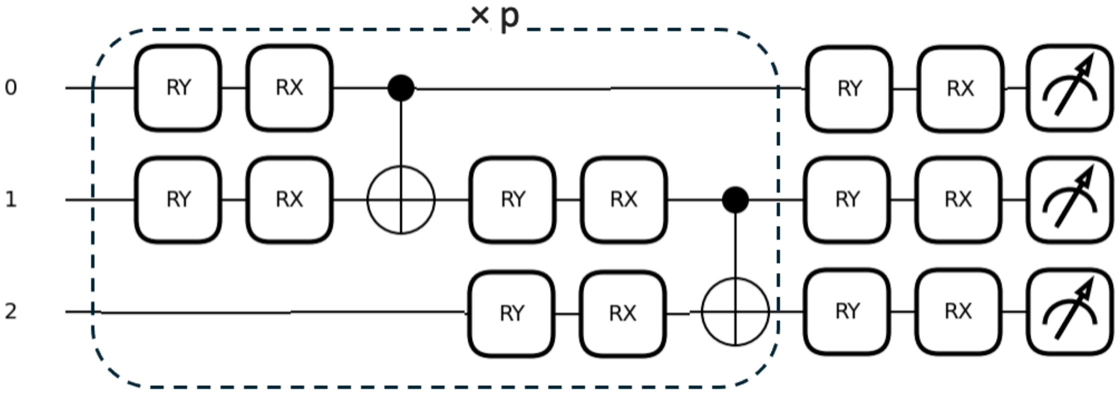
1. Two bits: The two bits ansatze includes a Rotation-Y on the first and second bit and Rotation-X on the first and second bit for one layer. The number of layers p shows the number of Rotation-Y and Rotation-X gates that there are. There is a CNOT gate in each

layer. After the layers have been introduced, there is a final addition of Ry and Rx on the first and second bits before they are measured.



Hardware-Efficient Ansatz for Two Qubits

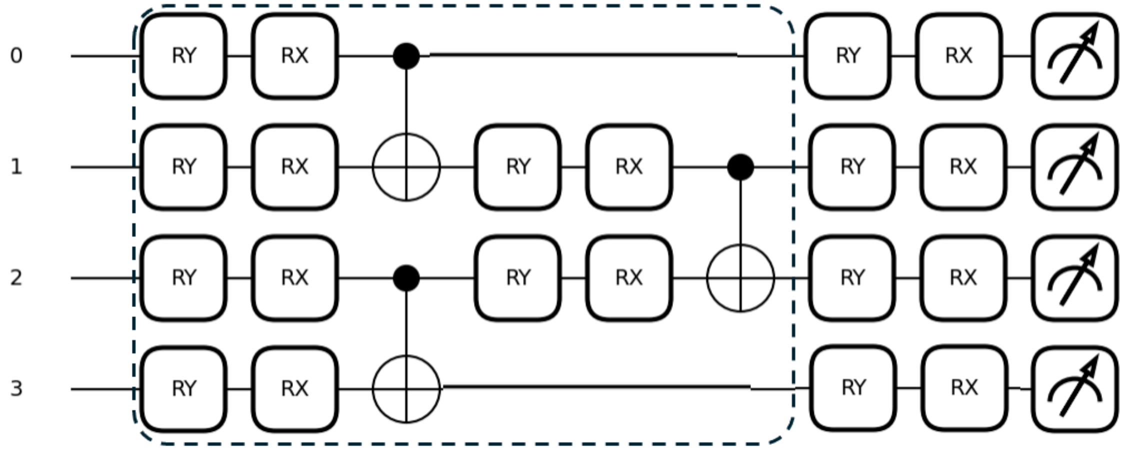
2. Three bits: The three bits ansatze includes a Rotation-Y on the first and second bit and Rotation-X on the first and second bit. After this, there is a CNOT gate. Then there is a RY and RX for the second and third bit. Then there is a CNOT gate. The number of layers p shows the number of Rotation-Y and Rotation-X gates that there are. After the layers have been introduced, there is a final addition of Ry and Rx on the first and second bits before they are measured.



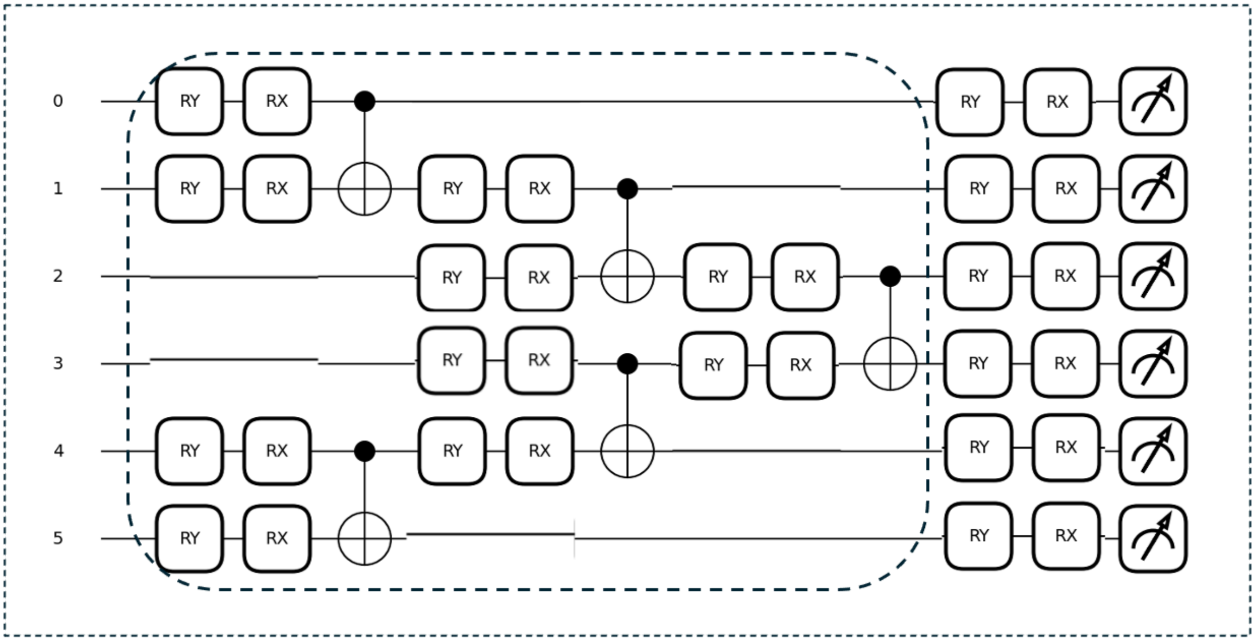
Hardware-Efficient Ansatz for Three Qubits

3. Four bits: The four bits ansatz include a RY on all four bits, then RX on all four bits. After this, there are two CNOT gates being controlled on the second and fourth qubits and controlled by the first and third qubits. After this, RY and RX gates are placed on qubit 1 and 2. After this, the CNOT gate is created. Then the RY and RX gates are on all four qubits, as was done for two qubits and three qubits.

Hardware-Efficient Ansatz for Four Qubits



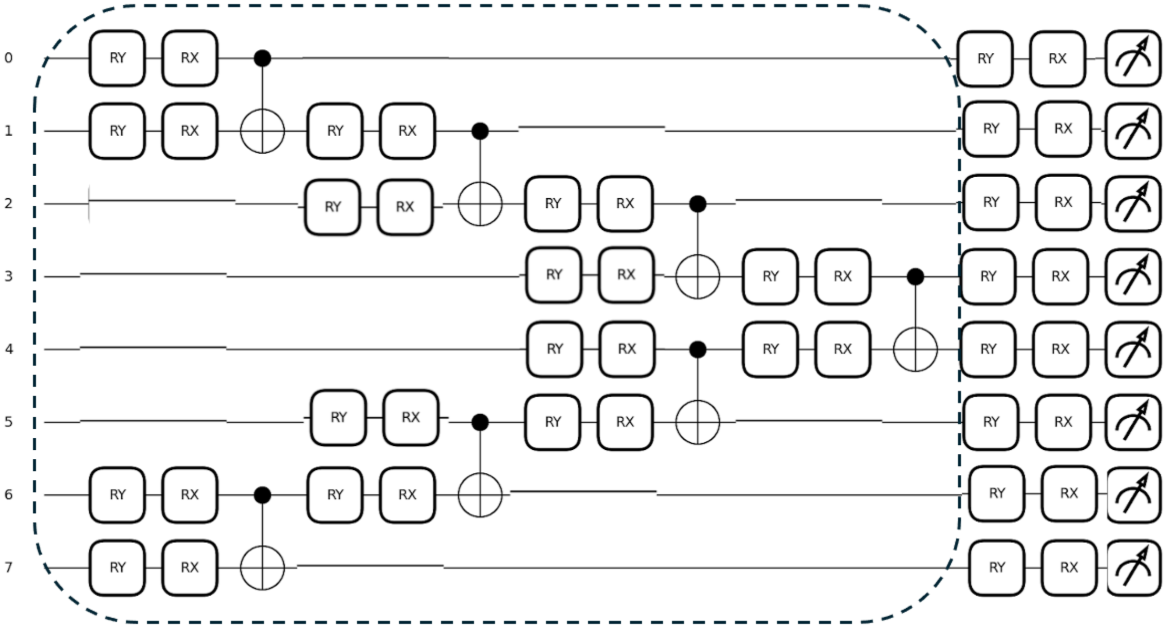
4. Six bits: The six bits ansatz include a RY and RX on the qubits 0,1,4 and 5. The CNOT gates are then also placed on bits 0,1,4 and 5 respectively. After this, RY and RX are also added on qubits 1,2,3, and 4, with two respective CNOT gates. After this, there are RY and RX on qubits 2 and 3, as well as CNOT gates controlled by the qubit 2 and on the qubit 3. After this is a set of RY and RX gates on all the qubits.



Hardware-Efficient Ansatz for Six Qubits

5. Eight Bits: The eight bits ansatz include a RY and RX on the qubits 0,1,6 and 7. The CNOT gates are then also placed on bits 0,1,6 and 7 respectively. After this, RY and RX are also added on qubits 1,2,5, and 6, with two respective CNOT gates. After this, RY and RX are also added on qubits 2,3,4 and 5, with two respective CNOT gates.

After this, there are RY and RX on qubits 3 and 4, as well as CNOT gates controlled by the qubit 3 and on the qubit 4. After this is a set of RY and RX gates on all the qubits.



Hardware-Efficient Ansatz for Eight Qubits

After this, we calculate the Kullback-Leibler Divergence between two probability distributions P and Q. Then Jensen-Shannon Divergence is calculated. JSD is a symmetrized and smoothed version of KLD, measuring the similarity between two probability distributions. KLD measures the difference between two probability distributions, quantifying the information lost when Q is used to approximate P.

Other parts of the QCBM include:

1. The QCBM optimizer: The optimizer function optimizes the parameters of a QCBM to minimize the loss between the generated probability distribution and a given dataset. The function creates a quantum device using qml.device() with the specified number of qubits. The number of qubits, number of layers and the dataset are the inputs, while the output is the dataset, the optimized probability distribution and the list of loss values over time. The QCBM optimizer framework in quantum computing employs a variational approach using parametrized quantum circuits to emulate target probability distributions, optimizing parameters through gradient descent with the Adam optimizer. Initialization starts with qubits in a ground state, and an ansatz of quantum gates iteratively adjusts to minimize the loss against a given dataset over a fixed number of iterations. The process is monitored by tracking the reduction in loss, indicating the quantum model's increasing accuracy in approximating the desired distribution.
2. Run_optimizer_for_dataset function: The run_optimizer_for_dataset function in the notebook executes the Quantum Circuit Born Machine (QCBM) optimization multiple times across a range of circuit layers to determine the ideal quantum circuit configuration for approximating a given probability distribution. It compiles the final loss values for each configuration into a dictionary and computes statistical measures—specifically the

median and interquartile range (IQR)—to capture the central tendency and dispersion of optimization performance. This iterative process is designed to assess the impact of circuit depth on the QCBM's ability to model the target dataset, thereby informing the selection of an optimal number of layers for the quantum model.

Reproducibility

In our pursuit to replicate the findings of "Identifying Overparameterization in Quantum Circuit Born Machines," we encountered challenges that illuminate key factors in the reproducibility of complex quantum machine learning (QML) experiments. This section details these challenges and their implications.

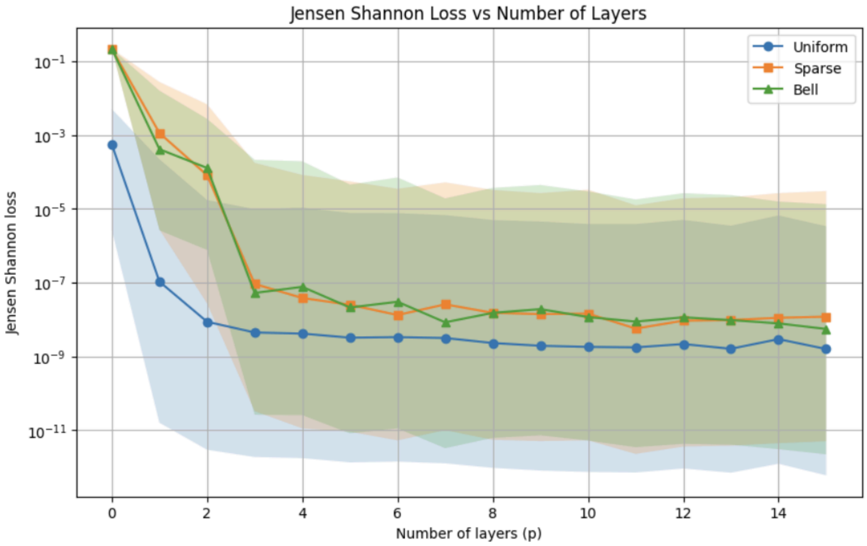
Computational Resources: A significant limiting factor was computational resources. The paper's experiments were extensive, often using a vast parameter space to train Quantum Circuit Born Machines (QCBMs). These models, which require considerable computational power, were explored in the paper with parameters in the thousands. In contrast, our limited computational capabilities meant that we had to substantially reduce the number of parameters for our QCBMs.

Implementation Details: Moreover, the original paper provided certain implementation details that we closely adhered to, such as the use of the hardware-efficient ansatz and the Adam optimizer for training the QCBMs. However, subtleties in the implementation of the ansatz for different qubit counts may lead to variances in our results compared to those reported in the paper. For instance, discrepancies in the specific Jensen-Shannon (JS) loss values and their distributions across training runs could potentially stem from the different ways the ansatz circuits were realized in our experiments. We noticed that to achieve consistently low JS loss values, we required a greater number of runs, suggesting that our implementation perhaps did not navigate the optimization landscape as efficiently as the paper's models.

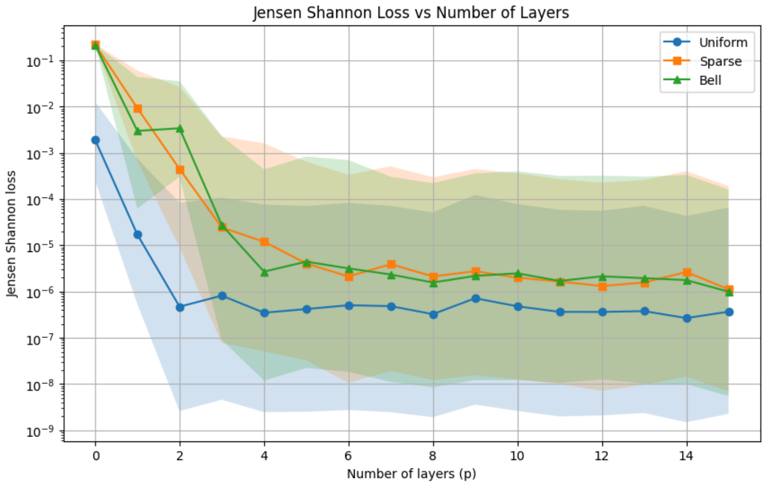
Trends in Jensen-Shannon Loss: We observed that the Jensen-Shannon (JS) loss decreases as the number of layers in the Quantum Circuit Born Machine (QCBM) increases, aligning with the paper's findings. Notably, the identification of a plateau layer, where the JS loss stabilizes at a low value, was also in harmony with the original research for the 4-qubit case. This behavior supports the hypothesis that increasing the depth of QCBMs improves their expressibility and allows for better approximation of the target probability distributions, up to a point of saturation. However,

Discrepancies in Loss Values and Training Runs: Discrepancies arose when comparing the specific JS loss values and the number of training runs needed to achieve them. Our experiments required approximately 300 runs to consistently achieve a JS loss below 10^-8 , significantly more than the paper's reported results. This deviation is indicative of the nuanced influence of ansatz circuit implementation on the model's performance.

Here is the Median (markers) and IQR (shaded) of final losses as a function of number of layers with 300 iterations:



Here is the Median (markers) and IQR (shaded) of final losses as a function of number of layers with 200 iterations:



The original paper elaborated on the hardware-efficient ansatz specifically for the 4-qubit case, but not for other configurations. In our reproduction efforts, we inferred the ansatz implementations for different qubit counts, potentially leading to variations in the expressibility and trainability of the QCBMs. Consequently, our QCBMs may have navigated a more complex or rugged loss landscape, necessitating additional training runs to find low-loss solutions.

Time to Solution and Parameter Counts: The paper establishes a Time to Solution (TTS) for a specified epsilon of 10^-8 within 200 training steps. In our reproduction experiments, a 2-qubit circuit with 15 parameters and 10 training runs required 300 steps to reach the same epsilon, underlining a substantial increase in optimization efforts. This sensitivity is attributed to overparameterization, where a higher parameter count often suggests a more flexible model that can efficiently explore the optimization landscape.

Empirical Adjustments: In the course of our experiments, we also had to make empirical adjustments, such as limiting the number of layers in the QCBMs due to the prohibitive computational cost of deeper circuits. The original paper was able to explore a more extensive range of circuit depths, thereby capturing more nuanced behavior as the number of layers increased. Our constraints meant that we could not fully investigate the plateauing behavior of the JS loss or confirm the presence of the overparameterization transition with the same level of certainty.

External Data Accessibility: The lack of access to certain datasets, such as those from the Large Hadron Collider for High Energy Physics (HEP) distributions, also posed challenges. While the paper used these complex distributions to benchmark the QCBMs, our inability to access similar data restricted us to simpler distributions that may not have highlighted the capabilities of the QCBMs as effectively.

Opinion and interpretation

Introduction and Summary

The paper talks about Quantum Circuit Born Machines (QCBMs) they are generative models that are used to fit arbitrary probability distributions. The paper studies the overparameterization in QCBMs trained using gradient based methods. The key findings are:

- There exists a critical circuit depth p_c above which the QCBM exhibits overparameterization.
- p_c depends on target distribution and the number of qubits. Entangled states take more layers compared to non-entangled states.

Strengths and Contributions

The paper makes several important contributions to understanding overparameterization in QCBMs

- To my knowledge, this is the first work characterizing overparameterization in QCBMs that relates critical depth to target distribution and qubit size.
- It identifies a critical circuit depth p_c above which the gradient descent reliably converges to low loss solutions.
- It shows that for small QCBMs with N<= 4 qubits, it provides approximate analytical bounds for the p_c based on the parameter dimension D_C and the rank of Quantum Fisher Information matrix.

Limitations and Open Questions

The paper has several limitations that highlight important open question for future research:

- The Dynamical Lie Algebra dimension is shown to be only a loose upper bound for p_c for all qubit sizes showing that the theoretical bounds are insufficient to fully predict the bounds.
- Understanding what drives the overparameterization and the behaviours of long-time training dynamics remain an open question.
- The paper focuses on ideal, noiseless QCBM - it might be a good future direction to extend the analysis to noisy quantum systems, thus giving us data for realistic settings.
- The paper provides a conclusion for QCBM which is supposed to work for any distribution, so it might be good to have more distribution types to test if these results are general.

Prior Work

The paper builds on related work such as

- Theory of overparameterization in quantum neural networks, which looks at the loss landscapes of QNNs by studying overparameterization.

This paper is novel in focusing on overparameterization in QCBMs

Future Direction

Here are some ideas for future direction in expanding upon this paper

- This paper’s simulation can be run on real systems such that these results can be expanded upon realistic systems.
- The paper’s simulation can be run on noise simulated systems to account for real scenarios.
- To generalize better on arbitrary distributions, the paper can expand upon the scope of the distributions.

References

1. Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals, “Understanding deep learning requires rethinking generalization,” Communications of the ACM, 64, 107–115 (2021).
2. Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro, "Exploring generalization in deep learning," NeurIPS (2017).
3. Marcello Benedetti, John Realpe-Gómez, Rupak Biswas, and Alejandro Perdomo-Ortiz, "Quantum-assisted learning of graphical models with arbitrary pairwise connectivity," Phys. Rev. X (2019).
4. Kathleen E. Hamilton, Andrea J. Miszczak, and Michał Oszmaniec, "Quantum machine learning with parameterized quantum circuits," Quantum (2019).
5. Aram W. Harrow and John C. Napp, "Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms," Phys. Rev. Lett., 126, 140502 (2021).
6. Ian J. Goodfellow et al., "Generative Adversarial Nets," NIPS (2014).
7. Jarrod R. McClean et al., "Barren plateaus in quantum neural network training landscapes," Nature Communications (2018).
8. M. Cerezo et al., "Cost-Function-Dependent Barren Plateaus in Shallow Quantum Neural Networks," arXiv preprint arXiv:2001.00550 (2021).
9. Giulio Biroli, Jean-Philippe Bouchaud, and Marc Potters, "Complex landscapes and random matrices: Barriers, saddles and degeneracies," Phys. Rev. E (2018).
10. Chen, J. (n.d.). *Uniform distribution: Definition, how it works, and example*. Investopedia. https://www.investopedia.com/terms/u/uniform-distribution.asp#:~:text=Uniform%20distributions%20are%20probability%20distributions,the%20mean%20occur%20more%20frequently.
11. Delgado, Andrea, Francisco Rios, and Kathleen E. Hamilton. "Identifying overparameterization in Quantum Circuit Born Machines." arXiv preprint arXiv:2307.03292 (2023).
12. https://www.connectedpapers.com/main/d87ebfec8e8d99f44d637f67deb3f3dba903946f/Identifying-overparameterization-in-Quantum-Circuit-Born-Machines/prior