UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ismael Cesar da Silva Araujo

Adaptive Ansatz based on Low-Rank state preparation

Recife

2023

Ismael Cesar da Silva Araujo

Adaptive Ansatz based on Low-Rank state preparation

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Inteligência Computacional

Orientador (a): Adenilton José da Silva

Recife

2023

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

A663a Araújo, Ismael Cesar da Silva

Adaptive ansatz based on low-rank state preparation / Ismael Cesar da Silva Araújo. – 2023.

52 f.: il., fig., tab.

Orientador: Adenilton José da Silva.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2023.

Inclui referências.

1. Inteligência computacional. 2. Aprendizagem de máquina. I. Silva, Adenilton José da (orientador). II. Título.

006.31 CDD (23. ed.) UFPE - CCEN 2023-180

Ismael Cesar da Silva Araújo

“Adaptive Ansatz based on Low-Rank state preparation”

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Inteligência Computacional

Aprovado em: 4 de agosto de 2023.

BANCA EXAMINADORA

Prof. Adenilton José da Silva
Centro de Informática / UFPE
(**Orientado**r)

Prof. Dr. Israel Ferraz de Araújo
Yonsei University - Coreia do Sul

Prof. Dr. Askery Alexandre Canabarro Barbosa da Silva
Instituto de Física / UFAL

Aos meus pais, por sempre terem acreditado em mim.

"The fear of the LORD is the beginning of wisdom: a good understanding have all they that do his commandments: his praise endureth for ever." (HOLLY..., 1769)

ABSTRACT

Quantum State Preparation Algorithms consist of defining a sequence or unitary operations to load a specific target state on a quantum computer. We can use those algorithms in applications such as quantum machine learning. However, some state preparation algorithms have exponential circuit complexity with the number of qubits on the system. That is the case of amplitude encoding algorithms, which is an encoding type for loading normalized data into the probability amplitudes of the state. To circumvent this overhead in circuits' complexity, works explore specific properties of quantum states to optimize the circuit's complexity, such as sparsity or symmetry. Other works explore simplifying the quantum circuit to load an approximate quantum state. It is the case of Quantum Generative Adversarial Networks, which use a specific circuit architecture comprised of alternating blocks of single-qubit rotations and two-qubit entangling controlled gates. But when trained to load random distributions on, we observed the performance deteriorates as the number of qubits increases in terms of relative entropy. In this work, we propose different architectures for the Quantum Generative models based on the state preparation algorithm known as Low-Rank. Through experiments for loading the log-normal distribution, we show error reductions in quantum state initialization.

Keywords: quantum computing; quantum machine learning; variational quantum circuits; quantum generative adversarial networks.

RESUMO

Algoritmos de preparação do estado quântico consistem em definir de uma sequência de operações unitárias para carregar um estado-alvo específico em um computador quântico. Podemos utilizar estes algoritmos em aplicações como Aprendizagem de Máquina Quântica. No entanto, alguns algoritmos para inicialização de estados quânticos têm uma complexidade de circuito exponencial com o número de qubits no sistema. É o caso dos algoritmos de codificação nas amplitudes, que é um tipo de codificação para carregar dados normalizados nas amplitudes de probabilidade do estado. Para contornar esta sobrecarga na complexidade, trabalhos exploram propriedades específicas dos estados quânticos para otimizar a complexidade do circuito, como a esparsidade ou a simetria. Outros trabalhos exploram a simplificação do circuito quântico para carregar um estado aproximado. É o caso das Redes Generativas Adversariais Quânticas, que utilizam uma arquitetura de circuito específica composta por blocos alternados de rotações de um qubit e portas controladas de emaranhamento de dois qubits. Porém, quando treinadas para carregar distribuições aleatórias, observamos que o desempenho se deteriora à medida que o número de qubits aumenta segundo a entropia relativa. Neste trabalho, propomos uma arquitetura diferente para os modelos generativos quânticos, baseada no algoritmo de preparação de estados conhecido como Low-Rank. E através de experimentos para carregar a distribuição log-normal, mostramos redução no erro da inicialização dos estados quânticos.

Palavras-chave: computação quântica; apredizagem de máquina quântica; circuitos quânticos variacionais; redes generativas adversariais quânticas.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BIT	Binary Digit
CNOT	Controlled-not
GAN	Generative Adversarial Network
NISQ	Noisy Intermediate-Scale Quantum
PQC	Parameterized Quantum Circuit
QGAN	Quantum Generative Adversarial Network
QSPA	Quantum State Preparation Algorithm
QUBIT	Quantum Binary Digit
RE	Relative Entropy
SVD	Singular Value Decomposition
VQA	Variational Quantum Algorithm
VQC	Variational Quantum Circuit

CONTENTS

1 INTRODUCTION

1.1 THE INPUT PROBLEM

In quantum computing, a Quantum State Preparation Algorithm (QSPA) consists of a sequence of unitary transformations $U := U_0 U_2 \cdots U_{n-1}$ acting on an n-qubit system in order to perform the mapping $U|\xi\rangle \mapsto |\psi\rangle$. For a target state $|\psi\rangle$ and some initial state $|\xi\rangle$, which in literature is usually $|0\rangle^{\otimes n}$ (MOTTONEN et al., 2004; GLEINIG; HOEFLER, 2021). We use those algorithms to load classical information into a quantum device. Information can be encoded the quantum states in different ways (SCHULD; PETRUCCIONE, 2018), as an example we will be focusing on basis encoding(TRUGENBERGER, 2001; VENTURA; MARTINEZ, 1999) and amplitude encoding(MOTTONEN et al., 2004; SHENDE; BULLOCK; MARKOV, 2005).

We can represent quantum states as complex unit vectors. And a quantum superposition can be mathematically described as a linear combination of basis states. Basis encoding consists of loading the data directly into the basis states of the systems in a uniform superposition. Consider a a dataset of m binary patterns with n bits $\mathcal{P} = \{x_0^n, x_1^n, \ldots, x_m^n\}$. A basis encoding procedure loads the dataset in a superposition as in $\frac{1}{\sqrt{m}} \sum_{i=0}^{m} |x_i^n\rangle$. It allows the loading of exponentially large superposition of binary patterns into a quantum state. Enabling quantum algorithms to simulate classical circuits' arithmetic (NIELSEN; CHUANG, 2000).

However, most of the data processing involves several bytes of information. And some real-life applications need the algorithms to have some level of reliability. And, despite the advancements, today's quantum devices are small and noisy. Implementing algorithms and methods in current devices requires attention to the devices' limitations (PRESKILL, 2018).

Amplitude encoding consists of loading the normalized data into the amplitudes of a quantum state. Thus, the i-th component of the superposition $\sum_i x_i |i\rangle$ corresponds to the i-th position in the normalized vector $x^T = [x_0, \ldots, x_{n-1}]$. This corresponds to an exponential gain in data representation because we need n qubits to represent 2^n amplitudes. Nevertheless, the works that explore quantum circuits for amplitude encoding (MOTTONEN et al., 2004; SHENDE; BULLOCK; MARKOV, 2005; BERGHOLM et al., 2005; PLESCH; BRUKNER, 2011) result in an exponential circuit complexity on the number of qubits for preparing an arbitrary state. Thus the complexity of a QSPA tends to dominate the overall complexity of the circuit. This bottleneck complexity is known as the input problem (BIAMONTE et al., 2017).

Some works propose QSPA optimized for states with specific properties to reduce the exponential cost of amplitude encoding, such as sparse quantum states (GLEINIG; HOEFLER, 2021; VERAS; SILVA; SILVA, 2022) or symmetric and highly entangled states (AKTAR et al., 2022; CRUZ et al., 2019; BÄRTSCHI; EIDENBENZ, 2019).

1.2 PREPARING APPROXIMATE QUANTUM STATES

The motivation for preparing an approximate state is that, by allowing some fidelity loss from reducing the quantum circuit's complexity, the error from noise for preparing the approximate quantum state $\left|\tilde{\psi}\right\rangle$ is smaller than the error from the noise for preparing the target state $|\psi_t\rangle$.

In (ARAUJO et al., 2023), the authors generalize a state preparation algorithm based on Schmidt decomposition introduced in (PLESCH; BRUKNER, 2011). The Schmidt decomposition of an arbitrary pure state $|\psi_{AB}\rangle$ of a composite system $\mathcal{H}_A \otimes \mathcal{H}_B$ is written as $|\psi_{AB}\rangle = \sum_{i=0}^{r} \lambda_i |\phi_i\rangle |\gamma_i\rangle$ with (λ_i), the i-th Schmidt number, with the number of non-zero Schmidt coefficients is called Schmidt rank $(r = |\{\lambda_i\}_{i=0}^{r}|)$. The method introduced in (ARAUJO et al., 2023) approximates $|\psi_{AB}\rangle$ by defining a quantum circuit that prepares an approximate state up to a desired Schmidt rank r_s with $r_s < r$. In this work, this approximation method will be called Low-Rank. And it will be used as the basis for our modification proposal.

Loading approximate states represents some advantages considering the limits of current devices. The Noisy Intermediate-Scale Quantum (NISQ) devices (PRESKILL, 2018) have noisy qubits and operations with low decoherence time, rendering the preparation of an arbitrary state impractical. Therefore, by allowing a fidelity loss, we can load the main properties of some target state $|\psi_t\rangle$ by preparing the approximate state $\left|\tilde{\psi}\right\rangle$.

Other works explore the use of Variational Quantum Algorithms (VQAs) to perform state approximation (NAKAJI et al., 2022; MARIN-SANCHEZ; GONZALEZ-CONDE; SANZ, 2021). They consist of constructing a small Parameterized Quantum Circuit (PQC), or ansatz. And by optimizing its parameters through different methods, the desired quantum state is approximated (CEREZO et al., 2021; BENEDETTI et al., 2019).s

For example, in (NAKAJI et al., 2022), the authors use auxiliary qubits to incorporate the signal of the functions they want to approximate on the amplitudes of the states. In (MARIN-SANCHEZ; GONZALEZ-CONDE; SANZ, 2021), the approximate function loading is generalized, where the authors define their approximation on a QSPA based on properties presented in (GROVER; RUDOLPH, 2002). The circuit's architecture is truncated according to some fidelity loss $\mathcal{L}_{fid} = 1 - \epsilon$ for some error ϵ and the process of VQA further optimizes the approximation.

1.3 APPROXIMATING DENSITY FUNCTIONS WITH GENERATIVE MODELS

A Generative Adversarial Network (GAN) is a machine learning method composed of a pair of machine learning models. Those are the generative network $\mathcal{G}$ for sampling new synthetic data, often known as the generator. And a discriminative network $\mathcal{D}$ for telling apart between fake and real data samples, also known as the discriminator. $\mathcal{G}$ is optimized to sample data that can deceive $\mathcal{D}$, and $\mathcal{D}$ is optimized to determine if a data sample is fake or real more accurately. (GOODFELLOW et al., 2020).

In a quantum computing context, Quantum Generative Adversarial Networks (QGANs), the generator, or discriminator, can be defined as quantum or classical models (LLOYD; WEEDBROOK, 2018). However, for approximating a density function, it is commonplace to fixate the definition of the generator as quantum and allow definition flexibility for the discriminator as either quantum or classical. We use QGAN for sampling data according to a density function, where the state loaded by the generator mimics the target density function.

Across the literature, some works explore using QGANs for approximating the data distribution. However, they rely on a specific architecture type for the PQC (SITU et al., 2020; ZENG et al., 2019; ZOUFAL; LUCCHI; WOERNER, 2019). In works as (ROMERO; ASPURU-GUZIK, 2021; NIU et al., 2022), the authors investigate using a quantum discriminator in the adversarial process. However, they need more qubits than a QGAN with a quantum generator and a classical discriminator, which, for a NISQ device, means introducing more noisy qubits and operations. Or require some post-processing for data sampling, as in (ROMERO; ASPURU-GUZIK, 2021).

1.4 OUR APPROXIMATION PROPOSAL

In our methodology, we focus on defining the PQC of the generator in the QGAN. As previously stated, some works rely on a specific architecture type to make the PQC. The architecture tipically consists of alternating blocks of single-qubit rotations and two-qubit entangling controlled operations (ZOUFAL; LUCCHI; WOERNER, 2019; CHEN et al., 2020). This architecture is known to be hardware-efficient (KANDALA et al., 2017). But we propose a PQC based on Low-Rank QSPA adaptive on the target Schmidt rank r_s and qubit register split into $m = \lfloor \frac{n+1}{q} \rfloor$ groups. For q, the maximum number of qubits per group. For the remainder of this work, the terms ansatz and PQC will be used interchangeably for referring to quantum circuits with optimizable parameters

1.5 MOTIVATION

The generative model presented in (ZOUFAL; LUCCHI; WOERNER, 2019) serves as the basis of our proposal. The authors trained the generative model for loading random distributions into the probabilities of quantum states. One of which was the log-normal $(\mu = 1, \sigma = 1)$ distribution. In this work, we will focus on the log-normal distribution because the authors in (ZOUFAL; LUCCHI; WOERNER, 2019) used it not just as a proof of concept but also in an application of Quantum Finance.

The performance of the QGAN in our basis reference uses the hardware-efficient architecture to define the ansatz. And its approximation quality seems to deteriorate as the number of qubits in the system grows in terms of Kullback-Leibler divergence. Authors in (ARAUJO et al., 2023; AGLIARDI; PRATI, 2022) also noted this limitation.

Using a QSPA to define an ansatz architecture is not a novel approach (MARIN-SANCHEZ; GONZALEZ-CONDE; SANZ, 2021). Nevertheless, we must remark that without the appropriate optimizations, the resulting circuit can be just as complex as a circuit for preparing an arbitrary quantum state. And methods such as the one previously mentioned, though resulting in good approximations, rely heavily on utilizing previous knowledge of the target distribution. That is impractical for applications of QGANs, where that knowledge is often unknown.

1.6 OBJECTIVES

We seek to define an adaptive ansatz based on Low-Rank QSPA(ARAUJO et al., 2023) to reduce the KL divergence, or relative entropy (RE), between the target distribution and the distribution loaded by the circuit. Given some desired Schmidt rank r_s, we build the ansatz up to a target rank r_s with optimizable parameters randomly initialized. And since we split the qubits of the system into $m = \lfloor \frac{n+1}{q} \rfloor$ groups, the architecture of the Low-Rank ansatz is applied to each of those groups separately in parallel, further simplifying the circuit's overall architecture.

1.6.1 Specific objectives

1. Improve the QGANs performance on systems of higher dimension, by using the Low-Rank ansatz on the generator without any previous knowledge of the target distribution.
2. Quantify improvements achieved using the Low-Rank ansatz on the QGAN for different qubit group splits against the existing method (ZOUFAL; LUCCHI; WOERNER, 2019) on numerical simulations.

1.7 CONTRIBUTIONS

The main contribution of this work is proposing an ansatz with an architecture defined by the QSPA Low-Rank adaptive on the target rank (r_s) and on the number of layers and qubit groups. In addition, we also improved the approximations generated by the QGAN using the Low-Rank ansatz.

Other contributions parallel to the one presented in this work were in (ARAUJO et al., 2023) published in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, which our main contributions were perfoming some simulations comparing the method introduced in the work with the approximation method using QGANs, and other minor adjustments. Another contribution was made in (VALE et al., 2023), which was submitted to an international journal, where we contributed to the implementation of the method, writing the text and by performing experiments comparing the method introduced the work with other existing procedures.

1.8 OUTLINE

We organized the remainder of this work in the following: Chapter 2 presents some theoretical background of quantum computing, generative adversarial networks, and VQAs. Chapter 3 introduces the QGAN for loading probability distributions on quantum states, as well as more details concerning its limitations. Chapter 4 presents the QSPA based on Schmidt decomposition and how we used it in our methodology. Chapter 5 discusses the experiment results and comparisons between the replicate QGAN and our method. And chapter 6 presents some concluding remarks on our results and discussions for possible future works.

2 THEORETICAL BACKGROUND

2.1 QUANTUM COMPUTING

Quantum computing consists of taking advantage of intrinsic properties in quantum physics by encoding information into quantum states and using it to perform computations. However, when explaining quantum computing, it is common practice in the literature to compare it with a simple example of how classical computers process and encode information.

The Binary Digit (BIT) is the smallest unit of information in the computer, which can assume two different states, either 1 or 0, exclusively. It is by combining several bits that a classical computer performs most of the complex computation (NIELSEN; CHUANG, 2000; MCMAHON, 2007; YANOFSKY; MANNUCCI; MANNUCCI, 2008).

As the classical bit, the smallest unit in a quantum computer is also composed of a two-state system called Quantum Binary Digit (QUBIT). However, unlike its classical counterpart, a qubit can assume the states 1, 0, or a superposition of both states. The last property is known as superposition.

We can mathematically represent the qubit using Dirac's notation $|\rangle$, with $|0\rangle$ as the quantum analog of the state 0 and $|1\rangle$ of the state 1. We can write the column vectors equivalents of the qubit states previously described as follows:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

We write the superposition of a single-qubit quantum state as a linear combination of the basis states. With each basis state multiplied but a scalar in $\mathbb{C}$, representing the probability amplitude of its corresponding state.

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

Meaning that, though the qubit is in both states at the same time, the outcome of the superposition is probabilistic because whenever we try to measure it, the superposition collapses to one of the states. With $|\alpha|^2$, the probability for the qubit to collapse to $|0\rangle$, and $|\beta|^2$ for it to collapse to $|1\rangle$. As expected from probability properties, they must sum up to one: $|\alpha|^2 + |\beta|^2 = 1$.

We can join several single qubit states to represent more complex states on quantum computers. The tensor product $|0\rangle \otimes |1\rangle$ represents this combination of several qubit states. However, works across the literature usually omit the symbol representing the tensor product, with multi-qubit systems written as $|0\rangle\,|1\rangle$ or $|01\rangle$. Thus, a superposition of an n-qubit system would be defined as a linear combination of the states, as shown in Eq. (2.1). With $\{0,1\}^n$ as the set of all possible binary strings of size n. And $|\alpha_x|^2$, the probability of measuring the x-th state.

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x \,|x\rangle \tag{2.1}$$

We use quantum operators, or gates, to manipulate the basis states and their corresponding amplitudes. They are represented mathematically as unitary matrices $U \in \mathbb{C}^{N\times N}$, with $N = 2^n$ in an n-qubit system. The most basic set of gates is called the Pauli gates, which consist of single-qubit operations.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}; Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

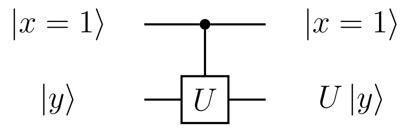
We can also define single-qubit operations by any parameterized unitary gates, which can be particularly useful for loading specific amplitudes or building new custom operations. The parameterized rotations mathematically represent unitary operators according to some angle θ.

$$R_x(\theta) = \begin{bmatrix} cos(\frac{\theta}{2}) & -isin(\frac{\theta}{2}) \\ -isin(\frac{\theta}{2}) & cos(\frac{\theta}{2}) \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} cos(\frac{\theta}{2}) & -sin(\frac{\theta}{2}) \\ sin(\frac{\theta}{2}) & cos(\frac{\theta}{2}) \end{bmatrix} \quad R_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}$$

In a multi-qubit system, single qubit operators can be applied to qubits in parallel, following the properties of the tensor product. Given two unitary gates A and B, and the state $|xy\rangle$, we employ those gates to the state as in $(A \otimes B)\,|xy\rangle = A\,|x\rangle \otimes B\,|y\rangle$. We can exploit this property for a state in superposition because we can perform operations in several states concurrently due to their linearity.

Controlled gates are specific in terms of manipulating states.These gate types use a set of one or more qubits as the control for applying an operation U on a target qubit.The control qubits determine the condition state to which the operation should apply an action to the target. Consider the state $|xy\rangle$. The controlled-U with the first state as control and the second as the target employs the operation U on state $|y\rangle$ if the basis state $|x\rangle$ is equal to some desired basis state $|s\rangle$. Fig. 1 displays an example of a controlled state $|x = 1\rangle$.

Figure 1 – Action of a controlled U when the control state x is 1

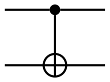


Source: The author (2023)

The controlled gate called controlled-NOT, or CNOT for short, consists of flipping the target qubit if the control qubit is on state $|1\rangle$. Its importance comes from the fact that one can decompose any unitary into a sequence of single-qubit gates and CNOTs (DIVINCENZO, 1995; LLOYD, 1995; DIVINCENZO, 1998). Having a universal gate set represents one of the criteria for building quantum computers (DIVINCENZO, 2000).

Figure 2 – Matrix and quantum circuit notation of a CNOT

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} =$$



Source: NIELSEN; CHUANG (2000)

One must apply measurement operations to the system to extract any information encoded in the quantum state. If the circuit loads a superposition of basis states, as shown in Equation (2.1), then its output is probabilistic. When measuring the quantum circuit's qubits, the probability of it returning a basis state y is $p(y) = |\alpha_{x=y}|^2$. Projective measurements (DJORDJEVIC, 2021) can describe the probability value $p(y)$. Given a measurement operator M_y on an arbitrary quantum state in superposition $|\psi\rangle$, the notation for probability in terms of M_y is as follows:

$$p(y) = \langle\psi|\, M_y^\dagger M_y \,|\psi\rangle \tag{2.2}$$

And the state post-measurement is defined as in Equation(2.3), where the factor $\frac{1}{\sqrt{\langle\psi|M_y^\dagger M_y|\psi\rangle}}$ normalizes the resulting state so that probability amplitudes may still sum up to 1. The probabilities relative to a specific state arise from the projective measurements $M_y = |y\rangle\,\langle y|$ for some $y \in \{0, 1\}^n$.

$$M_y |\psi\rangle = \frac{M_y |\psi\rangle}{\sqrt{\langle\psi| M_y^\dagger M_y |\psi\rangle}} \tag{2.3}$$

Observables P further generalize projective measurements. Provided such observables have a spectral decomposition $\hat{P} = \sum_y m_y P_y$, for some projection onto the eigenspace of P, $P_y = |e_y\rangle \langle e_y|$, where the elements $\{|e_y\rangle\}$ and $\{m_y\}$ represent its eigenvectors and their corresponding eigenvalues, respectively. Thus the probability for measuring a certain m_i is defined as follows:

$$p(m_y) = m_y \langle\psi| P_y |\psi\rangle \tag{2.4}$$

Since the circuit's output is probabilistic, it is common practice to perform several measurements for a better result estimate. Therefore, when measuring the results relative to a specific observable $\hat{P}$, one can compute its expectation value as in equation (2.5). This result will be employed in methods that perform circuit parameter optimization discussed in Section 2.2.

$$\langle\hat{P}\rangle = \sum_y m_y \langle\psi| \hat{P}_y |\psi\rangle \tag{2.5}$$

2.1.1 Quantum State Preparation Algorithms

A QSPA is an algorithm responsible for loading data into quantum states. This algorithm allows researchers to encode specific data. Two examples of information encoding strategies are basis and amplitude encoding when dealing with information encoding on the states. (SCHULD; PETRUCCIONE, 2018). In this work, we will focus on amplitude encoding algorithms.

Basis encoding is the act of encoding information on the basis states. For a system that uses qubits as basis states, this is analogous to encoding boolean logic into the states. However, a few adaptations are necessary because operations onto the quantum states must be reversible. For example, (TRUGENBERGER, 2001) introduces an algorithm that prepares a uniform superposition of a set of M binary patterns of size n.

$$|M\rangle = \frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} |x_i\rangle \tag{2.6}$$

Now let A be a unitary operator that encodes the function f, where $A\left|x\right\rangle\left|0\right\rangle = \left|x\right\rangle\left|f(x)\right\rangle$. The superposition in equation(2.6) Allows us to evaluate the function f on an exponentially large number of binary patterns in parallel.

$$A\left|M\right\rangle\left|0\right\rangle = \frac{1}{\sqrt{M}}\sum_{i=0}^{M-1}\left|x_i\right\rangle\left|f(x_i)\right\rangle \tag{2.7}$$

A drawback of this loading method is that applications need more qubits to encode the necessary data and operations. It is an overhead particularly disadvantageous for NISQ devices with low decoherence time and noisy qubits.

An amplitude encoding algorithm is responsible for loading the desired data into the probability amplitudes of the state. We must normalize classical data before the execution of the encoding algorithm. This type of encoding allows to load data into exponentially large spaces. That represents an advantage for applications in machine learning.

Amplitude encoding consists of setting the angles of parameterized gates so that the probability amplitudes of the quantum state resembles the desired data we want to load in the state. In the case of a single qubit system, we accomplish this by computing the angles of the operations that perform the following mapping:

$$U^{\dagger}\left(a_0\left|0\right\rangle + a_1\left|1\right\rangle\right) \mapsto \left|0\right\rangle \tag{2.8}$$

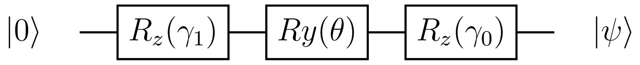
Where $U^{\dagger}$ represents a sequence of unitary operators and a_0, a_1 are the probability amplitudes in $\mathbb{C}$, where $|a_0|^2 + |a_1|^2 = 1$. By writing the probability amplitudes in their polar form, as in $\left|\psi\right\rangle = r_0 e^{i\phi_0}\left|0\right\rangle + r_1 e^{i\phi_1}\left|1\right\rangle$. It is possible to compute the angles of the rotations that map $\left|\psi\right\rangle \mapsto \left|0\right\rangle$. Since the sum of the square magnitude of each probability amplitude sum up to 1 we can rewrite $\left|\psi\right\rangle$ as in Equation(2.9).

$$\left|\psi\right\rangle = cos(\theta)e^{i\phi_0}\left|0\right\rangle + sin(\theta)e^{i\phi_1}\left|1\right\rangle \tag{2.9}$$

By making $-\gamma_0 = \phi_1 - \phi_0$ and $-\gamma_1 = -\phi_0 - \phi_1$. The sequence of rotations is given in Equation(2.10). Where $U^{\dagger} = R_z(-\gamma_1)Ry(-\theta)R_z(-\gamma_0)$. Thus, to prepare the state $\left|\psi\right\rangle$, one would need to apply U. Figure 3 displays the gate sequence for preparing $\left|\psi\right\rangle$ in quantum circuit notation.

$$R_z(-\gamma_1)Ry(-\theta)R_z(-\gamma_0)\left|\psi\right\rangle = \left|0\right\rangle \tag{2.10}$$

Figure 3 – Gate sequence for preparing $|\psi\rangle$ which consists of the decomposition of a $SU(2)$ gate



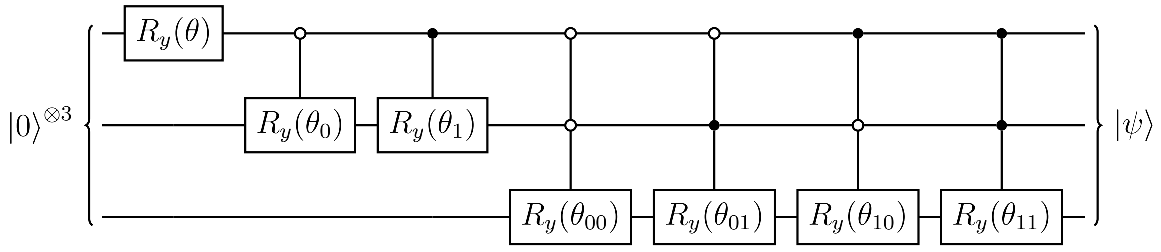
Source: NIELSEN; CHUANG (2000)

For an n-qubit system, the gate sequence for preparing an arbitrary state $|\psi_n\rangle$ must be more elaborate. That is because, when encoding a small set of data values into a superposition, we must ensure it does not affect previously encoded data values. Thus, the operation sequence that performs the mapping $U^\dagger |\psi_n\rangle \mapsto |0\rangle^{\otimes n}$ occurs in different stages. Each corresponds to the disentanglement of a qubit, as in $U_n^\dagger |\psi_n\rangle \mapsto \left|\hat{\psi}_{n-1}\right\rangle |0\rangle$. For $\left|\hat{\psi}_{n-1}\right\rangle$, an arbitrary $(n-1)$-qubit state, with 2^{n-1} amplitudes. And at m-th stage the remaining m qubits would already have been disentangled, resulting in the disentanglement of the m-th qubit.

$$U_m^\dagger \left|\hat{\psi}_{n-m+1}\right\rangle |0\rangle^{\otimes m-1} \mapsto \left|\hat{\psi}_{n-m}\right\rangle |0\rangle^{\otimes m} \tag{2.11}$$

In (MOTTONEN et al., 2004), the authors realize the gate sequence that performs the mapping in Equation (2.11) using uniformly controlled rotations. Given that we have $n-m+1$ qubits in superposition, we apply 2^{n-m+1}-controlled operations with controls over all 2^{n-m+1} possible basis states. And for each basis state used as control, we compute the angles for each rotation to map the m-th qubit to the target basis state (which, in this case, is $|0\rangle$). When the full gate sequence is defined, its inverse performs the state preparation $U|0\rangle^{\otimes n} \mapsto |\psi_n\rangle$ (Figure 4).

Figure 4 – State preparation of a 3-qubit system using uniformly controlled R_y rotations



Source: MOTTONEN et al. (2004)

One disadvantage of this method is that each multi-controlled adds up to the overall complexity of the circuit in terms of depth and CNOT count. In (SHENDE; BULLOCK; MARKOV, 2005), the authors further optimize the construction of the QSPA. The angles of each stage are computed similarly but are applied to the circuit using quantum multiplexers. The method using multiplexers is further generalized by (BERGHOLM et al., 2005), where the authors multiplex single-qubit U gates on each stage. And by incorporating phases into the multiplexed gates, the quantum circuit complexity for state preparation is reduced by half.

2.2 VARIATIONAL QUANTUM ALGORITHMS

Variational Quantum Algorithms (VQAs) (CEREZO et al., 2021) are a class of quantum algorithms that use a hybrid quantum-classical approach for solving specific problems. Using a low-depth parameterized quantum circuit to initialize a trial state, $|\psi(\Theta)\rangle$, with the parameters Θ. Given a target state $\left|\hat{\psi}\right\rangle$, One can approximate the output trial state by optimizing the parameters Θ iteratively using classical methods analogous to a neural network.

VQAs solutions are promising because they involve using low-depth quantum circuits, offloading the parameter processing and optimization to the classical computer. Current devices are still noisy and of small scale if compared to what would be required to compose a fully realized error-corrected quantum computer. Thus, researchers must consider noisy qubits, operations, and low decoherence times when proposing new solutions to real-life applications.

Across the literature, this class of algorithms usually involves three key components: The ***Ansatz***, ***Loss function*** and the ***Optimization method*** (CEREZO et al., 2021; PRIETO, 2022; MITARAI et al., 2018).

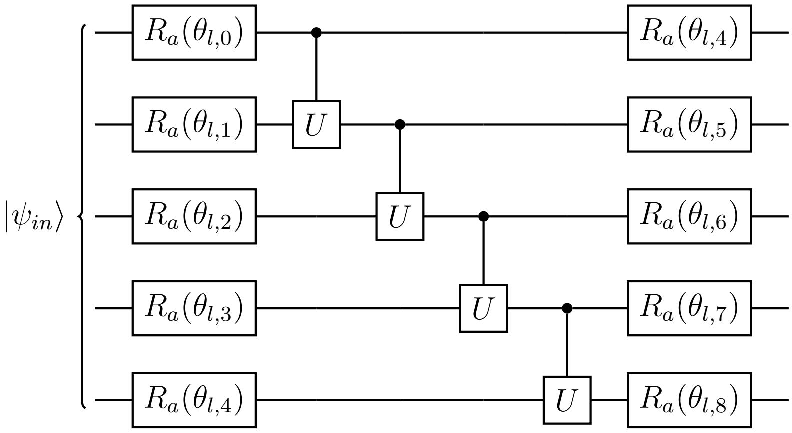
2.2.1 Ansatz

The ansatz refers to the quantum circuit's architecture. That is, the sequence of controlled and parameterized operations and on which qubit they must act. Some works also refer to it as Variational Quantum Circuit (VQC) or Parameterized Quantum Circuit (PQC). We optimize the parameters from the quantum circuit to approximate the desired result.

The gate configuration of the ansatz and its initial parameters depends on the application and the device. (CEREZO et al., 2021). Across the literature, it is common to find works using a sequence of rotation gates, often called rotation blocks, intertwined with a series of controlled entangling operations called the entanglement block. (AGLIARDI; PRATI, 2022; CHEN et al., 2020; ROMERO; ASPURU-GUZIK, 2021; DUAN; HSIEH, 2022; SITU et al., 2020). In Figure 5, we show an example of this architecture.

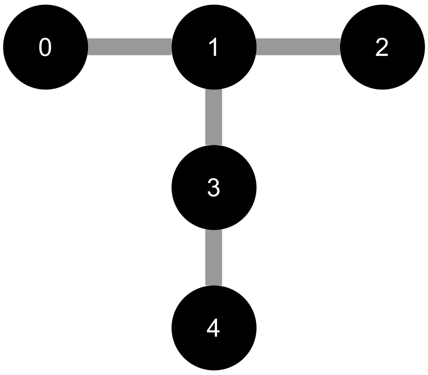
The architecture presented in Figure 5 is said to be hardware efficient (MITARAI et al.,

Figure 5 – Hardware-efficient ansatz architecture for a 5-qubit system. Where a refers to the rotation axis $a \in \{x, y, z\}$. And U can be any parameterized or non parameterized controlled unitary. This is one example of the many possible models.



Source: The author (2023)

Figure 6 – Example of qubit connectivity from the device *ibmq_belem*



Source: IBM (2023)

2018; KANDALA et al., 2017). That is the case because, on a physical device, qubits follow a specific connectivity (as an example, see Figure 6). Thus, each qubit has a certain number of neighboring qubits over which one can perform controlled operations without applying quantum swaps to bring them closer together. Therefore, to use a controlled gate on non-neighboring qubits, we must add some swap gates so the quantum operation may act on the system as expected. Thus, we achieve hardware efficiency by changing the entanglement block layout to match the device's qubit connectivity.

2.2.2 Loss function

Also known as the objective function, error function, cost function, or criterion, it determines the performance quality of the machine learning model (GOODFELLOW; BENGIO; COURVILLE, 2016). We use this function type in gradient-based optimization of parameters for some function $f(x, \Theta)$, which represents the output of model f on point x with the parameter set Θ.

The definition of the loss function $\mathcal{L}$ heavily depends on the problem the researcher is trying to solve. For example, in a supervised machine learning problem, the loss function is defined according to a given desired label y. We use a quantum circuit as a machine learning model with circuit measurements defining the output labels (NEUMANN; PHILLIPSON; VERSLUIS, 2019; SCHULD et al., 2020).

2.2.3 Optimization method

When referring to optimization methods, we refer to methods for updating the VQC parameters to minimize $\mathcal{L}$. For applications involving VQA, the optimization methods can be either gradient-based or gradient-free (CEREZO et al., 2021). The genetic algorithms are the methods that derive gradient-free approaches. However, they tend to be costly as the number of parameters or qubits increases (PRIETO, 2022). In contrast, gradient-based optimization optimizes quantum circuit parameters according to the gradient vector (CEREZO et al., 2021; PRIETO, 2022; MITARAI et al., 2018). In this work, we shall focus on VQAs using gradient-based optimization.

In classical machine learning, optimizing the model to improve output quality would be analog to optimize its parameters by minimizing the loss function $\mathcal{L}$. We must compute the gradient of $\mathcal{L}$ to determine how much the parameters are updated. And, by calculating the difference between the parameter vector and its gradient according to x, we obtain a new θ_j closer to a minimal point of $\mathcal{L}$. To prevent parameter updates θ_{j+1} from beyond the minima, we multiply the gradient by a small factor η. Often referred to as *learning rate*. The scalar η defines how much the parameters are updated. Equation (2.12) exemplifies a parameter update. This process is known as gradient descent. (GOODFELLOW; BENGIO; COURVILLE, 2016; BISHOP; NASRABADI, 2006).

$$\theta_{j+1} = \theta_j - \eta \frac{\partial}{\partial \theta_j} \mathcal{L}(y, f(x, \theta_j)) \tag{2.12}$$

In deep neural networks, we use the gradient derived from the loss function to update the parameters of the intermediate layers. This process is known as the error back-propagation (GOODFELLOW; BENGIO; COURVILLE, 2016; BISHOP et al., 1995). One cannot access the intermediate states of a sequence of unitary operations in a quantum circuit without affecting the circuit's output.

One can infer the circuit's output by computing the quantum circuit's expectation value. Due to the intrinsic properties of quantum mechanics, the circuit's output is probabilistic. However, by querying the quantum circuit several times and performing measurements according to some observable $\hat{K}$, it is possible to compute its expectation value $\langle \hat{K} \rangle$. Thus, for a given parameter θ_j, one can analytically infer its gradient by shifts of $\theta_j \pm \epsilon$, for some shift angle ϵ. In the case of Pauli observables, we define the shift angle as $\epsilon = \frac{\pi}{2}$ (MITARAI et al., 2018; BENEDETTI et al., 2019; SCHULD et al., 2019).

$$\frac{\partial \langle \hat{K} \rangle}{\partial \theta_j} = \frac{\langle \hat{K} \rangle_{\theta_j + \epsilon} - \langle \hat{K} \rangle_{\theta_j - \epsilon}}{2} \tag{2.13}$$

Now that we defined how we compute the gradient, one can use the traditional updating methods traditionally used in classical machine learning to update the circuit's parameters. In this work, we shall use ADAM (KINGMA; BA, 2014) because the authors (ZOUFAL; LUCCHI; WOERNER, 2019) used it for optimizing QGANs.

2.3 GENERATIVE ADVERSARIAL NETWORKS

Generative adversarial networks, or *GANs*, are a type of machine learning model consisting of a pair of deep neural networks. The neural networks composing this model are the generator, denoted as G, and the discriminator, represented as the letter D. The role of the discriminator is to distinguish the input as real or fake data created by the generator. And the generator generates data samples that are capable of deceiving the discriminator.

Thus, both networks are optimized competitively. Let $x_G = G(z)$ be the fake samples created by the generator. With z, some random noise. And let $P = \{x_1, \ldots, x_p\}$, the dataset batch with p samples, where the fake samples x_g have the same dimension as the real ones.

And the output of the discriminator $y = D(x_i)$ is within $\{0, 1\}$, indicating if the input received is fake or real. For $y_g = D(G(z))$, the discriminator's labeling generated data as being fake.

We train the discriminator to maximize its performance towards better discrimination between real or fake samples. Equation (2.14) shows the loss function for the discriminator. We can see that the discriminator is optimized by minimizing the loss as $y^g \to 1$. And maximizing as $y \to 0$.

$$\mathcal{L}_D = \frac{1}{p}\sum_{i=1}^{p}[log(y_i) + log(1 - y_i^g)] \tag{2.14}$$

We train the generator to minimize the performance of the discriminator for distinguishing the generated data as fake.

$$\mathcal{L}_G = -\frac{1}{p}\sum_{i=1}^{p}\log(y_i^g) \tag{2.15}$$

This type of optimization enables the generator to sample synthetic data gradually closer to samples from the dataset, thus learning the intrinsic distribution of the data. (GOODFELLOW et al., 2020).

3 QUANTUM GENERATIVE ADVERSARIAL NETWORKS

We can consider different configurations when building the models for exploring the potential of quantum computing in adversarial learning. In (LLOYD; WEEDBROOK, 2018), the authors present some discussions for cases where either the data from which the underlying distribution defines the samples or where the generative or discriminative network can be either quantum or classical. They further state that, for quantum data, a quantum generative network is better suited for the problem.

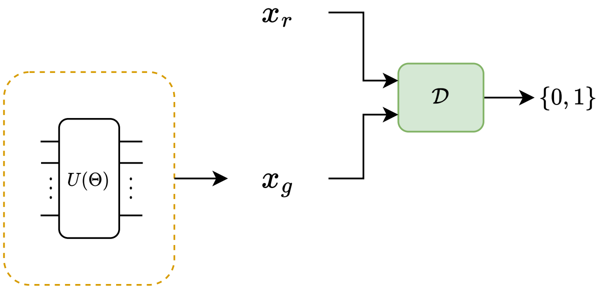
However, this work will focus on quantum generative networks for sampling classical data with a classical discriminator dedicated to distinguishing classical fake samples from real ones. The generative network is a variational quantum circuit trained to load the dataset's underlying distribution into the quantum state.

3.1 DISTRIBUTION LOADING

QGANs, as its classical counterpart, can be used as a model for sampling synthetic data from the underlying distribution of a dataset. Works as in (SITU et al., 2020; ROMERO; ASPURU-GUZIK, 2021; ZENG et al., 2019) present a hybrid quantum-classical framework. The authors use VQC for classical data sampling, where the expected value of the measurements samples the data. And the data sampling is possible because the circuit's parameters are optimized so that the probability amplitudes of the state it produces approximate the data's distribution.

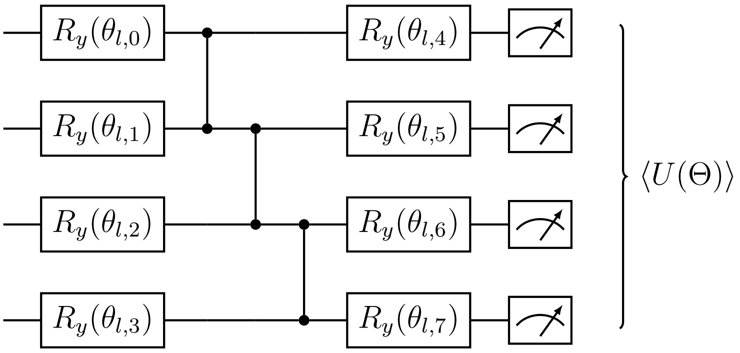
In (ZOUFAL; LUCCHI; WOERNER, 2019), the authors present the same hybrid quantum-classical framework, using a quantum generator and a classical discriminator, training the quantum circuits for loading the data's distribution. The architecture of the VQC used by

Figure 7 – Hybrid quantum-classical framework for quantum generative adversarial learning. Where x_r is a real data sample, and x_g a generated sample. And $\mathcal{D}$ is a classical discriminator.



Source: The author (2023)

Figure 8 – Architecture of VQC proposed by (ZOUFAL; LUCCHI; WOERNER, 2019), for a 4-qubit system. Where l refers to the layer index.



Source: ZOUFAL; LUCCHI; WOERNER (2019)

the authors consists of a sequence o rotation gates alternated with blocks of controlled gates (Figure 8). We will refer to those blocks as rotation and entanglement.

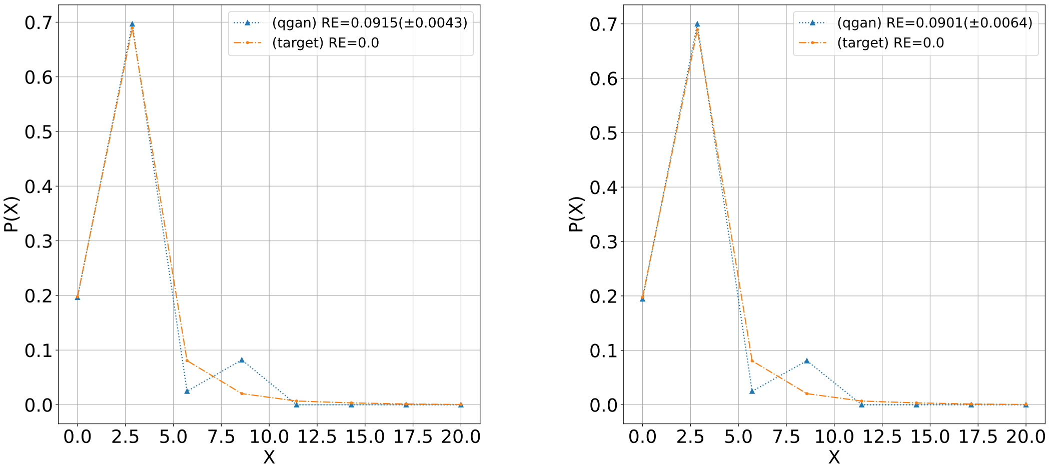
In (ZOUFAL; LUCCHI; WOERNER, 2019), the authors derived results from the numerical simulations to show the potential of training QGAN to load a random distribution. For a dataset of $20,000$ randomly generated uni-dimensional samples following a specific probability distribution $p(x)$. The authors optimized the generative network so that the frequencies produced by the measurements of the qubits would approximate $p(x)$.

The QGAN implemented in (ZOUFAL; LUCCHI; WOERNER, 2019) the authors trained to sample integers within the range $[0, 2^n - 1]$, for n as the number of qubits in the system. The discriminative network tries to distinguish the frequencies returned by the quantum circuit against frequencies inferred from a batch with 2000 samples. Thus, the VQC updates the parameters according to the loss presented in 2.3. The authors performed numerical simulations of a QGAN for loading different distributions. Those were log-normal, triangular, and bimodal.

In this chapter, we will focus on the log-normal distribution $(\mu = 1, \sigma = 1)$ because the authors in (ZOUFAL; LUCCHI; WOERNER, 2019) also use it in a real-life application for Quantum Finance. And In (ARAUJO et al., 2023), the authors could reuse the implementation introduced in (ZOUFAL; LUCCHI; WOERNER, 2019) for sampling points within the interval $x_i \in [0, \dots, 20]$ where every x_i is equidistant.

We were able to replicate the experiments loading the log-normal distribution $(\mu = 1, \sigma = 1)$ in (ZOUFAL; LUCCHI; WOERNER, 2019), by training the QGAN. The QGAN was trained to mimic frequencies of the discretized log-normal distribution on 2^n equidistant points in the interval $[0, 20]$ using two different initialization types. For all initialization types, the initial state was in $|0\rangle^{\otimes n}$, with n representing the number total of qubits on the system. Each initialization

Figure 9 – Comparison of the mean frequencies infered by the state vector from the 10 executions returned by the QGAN's VQC in the log-normal distribution $(\mu = 1, \sigma = 1)$ discretized in 2^3 points in $[0, \ldots, 20]$.


(a) *Random* initialization

(b) *Uniform* initialization

Source: The author (2023)

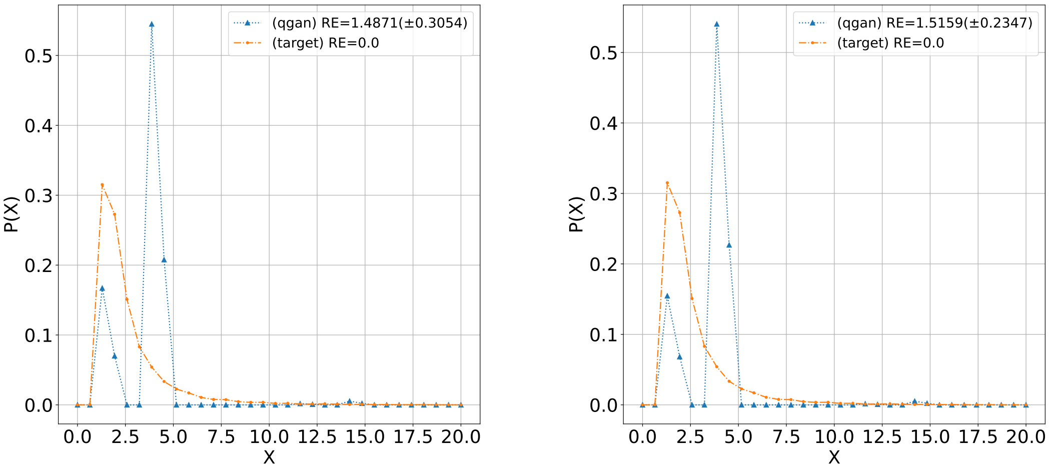
type was set to prepare the input state for the VQC and they shall be referred to as *uniform* and *random* initialization for the remainder of this work.

For the *random* initialization, the input state of the ansatz remained as $|0\rangle^{\otimes n}$ with randomly initialized parameters following a uniform distribution in $[-\pi, \pi]$. To prepare the input state using *uniform* initialization, besides parameters random initialization, we applied Hadamard gates to each qubit $H^{\otimes n}$. To determine how similar are the fake and real frequencies we used KL-divergence, also referred to as Relative Entropy (RE).

$$D(p||g) = \sum_{x_i} p(x_i) log \left(\frac{p(x_i)}{g(x_i)} \right) \tag{3.1}$$

We inferred the frequencies displayed in Figure 9 from the state vector obtained by the numerical simulations, then we computed the mean of the frequencies from 10 executions. We defined the generative model as VQC trained to load the log-normal distribution $(\mu = 1, \sigma = 1)$ over $1,000$ training epochs. We created the dataset with $20,000$ randomly generated uni-dimensional samples in the interval $[0, 20]$, with each training epoch using a batch size of $2,000$ samples. The standard deviation of the frequencies between executions was below order of 10^{-4}, so we will not display them in the figures.

Figure 10 – Comparison of the mean frequencies from 10 executions returned by the QGAN's VQC in the log-normal distribution $(\mu = 1, \sigma = 1)$ discretized in 2^5 points in $[0, \ldots, 20]$.

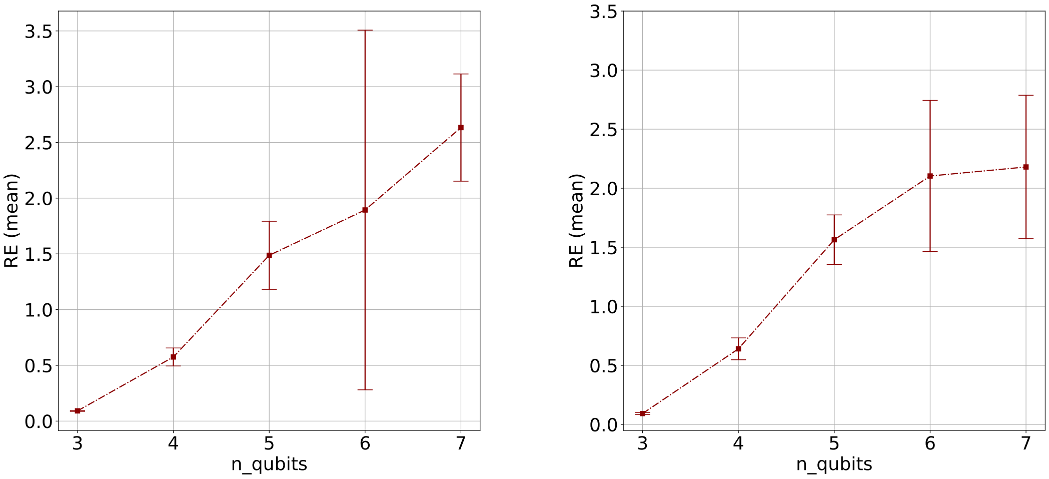


(a) *Random* initialization

(b) *Uniform* initialization

Source: The author (2023)

Figure 11 – Mean relative entropies and errors over all 10 executions, for quantum systems with qubits ranging from 3 to 7 qubits.



(a) *Random* initialization

(b) *Uniform* initialization

Source: The author (2023)

3.2 LIMITATIONS

Indeed, using a shallow VQC in a 3-qubit system, one might be able to approximate a random distribution, as in the case of the log-normal distribution. However, by replicating the experiments with the same configurations on systems with a higher number of qubits, we observed the performance of the QGAN's VQC deteriorates as the number of qubits on the system increase. Figure 10 illustrates the mean frequencies generated by a VQC on a 5-qubit system.

By varying the number of qubits within the range $[3, 7]$, we could compute the mean relative entropies over all 10 executions. The obtained results further evidence the QGAN limitation concerning the number of qubits. We displayed the estimated mean relative entropies and the errors in Figure 11. Works as in (AGLIARDI; PRATI, 2022; ARAUJO et al., 2023) also observed this limitation.

The results acquired from replicating (ZOUFAL; LUCCHI; WOERNER, 2019) and its limitations served as the basis reference and inspiration for our modification proposal.

4 METHODOLOGY

In this Section, we introduce the QSPAs used as the base of our methodology. And we further discuss what elements of those algorithms we used to define the structure of the ansatz.

4.1 STATE PREPARATION BASED ON SCHMIDT DECOMPOSITION

In (PLESCH; BRUKNER, 2011), the authors propose a state preparation algorithm that exploits the bi-partitioning of a quantum state. They introduced bi-partitioning to optimize the quantum circuit that loads the quantum state into the system. They based the algorithm on the Schmidt decomposition of a quantum state. And in this Section, we will detail its most relevant steps.

The Schmidt decomposition is a restatement of the Singular Value Decomposition (SVD) applied to tensors in Hilbert spaces (NIELSEN; CHUANG, 2000). Consider a given quantum state $|\psi_{AB}\rangle$ in a composite system $\mathcal{H}_A \otimes \mathcal{H}_B$. We can rewrite the system in Equation (4.1). This expression is called the Schmidt decomposition of state $|\psi_{AB}\rangle$. For $\{|\phi_i\rangle\}_{i=0}^{2^m} \in \mathcal{H}_A$ and $\{|\gamma_i\rangle\}_{i=0}^{2^{n-m}} \in \mathcal{H}_B$, with $m = \lfloor \frac{n}{2} \rfloor$ and the values $\{\lambda_i\}$ correspond to the Schmidt coefficients, with $\sum_i |\lambda_i|^2 = 1$.

$$|\psi_{AB}\rangle = \sum_{i=0}^{r} \lambda_i |\phi_i\rangle |\gamma_i\rangle \tag{4.1}$$

The number of non-zero Schmidt coefficients represents the Schmidt rank $r = |\{\lambda_i\}_{i=0}^{r}|$. Each of these coefficients corresponds to a singular value in the SVD of the system $UDV^\dagger$. Let $\{|\alpha_j\rangle\}$ and $\{|\beta_k\rangle\}$, two sets of any fixed orthonormal basis for $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively, then we can rewrite the state $|\psi_{AB}\rangle$ as in Equation (4.2) with $\lambda_i = d_{ii}$ corresponding the entries of the diagonal matrix D, known as singular values. And for u_{ji} and v_{ik} corresponding to the entries in the matrices U and $V^\dagger$, respectively.

$$|\psi_{AB}\rangle = \sum_{j,i,k} u_{ji} d_{ii} v_{ik} |\alpha_j\rangle |\beta_k\rangle \tag{4.2}$$

The algorithm starts by computing the Schmidt decomposition of a desired state $|psi_{AB}\rangle$. For simplicity, let us consider $|\psi_{AB}\rangle$ as a 4-qubit state we wish to encode in a superposition as in Equation (4.3). For a complex value c_x, the probability amplitude of the basis state $|x\rangle$.

$$\begin{aligned}\left|\hat{\psi}_{AB}\right\rangle = c_{0000}\left|0000\right\rangle + c_{0001}\left|0001\right\rangle + c_{0010}\left|0010\right\rangle + c_{0011}\left|0011\right\rangle +\\ c_{0100}\left|0100\right\rangle + c_{0101}\left|0101\right\rangle + c_{0110}\left|0110\right\rangle + c_{0111}\left|0111\right\rangle +\\ c_{1000}\left|1000\right\rangle + c_{1001}\left|1001\right\rangle + c_{1010}\left|1010\right\rangle + c_{1011}\left|1011\right\rangle +\\ c_{1100}\left|1000\right\rangle + c_{1101}\left|1101\right\rangle + c_{1110}\left|1110\right\rangle + c_{1111}\left|1111\right\rangle\end{aligned} \tag{4.3}$$

In textbooks, SVD is a method for factoring matrices. Nevertheless, to perform Schmidt decomposition of the state $\left|\hat{\psi}_{AB}\right\rangle$, we must define how we wish to partition the state. We accomplish state partitioning by creating a partition matrix from which the rows and columns map to different sets of qubits. Let us label each qubit as $|wxyz\rangle$, with wx as the basis states of the two most significant qubits and yz labeling the two least-significant qubits. Thus, we can define the partition matrix as in Equation (4.4).

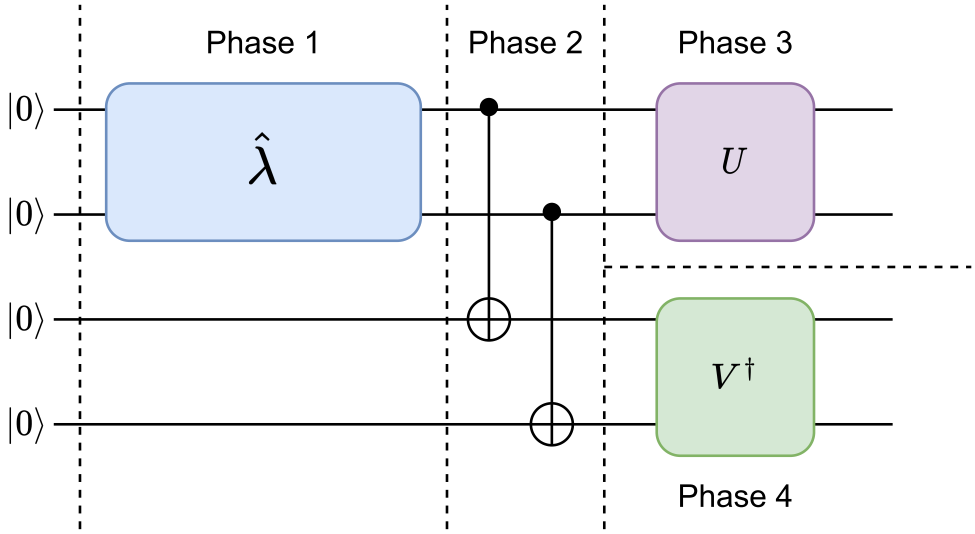
$$\begin{array}{c|c|c|c|c|c} & \multicolumn{5}{c}{|yz\rangle} \\ \hline & & |00\rangle & |01\rangle & |10\rangle & |11\rangle \\ \hline & |00\rangle & c_{0000} & c_{0001} & c_{0010} & c_{0011} \\ |wx\rangle & |01\rangle & c_{0100} & c_{0101} & c_{0110} & c_{0111} \\ & |10\rangle & c_{1000} & c_{1001} & c_{1010} & c_{1011} \\ & |11\rangle & c_{1100} & c_{1101} & c_{1110} & c_{1111} \end{array} \tag{4.4}$$

With the partition matrix defined, we must now compute its SVD, thus obtaining the Schmidt coefficients $\{\lambda_i\}$ and the unitary operators U and $V^{\dagger}$. Afterwards, we begin the construction of the quantum circuit for encoding $|\psi_{AB}\rangle$. As in any QSPA, assuming the initial state is $|0\rangle^{\otimes n}$, the main goal is to compute a sequence of unitary transformations U that perform the mapping $U|0\rangle^{\otimes n} \mapsto |\psi_{AB}\rangle$. The method proposed by (PLESCH; BRUKNER, 2011) accomplishes the preparation of $|\psi_{AB}\rangle$ by dividing it into four phases, as shown in Figure 12.

For more details on how we decompose the operators $\hat{\lambda}, U$ and V, in Figure 12, into single qubit gates and CNOTs, please refer to (SHENDE; BULLOCK; MARKOV, 2005; BERGHOLM et al., 2005). On phase 1, the operator $\hat{\lambda}$ prepares the Schmidt coefficients in the quantum state, resulting in the following superposition:

$$(\hat{\lambda} \otimes I^{\otimes 2})\left|00\right\rangle\left|00\right\rangle = \lambda_{00}\left|00\right\rangle\left|00\right\rangle + \lambda_{01}\left|01\right\rangle\left|00\right\rangle + \lambda_{10}\left|10\right\rangle\left|00\right\rangle + \lambda_{11}\left|11\right\rangle\left|00\right\rangle \tag{4.5}$$

Figure 12 – Example of a 4-qubit quantum circuit defined to load $|\psi_{AB}\rangle$ using the QSPA proposed by (PLESCH; BRUKNER, 2011)



Source: The author (2023)

In phase 2, the quantum circuit applies a sequence of CNOTs on the system, using the qubits of the first register as controls, thus "copying" the basis states to the second register. The resulting superposition after we apply the CNOTs to the system is as follows:

$$\lambda_{00} |00\rangle |00\rangle + \lambda_{01} |01\rangle |01\rangle + \lambda_{10} |10\rangle |10\rangle + \lambda_{11} |11\rangle |11\rangle \tag{4.6}$$

Phases 3 and 4 occur in parallel, and the operators U and $V^\dagger$ are responsible for mapping each basis state to their corresponding quantum state in $\{|\phi_i\rangle\}$ or $\{|\gamma_i\rangle\}$, respectively, for $i \in \{0,1\}^2$. The resulting superposition after phases 3 and 4 is as follows:

$$|\psi_{AB}\rangle = \sum_{i \in \{0,1\}^2} \lambda_i |\phi_i\rangle |\gamma_i\rangle \tag{4.7}$$

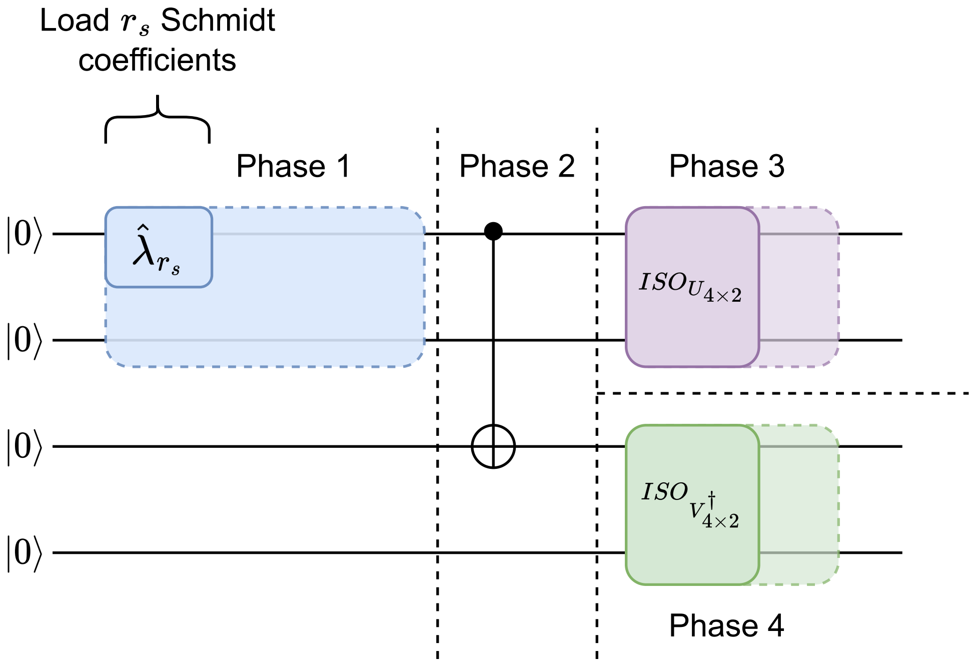
Though we have chosen specific groups of qubits for partitioning the desired state, one can apply this method with other qubit partitionings.

4.2 LOW-RANK

In (ARAUJO et al., 2023), the authors proposed a QSPA that further generalized the algorithm introduced in (PLESCH; BRUKNER, 2011). They call their method Low-Rank. And it is used as a basis for our method. The Low-Rank QSPA can work as in (PLESCH; BRUKNER, 2011) by first loading all coefficients, but it can also prepare an approximate state, starting by partially loading the Schimidt Coefficients. The approximation is derived from the low-rank approximation of matrices using SVD. Thus, allowing the definition of a quantum circuit that loads a quantum state approximate to $|\psi_{AB}\rangle$ up to a desired Schmidt rank r_s, with $r_s < r$.

Given an arbitrary quantum state $|\psi_{AB}\rangle$ with a Schmidt Rank $r = |\{\lambda_i\}|$. A quantum circuit construction for a Low-Rank approximation of $|\psi_{AB}\rangle$ would load the first r_s Schmidt Coefficients of the state partitioning $|\psi_{AB}\rangle = \sum_i^{r_s} \lambda_i |\phi_i\rangle |\gamma_i\rangle$, since the coefficients are sorted from greatest to smallest, choosing the first r_s will lead to the best possible approximation (ECKART; YOUNG, 1936). Leaving us with an approximation of rank $r_s = |\{\lambda_i\}_{i=0}^{r_s}|$, where we simplify the operators U and $V^\dagger$ to isometries. The overall structure of the circuit is exemplified in Figure 13.

Figure 13 – Quantum Circuit for a Low-Rank approximation on a 4-qubit system. With $r_s = 2$.



Source: The author (2023)

Therefore, by using the Low-Rank approximation of a state, we can reduce the quantum circuit's complexity relative to the operation count. Nevertheless, the approximated QSPA offers benefits on current devices. NISQ devices have noisy qubits and operations with low decoherence time, thus making the exact loading arbitrary state impractical. Therefore, by allowing a fidelity loss $\mathcal{L}_{\text{fid}} = 1 - |\langle\psi_t|\psi_o\rangle|^2$, for some target state $|\psi_t\rangle$ and some output state $|\psi_o\rangle$, we can load the main properties of $|\psi_t\rangle$ by reducing the number of noisy operations. This is done so that the error for preparing the approximate state, on a real device, is less than preparing the exact state.

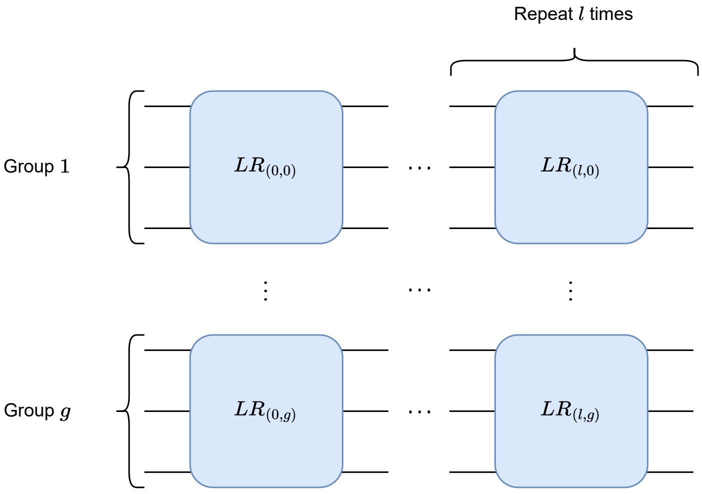
4.3 ADAPTIVE ANSATZ

We based the adaptive ansatz of our proposed method on the same construction presented in (ARAUJO et al., 2023). However, instead of computing the circuit's operations and parameters according to a target state $|\psi_t\rangle$ with a desired rank r_s, it is possible define the quantum circuit for a desired rank r_s with optimizable randomly initialized parameters. And we use parameters that are optimizable through the process of VQA. That is, we train the circuit to approximate $|\psi_t\rangle$.

Because we do not use previous knowledge of $|\psi_t\rangle$ to define the circuit for a desired rank r_s, we must infer the dimensions of operations such as U and $V^\dagger$. Given the partitions size $m = \lfloor \frac{n}{2} \rfloor$ and $\bar{n} = n - m$, for n, the total number of qubits. Then, $dim(U) = 2^{\bar{n}} \times 2^{\bar{n}}$ and $dim(V^\dagger) = 2^m \times 2^m$ define the dimensions of U and $V^\dagger$, respectively. And the effective rank is inferred from the previously computed dimensions $r = min(2^{\bar{n}}, 2^m)$.

We also incorporated the repetition of the circuit through a configurable number of layers. It is a property commonly employed by some works that use VQA (ZOUFAL; LUCCHI; WOERNER, 2019; AGLIARDI; PRATI, 2022; PRIETO, 2022). Besides building an ansatz using a configurable number of layers, we included the feature of qubit grouping. That is, grouping a subsets of qubits in different registers and defining the Low-Rank ansatz independently in each of those groups, in parallel. We added this feature to further simplify the resulting quantum circuit. Each ansatz per-group and each layer repetition was assigned with randomly initialized parameters independently. Figure 14 details the overall circuit structure.

Figure 14 – Qubit grouping and layering of the Low-Rank ansatz



Source: The author (2023)

5 EXPERIMENTS AND RESULTS

In our experiment configuration, we performed numerical simulations using the QGAN with generative network $\mathcal{G}$ defined as a VQC using our adaptive ansatz introduced in Chapter 4. We defined the discriminative network $\mathcal{D}$ as a simple multilayer perceptron using the PyTorch library (PASZKE et al., 2017). Our main goal was to train $\mathcal{G}$ to load a target distribution P, discretized on the interval of equidistant data points in $[0, 20]$, so that every data point sampled as $x_i \in [0, 20]$ follows the distribution approximated by $\mathcal{G}$.

For our experiments, we chosed to focus on the log-normal ($\mu = 1, \sigma = 1$) distribution because it is the distribution used by (ZOUFAL; LUCCHI; WOERNER, 2019) as proof of concept and as an application of Quantum Finance. To determine the quality of the approximation of $\mathcal{G}$, we used Relative Entropy (RE), presented in Equation (3.1), where the output frequencies were inferred from the output state vector of the distribution loaded by $\mathcal{G}$ for each point $x_i \in [0, 20]$. We also compared the QGAN frequencies with the discretized log-normal distribution using RE on the same data points. And we opted to focus on the *Random* and *Uniform* initialization methods, which are the same initialization methods used in Section 3.1.

To create the dataset, we sampled $20,000$ points within interval $[0, 20]$, following the log-normal ($\mu = 1, \sigma = 1$) distribution. We trained QGAN with $1,000$ epochs and a batch size of $2,000$ samples. Each single execution had its duration ranging from 30 minutes to more than one hour, depending on number of qubits on the system. We used ADAM (KINGMA; BA, 2014) optimization method with the same hyper-parameters defined to optimize the replicate in Section 3.1.

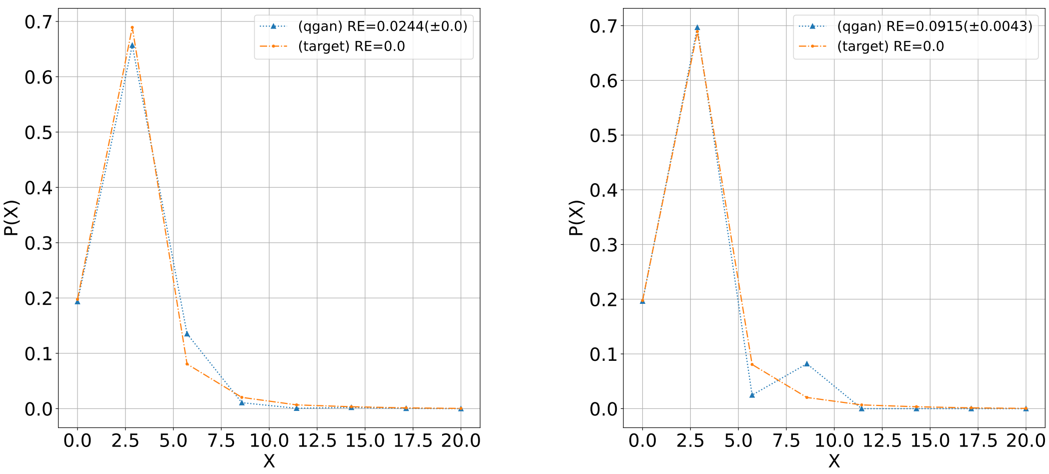
We varied the qubit number of the circuit within the range $[3, 7]$ as in the replicate of Section 3.2. And with the option of defining ansatzes with subgroups of qubits (see Figure 14), we also varied the number of qubit groups within the range $[2, q]$ for $q = \lfloor \frac{n+1}{2} \rfloor$ and a single layer for both Replicate QGAN and QGAN using Low-Rank defined ansatz. For the remainder of this chapter, the qubit grouping shall be called *build qubits*. And the QGAN using a Low-Rank ansatz shall be referred to as Low-Rank QGAN.

With those configurations, we ran 10 executions to perform statistical tests on the QGANs outputs. The numerical simulations for the quantum circuit implementations were carried out using Qiskit (ALEKSANDROWICZ et al., 2019) and did not target any specific device's coupling map. All the results were computed from noiseless simulations. Due to time limitations we did not run experiments on Low-Rank QGAN with ansatzes defined with more than one layer and different ranks.

5.1 EXPERIMENTS FOR 3 QUBIT SYSTEMS

For a 3-qubit system, $\mathcal{G}$ approximates the log-normal distribution with slightly better results for a $\mathcal{G}$ defined with Low-Rank ansatz, with a mean RE of $0.0244(\pm 0.0)$ for *Random* initialization and $0.0914(\pm 0.0)$ for *Uniform* initialization, against $0.0915(\pm 0.0043)$ and $0.092(\pm 0.0078)$ of the Replicate QGAN with *Random* and *Uniform* initialization respectively. Figure 15 displays mean frequencies for the Low-Rank QGAN with the best mean RE score, the Low-Rank QGAN with Random initialization, against the mean frequencies of the replicate with the same initialization. This result is already expected since we observed in Section 3.1 the QGAN proposed by (ZOUFAL; LUCCHI; WOERNER, 2019) can also approximate the log-normal distribution with the current configurations. We can see the progression of RE over the $1,000$ training epochs through all 10 executions displayed in Figure 16. With the replicate having a more stable optimization relative to the Low-Rank QGAN.

Figure 15 – Mean frequencies comparison for a 3-qubit system. Using *Random* initialization.



(a) Low-Rank QGAN, 2 *build qubits*

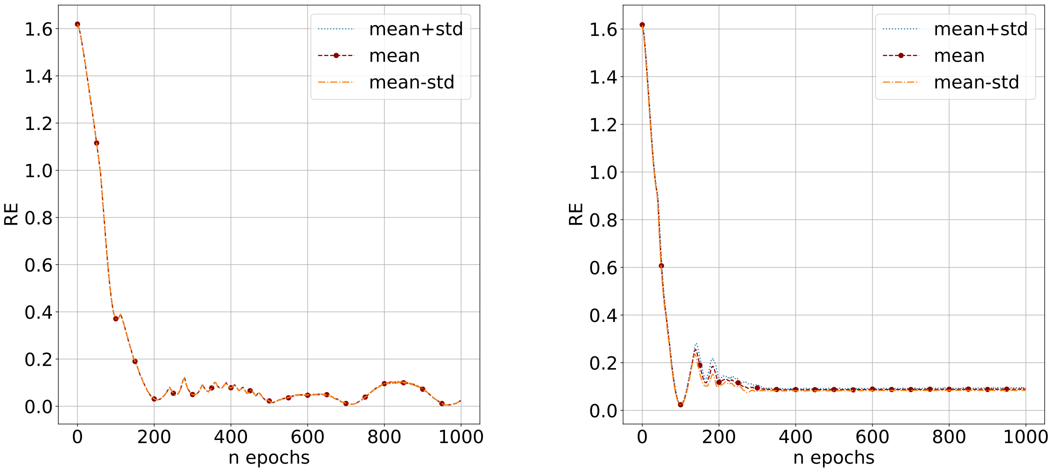
(b) Replicate QGAN

Source: The author (2023)

Figure 16 – Mean RE progression for a 3-qubit system.

(a) Low-Rank QGAN, 2 *build qubits*

(b) Replicate QGAN



Source: The author (2023)

5.2 EXPERIMENTS FOR SYSTEMS WITH QUBIT NUMBERS FROM 4 TO 6

When performing experiments on systems with qubit number higher than 3, we used different qubit grouping (or *Build qubits*) of the Low-Rank for testing different circuit architectures. Using the Low-Rank QGAN, we obtained better overall RE scores than the Replicate QGAN. Tables 1 and 2 summarize the mean RE scores of the 10 experiment iterations for Low-Rank and the Replicate QGAN.

Table 1 – Mean RE for the Replicate QGAN for *Random* and *Uniform* initialization

n-qubits **init type**	**4**	**5**	**6**
Random	$0.5751(\pm0.0808)$	$1.4871(\pm0.3054)$	$1.8939(\pm1.6134)$
Uniform	$0.591(\pm0.1386)$	$1.5159(\pm0.2347)$	$2.1781(\pm1.3231)$

Source: The author (2023)

From RE scores summary on table 2, we observe the initialization highly affects the resulting distribution approximation. Because though we were able to obtain better overall results using Low-Rank QGAN, the experiments on *Random* initialization still resulted in better RE scores.

For a 6-qubit system, the resulting ansatz of Low-Rank has a total of 3 and 8 CNOTs, using 2 and 3 *build qubits*, respectively. Against 5 CNOTs of the ansatz used in the Replicate QGAN. This evidences that we could obtain better approximations even with a slightly higher

Table 2 – Mean RE for the Low-Rank QGAN with different *Build qubits* using *Random* and *Uniform* initialization

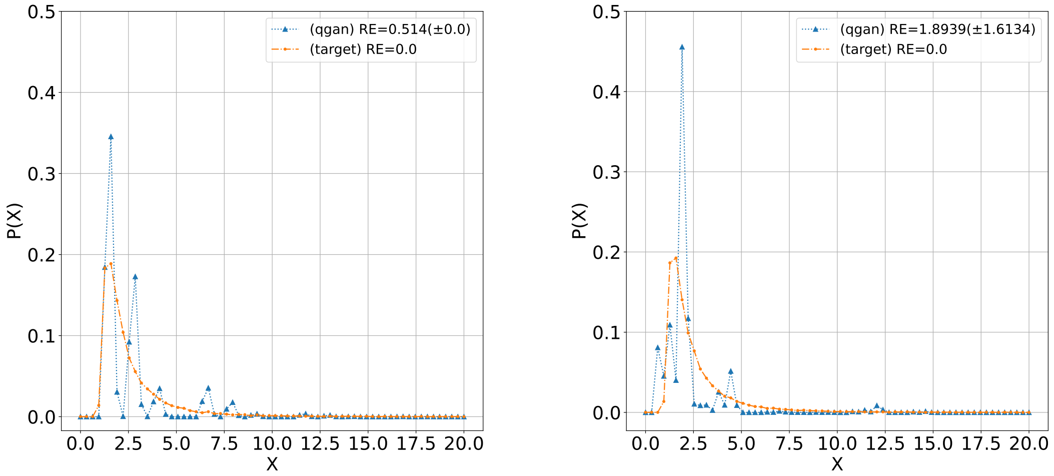
Random initialization		
Build qubits / **n-qubits**	**2**	**3**
4	$0.0847(\pm 0.0)$	-
5	$0.5961(\pm 0.0)$	$0.2229(\pm 0.0)$
6	$0.514(\pm 0.0)$	$0.735(\pm 0.0)$
Uniform initialization		
Build qubits / **n-qubits**	**2**	**3**
4	$0.4965(\pm 0.0)$	-
5	$1.1644(\pm 0.0)$	$0.5474(\pm 0.0)$
6	$0.8215(\pm 0.0)$	$0.5734(\pm 0.0)$

Source: The author (2023)

Figure 17 – Frequencies returned by the Low-Rank and Replicate QGANs for a 6-qubit system using *Random* initialization

(a) Low-Rank QGAN, 2 *Build Qubits*

(b) Replicate QGAN



Source: The author (2023)

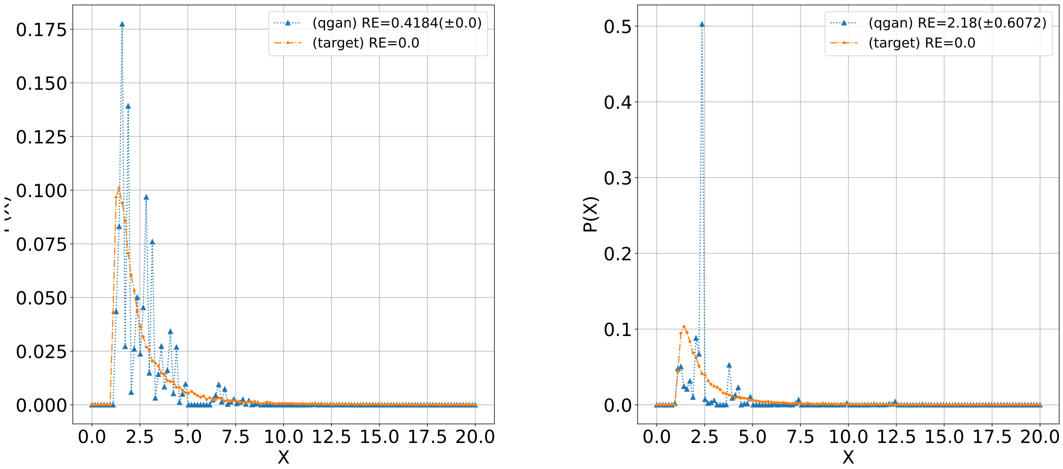
number of CNOTs.

Our goal is to improve the approximation of the distribution loaded by QGAN. Thus, for the remainder of this chapter we will solely focus on 7-qubit systems.

Figure 18 – Frequencies returned by the Low-Rank and Replicate QGANs for a 7-qubit system using *Random* and *Uniform* initialization respectively

(a) Low-Rank QGAN, 3 *Build Qubits*

(b) Replicate QGAN



Source: The author (2023)

5.3 EXPERIMENTS FOR 7 QUBIT SYSTEMS

By running experiments on 7-qubit systems, we obtained similar improvements in terms of RE scores. Tables 3 and 4 summarize the final mean scores of the 10 iterations for the Replicate and Low-Rank QGAN. The ansatz for the Replicate QGAN contained 6 CNOTs. For the ansatz used in Low-Rank QGAN the *build qubits* varied whithin the set $2, 3, 4$, resulting in ansatzes with 3, 8 and 13 CNOTs respectively. As the results summarized in table 2, there is not a big gap between the RE scores for both initialization types compared to the Replicate QGAN. However, unlike the results in table 2, the experiments on *Uniform* initialization resulted in better RE scores. A possible explanation for those differences is the random initialization of parameters starting on better regions, which is also a recurring point of interest in machine learning applications such as Deep Learning (SUTSKEVER et al., 2013; NARKHEDE; BARTAKKE; SUTAONE, 2022).

Table 3 – Mean RE for the Replicate QGAN for *Random* and *Uniform* initialization on 7-qubit system

Init type	***Mean RE***
Random	$2.6332(\pm 0.481)$
Uniform	$2.18(\pm 0.6072)$

Source: The author (2023)

Figure 18 displays the mean frequencies returned by the Low-Rank QGAN, with 3-build qubits, and the Replicate QGAN for a 7-qubit system. With the Low-Rank QGAN using *Random* initialization and the Replicate QGAN using the *Uniform* initialization. We chosed those models for displaying the obtained frequencies because they returned the best RE scores on their respective score summaries (see Tables 3 and 4). In Figure 18 we show that Low-Rank QGAN better approximates the log-normal distribution. This result was expected, given the mean RE score is smaller than the Replicate QGAN.

Table 4 – Mean RE for the Low-Rank QGAN for *Random* and *Uniform* initialization on 7-qubit system

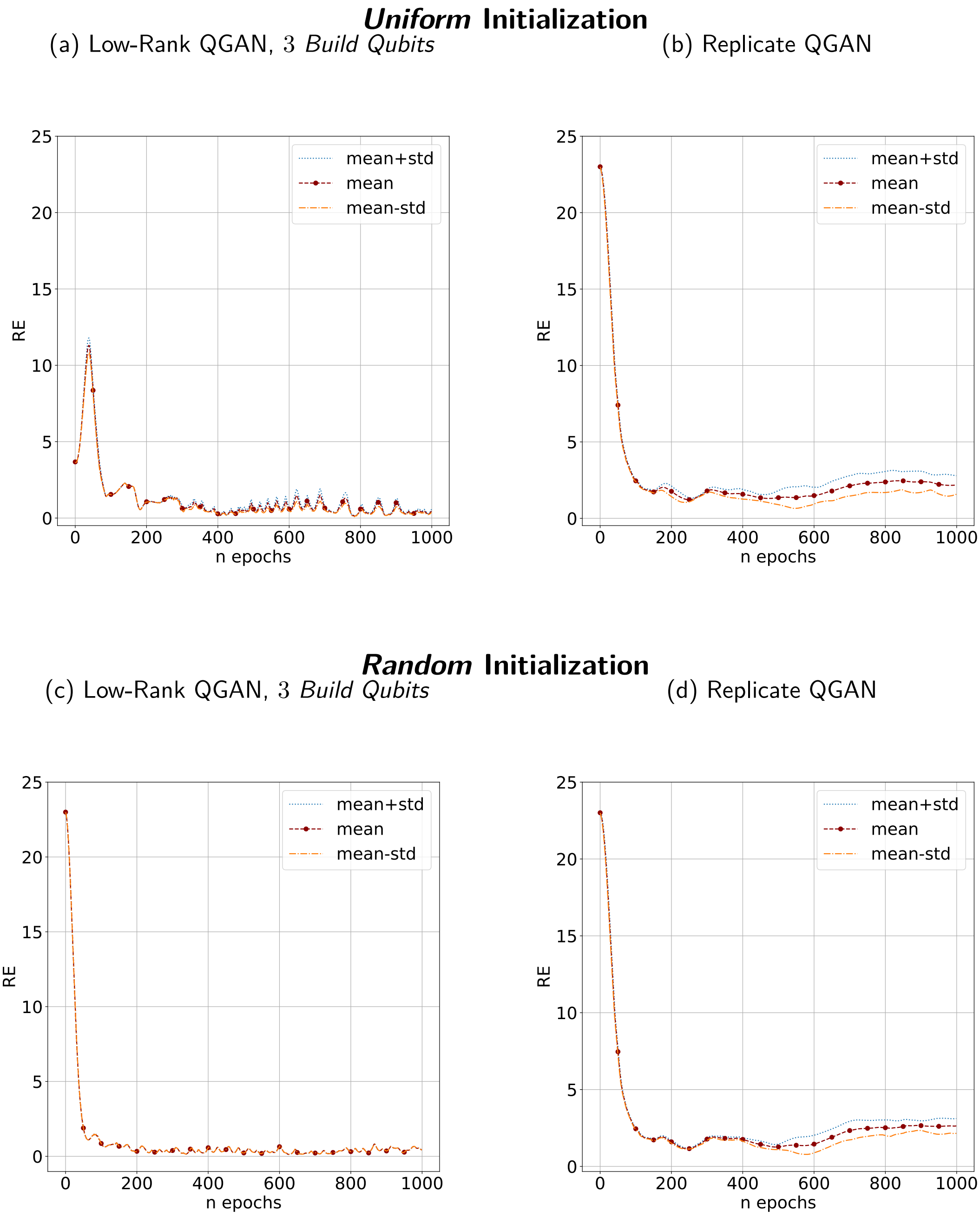
Build qubits **init type**	**2**	**3**	**4**
Random	$0.8973(\pm0.0)$	$0.4184(\pm0.0)$	$0.6013(\pm0.0)$
Uniform	$0.5561(\pm0.0497)$	$0.472(\pm0.102)$	$1.2194(\pm0.1477)$

Source: The author (2023)

Nevertheless, looking into the progression of the mean RE scores over the $1,000$ training epochs of the Replicate QGAN of Figure 19. We remark a slight oscillation of the scores of the Replicate QGAN compared to the Low-Rank QGAN with 3 *Build Qubits*. One possible explanation is that the quantum circuit architecture of a QSPA usually involve several parameters, which means optimizing the results on a larger parameter space than the ansatz used by the Replicate QGAN. Another possible explanation is that ansatz architecture and initial state highly influence how the parameters are optimized. Figures 19c-19d display the mean RE score progression.

Though the mean RE scores in Figure 19 do not represent much oscillation for the Replicate QGAN, we can still obtain better RE scores with the Low-Rank QGAN for the log-normal $(\mu = 1, \sigma = 1)$ distribution.

Figure 19 – Mean RE progression for a 7-qubit system for Low-Rank and Replicate QGANs.



Uniform **Initialization**

(a) Low-Rank QGAN, 3 *Build Qubits*

(b) Replicate QGAN

Random **Initialization**

(c) Low-Rank QGAN, 3 *Build Qubits*

(d) Replicate QGAN

Source: The author (2023)

6 CONCLUSION

In this work, we proposed a method for defining an adaptive ansatz based on the QSPA presented in (ARAUJO et al., 2023), which builds a quantum circuit for loading a quantum state up to a desired Schmidt rank r_s, called Low-Rank. We defined the ansatz without any previous knowledge of the desired quantum state. We used the Low-Rank ansatz in a Quantum Generative Adversarial Network (QGAN) proposed by (ZOUFAL; LUCCHI; WOERNER, 2019) for approximating the log-normal distribution ($\mu = 1, \sigma = 1$) on higher dimension states. We performed this modification because we observed that the performance of the original QGAN deteriorated in terms of Relative Entropy (RE) scores as we increased the number of qubits in the system.

By running experiments on QGANs with an ansatz defined by our methodology, we obtained better RE scores than the original QGAN over different initialization types. As a subsequent result, the frequencies for the distribution loaded by the QGAN using our methodology better approximate the desired log-normal distribution on systems with qubit numbers ranging from 4 to 7. As an example, for a 7-qubit, the best RE score was ≈ 0.514. An improvement if compared to the original QGAN best RE score of ≈ 2.4034, for a 7-qubit system.

We achieved improvements on QGANs using our methodology. However, there are still research opportunities that future works can explore. The frequencies loaded by the QGAN with our ansatz better approximate the desired distribution on systems with more than 3 qubits compared to the original QGAN, but they are not ideal. Relative to RE scores, it would be ≤ 0.05, which is slightly below the mean RE score obtained by our methodology on a 3-qubit system. Thus, a possible future work can be on how to improve the approximations of QGANs.

In this work, we solely focused on the log-normal ($\mu = 1, \sigma = 1$) distribution, which authors in (ZOUFAL; LUCCHI; WOERNER, 2019) used as a distribution for asset pricing on a quantum finance application. Another possible future work can explore different distributions and applications where we could use the QGANs. A subsequent research opportunity lies in the fact that to load other distributions or similar distributions with different parameters (mean, kurtosis, etc.), we must retrain the QGAN. In future works, researchers could investigate how to enable the QGAN to load different distribution types with distinct moments without training the model from scratch.

In our work, we relied heavily on the *random* initialization of parameters without using any previous knowledge of the data. Even in experiments with *uniform* initialization, we prepare the input state using the operator $H^{\otimes n}$. The model becomes susceptible to problems such as barren plateaus and having the optimization stuck at local minima. Another possible future work could explore how to incorporate some previous knowledge about the problem or the data. And apply that knowledge either in the definition of the ansatz or in some other components of the VQA to avoid such issues.

REFERENCES

AGLIARDI, G.; PRATI, E. Optimal tuning of quantum generative adversarial networks for multivariate distribution loading. *Quantum Reports*, MDPI, v. 4, n. 1, p. 75–105, 2022.

AKTAR, S.; BÄRTSCHI, A.; BADAWY, A.-H. A.; EIDENBENZ, S. A divide-and-conquer approach to dicke state preparation. *IEEE Transactions on Quantum Engineering*, IEEE, v. 3, p. 1–16, 2022.

ALEKSANDROWICZ, G. et al. *Qiskit: An Open-source Framework for Quantum Computing*. 2019.

ARAUJO, I. F.; BLANK, C.; ARAÚJO, I. C.; SILVA, A. J. da. Low-rank quantum state preparation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, IEEE, 2023.

BÄRTSCHI, A.; EIDENBENZ, S. Deterministic preparation of dicke states. In: SPRINGER. *International Symposium on Fundamentals of Computation Theory*. Cham, 2019. p. 126–139.

BENEDETTI, M.; LLOYD, E.; SACK, S.; FIORENTINI, M. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, IOP Publishing, v. 4, n. 4, p. 043001, 2019.

BERGHOLM, V.; VARTIAINEN, J. J.; MÖTTÖNEN, M.; SALOMAA, M. M. Quantum circuits with uniformly controlled one-qubit gates. *Physical Review A*, APS, v. 71, n. 5, p. 052330, 2005.

BIAMONTE, J.; WITTEK, P.; PANCOTTI, N.; REBENTROST, P.; WIEBE, N.; LLOYD, S. Quantum machine learning. *Nature*, Nature Publishing Group, v. 549, n. 7671, p. 195–202, 2017.

BISHOP, C. M.; NASRABADI, N. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006.

BISHOP, C. M. et al. *Neural networks for pattern recognition*. [S.l.]: Oxford university press, 1995.

CEREZO, M.; ARRASMITH, A.; BABBUSH, R.; BENJAMIN, S. C.; ENDO, S.; FUJII, K.; MCCLEAN, J. R.; MITARAI, K.; YUAN, X.; CINCIO, L. et al. Variational quantum algorithms. *Nature Reviews Physics*, Nature Publishing Group, v. 3, n. 9, p. 625–644, 2021.

CHEN, S. Y.-C.; YANG, C.-H. H.; QI, J.; CHEN, P.-Y.; MA, X.; GOAN, H.-S. Variational quantum circuits for deep reinforcement learning. *IEEE Access*, IEEE, v. 8, p. 141007–141024, 2020.

CRUZ, D.; FOURNIER, R.; GREMION, F.; JEANNEROT, A.; KOMAGATA, K.; TOSIC, T.; THIESBRUMMEL, J.; CHAN, C. L.; MACRIS, N.; DUPERTUIS, M.-A. et al. Efficient quantum algorithms for ghz and w states, and implementation on the ibm quantum computer (adv. quantum technol. 5-6/2019). *Advanced Quantum Technologies*, Wiley Online Library, v. 2, n. 5-6, p. 1970031, 2019.

DIVINCENZO, D. P. Two-bit gates are universal for quantum computation. *Physical Review A*, APS, v. 51, n. 2, p. 1015, 1995.

DIVINCENZO, D. P. Quantum gates and circuits. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, The Royal Society, v. 454, n. 1969, p. 261–276, 1998.

DIVINCENZO, D. P. The physical implementation of quantum computation. *Fortschritte der Physik: Progress of Physics*, Wiley Online Library, v. 48, n. 9-11, p. 771–783, 2000.

DJORDJEVIC, I. B. *Quantum information processing, quantum computing, and quantum error correction: an engineering approach*. [S.l.]: Academic Press, 2021.

DUAN, B.; HSIEH, C.-Y. Hamiltonian-based data loading with shallow quantum circuits. *Physical Review A*, APS, v. 106, n. 5, p. 052422, 2022.

ECKART, C.; YOUNG, G. The approximation of one matrix by another of lower rank. *Psychometrika*, Springer, v. 1, n. 3, p. 211–218, 1936.

GLEINIG, N.; HOEFLER, T. An efficient algorithm for sparse quantum state preparation. In: IEEE. *2021 58th ACM/IEEE Design Automation Conference (DAC)*. [S.l.], 2021. p. 433–438.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016.

GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial networks. *Communications of the ACM*, ACM New York, NY, USA, v. 63, n. 11, p. 139–144, 2020.

GROVER, L.; RUDOLPH, T. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002.

HOLLY Bible: King James Version. In: PSALMS. Bíblia Online, 1769. chap. 111:10. Available at: <https://www.bibliaonline.com.br/kjv/sl/111>. Accessed on: May 23rd, 2023.

IBM. *ibmq_belem device details and qubit connectivity*. 2023. Available at: <https://quantum-computing.ibm.com/services/resources?order=qubits%20ASC&system=ibmq_belem>. Accessed on: May 23rd, 2023.

KANDALA, A.; MEZZACAPO, A.; TEMME, K.; TAKITA, M.; BRINK, M.; CHOW, J. M.; GAMBETTA, J. M. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *nature*, Nature Publishing Group UK London, v. 549, n. 7671, p. 242–246, 2017.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

LLOYD, S. Almost any quantum logic gate is universal. *Physical review letters*, APS, v. 75, n. 2, p. 346, 1995.

LLOYD, S.; WEEDBROOK, C. Quantum generative adversarial learning. *Physical review letters*, APS, v. 121, n. 4, p. 040502, 2018.

MARIN-SANCHEZ, G.; GONZALEZ-CONDE, J.; SANZ, M. Quantum algorithms for approximate function loading. *arXiv preprint arXiv:2111.07933*, 2021.

MCMAHON, D. *Quantum computing explained*. [S.l.]: John Wiley & Sons, 2007.

MITARAI, K.; NEGORO, M.; KITAGAWA, M.; FUJII, K. Quantum circuit learning. *Physical Review A*, APS, v. 98, n. 3, p. 032309, 2018.

MOTTONEN, M.; VARTIAINEN, J. J.; BERGHOLM, V.; SALOMAA, M. M. Transformation of quantum states using uniformly controlled rotations. *arXiv preprint quant-ph/0407010*, 2004.

NAKAJI, K.; UNO, S.; SUZUKI, Y.; RAYMOND, R.; ONODERA, T.; TANAKA, T.; TEZUKA, H.; MITSUDA, N.; YAMAMOTO, N. Approximate amplitude encoding in shallow parameterized quantum circuits and its application to financial market indicators. *Physical Review Research*, APS, v. 4, n. 2, p. 023136, 2022.

NARKHEDE, M. V.; BARTAKKE, P. P.; SUTAONE, M. S. A review on weight initialization strategies for neural networks. *Artificial intelligence review*, Springer, v. 55, n. 1, p. 291–322, 2022.

NEUMANN, N.; PHILLIPSON, F.; VERSLUIS, R. Machine learning in the quantum era. *Digitale Welt*, eMedia Gesellschaft für Elektronische Medien mbH, v. 3, p. 24–29, 2019.

NIELSEN, M. A.; CHUANG, I. L. *Quantum computation and Quantum information*. [S.l.]: Cambridge University Press India, 2000.

NIU, M. Y.; ZLOKAPA, A.; BROUGHTON, M.; BOIXO, S.; MOHSENI, M.; SMELYANSKYI, V.; NEVEN, H. Entangling quantum generative adversarial networks. *Physical Review Letters*, APS, v. 128, n. 22, p. 220505, 2022.

PASZKE, A.; GROSS, S.; CHINTALA, S.; CHANAN, G.; YANG, E.; DEVITO, Z.; LIN, Z.; DESMAISON, A.; ANTIGA, L.; LERER, A. Automatic differentiation in PyTorch. In: *NIPS Autodiff Workshop*. [S.l.: s.n.], 2017.

PLESCH, M.; BRUKNER, Č. Quantum-state preparation with universal gate decompositions. *Physical Review A*, APS, v. 83, n. 3, p. 032302, 2011.

PRESKILL, J. Quantum computing in the nisq era and beyond. *Quantum*, Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, v. 2, p. 79, 2018.

PRIETO, C. B. Variational quantum architectures. applications for noisy intermediate-scale quantum computers. Universitat de Barcelona, 2022.

ROMERO, J.; ASPURU-GUZIK, A. Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. *Advanced Quantum Technologies*, Wiley Online Library, v. 4, n. 1, p. 2000003, 2021.

SCHULD, M.; BERGHOLM, V.; GOGOLIN, C.; IZAAC, J.; KILLORAN, N. Evaluating analytic gradients on quantum hardware. *Physical Review A*, APS, v. 99, n. 3, p. 032331, 2019.

SCHULD, M.; BOCHAROV, A.; SVORE, K. M.; WIEBE, N. Circuit-centric quantum classifiers. *Physical Review A*, APS, v. 101, n. 3, p. 032308, 2020.

SCHULD, M.; PETRUCCIONE, F. *Supervised learning with quantum computers*. [S.l.]: Springer, 2018.

SHENDE, V. V.; BULLOCK, S. S.; MARKOV, I. L. Synthesis of quantum logic circuits. In: *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*. Shanghai, China: [s.n.], 2005. p. 272–275.

SITU, H.; HE, Z.; WANG, Y.; LI, L.; ZHENG, S. Quantum generative adversarial network for generating discrete distribution. *Information Sciences*, Elsevier, v. 538, p. 193–208, 2020.

SUTSKEVER, I.; MARTENS, J.; DAHL, G.; HINTON, G. On the importance of initialization and momentum in deep learning. In: PMLR. *International conference on machine learning*. Atlanta, USA, 2013. p. 1139–1147.

TRUGENBERGER, C. A. Probabilistic quantum memories. *Physical Review Letters*, APS, v. 87, n. 6, p. 067901, 2001.

VALE, R.; AZEVEDO, T. M. D.; ARAÚJO, I.; ARAUJO, I. F.; SILVA, A. J. da. Decomposition of multi-controlled special unitary single-qubit gates. *arXiv preprint arXiv:2302.06377*, 2023.

VENTURA, D.; MARTINEZ, T. A quantum associative memory based on grover's algorithm. In: *Artificial Neural Nets and Genetic Algorithms*. Portorož, Slovenia: [s.n.], 1999. p. 22–27.

VERAS, T. M. de; SILVA, L. D. da; SILVA, A. J. da. Double sparse quantum state preparation. *Quantum Information Processing*, Springer, v. 21, n. 6, p. 1–13, 2022.

YANOFSKY, N. S.; MANNUCCI, M. A.; MANNUCCI, M. A. *Quantum computing for computer scientists*. [S.l.]: Cambridge University Press Cambridge, 2008.

ZENG, J.; WU, Y.; LIU, J.-G.; WANG, L.; HU, J. Learning and inference on generative adversarial quantum circuits. *Physical Review A*, APS, v. 99, n. 5, p. 052306, 2019.

ZOUFAL, C.; LUCCHI, A.; WOERNER, S. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Information*, Nature Publishing Group, v. 5, n. 1, p. 1–9, 2019.