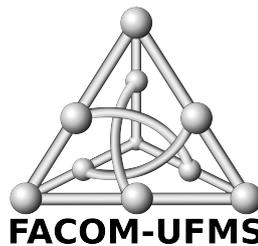


UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL  
FACULDADE DE COMPUTAÇÃO

# **A Computação Quântica e as Redes Adversárias Generativas**

Trabalho de Conclusão de Curso

Gustavo Felipe Gonçalves Reis



Campo Grande – Mato Grosso do Sul

2024

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL  
FACULDADE DE COMPUTAÇÃO

Gustavo Felipe Gonçalves Reis

## **A Computação Quântica e as Redes Adversárias Generativas**

Trabalho de Conclusão de Curso submetido à Faculdade de Computação (FACOM) da Universidade Federal de Mato Grosso do Sul (UFMS) como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador(a): Prof. Dr. Milton Ernesto Romero Romero

Campo Grande – Mato Grosso do Sul

2024

*Este trabalho é dedicado a todos aqueles que se movem e se desdobram pelo acreditar,  
pois acreditar nos faz mais humanos.*

# Agradecimentos

A Deus, meu amigo confidente. A minha família, meu pai <sup>1</sup>, mãe <sup>2</sup> e irmãs <sup>3</sup> que são a minha base. Agradeço também a todos os amigos de longa data e amigos que a jornada acadêmica me deu, por todo o companheirismo e momentos que tornaram essa jornada muito mais leve e especial, um agradecimento extraordinário para o Eduardo Valim e o Luiz Henrique Pio Freire que tiveram influência direta para a execução deste trabalho com recursos computacionais.

Um agradecimento a Universidade Federal de Mato Grosso do Sul (UFMS), que possibilitou a realização da valorosa graduação acadêmica e me capacitou para ser atuante na vasta área da computação, instituição que segue formando excelentes profissionais e pesquisadores. Gostaria de expressar minha gratidão à banca examinadora, ao Professor Doutor Evandro Mazina Martins pela cuidadosa avaliação, ao Professor Doutor Marco Aurélio Stefanos pela compreensão e disponibilidade e ao Professor Doutor Renato Porfirio Ishii pela avaliação e por toda ajuda na reta final de graduação. Por fim, agradeço ao Professor Doutor Milton Ernesto Romero Romero, por todo o suporte, conselhos e experiências compartilhadas, e claro pela orientação deste trabalho, pelas reuniões e correções que fizeram toda a diferença neste projeto.

---

<sup>1</sup> Sebastião Soares dos Reis

<sup>2</sup> Rosane da Silva Gonçalves Reis

<sup>3</sup> Andressa Helena Gonçalves Reis  
Débora Renata Gonçalves Reis

*Seja você quem for  
Seja qual for a posição social que você tenha na vida  
a mais alta ou a mais baixa  
Tenha sempre como meta muita força  
Muita determinação e  
Sempre faça tudo com muito amor e com muita fé em Deus  
Que um dia você chega lá  
De alguma maneira você chega lá!  
(Ayrton Senna)*

# Resumo

O avanço da computação quântica tem aberto novas possibilidades para resolver problemas computacionais complexos, destacando-se o aprendizado de máquinas quântico (QML) como alternativa aos métodos clássicos de inteligência artificial. Redes neurais adversárias quânticas (QGANs), uma adaptação das GANs para dispositivos quânticos, têm evoluído desde testes sintéticos até aplicações reais. Este trabalho apresenta uma alternativa para gerar imagens de 28x28 pixels em dispositivos quânticos, utilizando conjuntos de dados como MNIST e Fashion MNIST. Apresenta-se uma arquitetura QGAN capaz de gerar imagens com qualidade comparável a modelos clássicos, utilizando menos parâmetros. Os resultados obtidos demonstram o potencial das QGANs para revolucionar a geração de imagens e abrem caminho para novas aplicações em inteligência artificial.

**Palavras-chave:** qubit, superposição, computação quântica, circuitos quânticos, simulação quântica, aprendizado de máquina, redes generativas, geração de imagens.

# Abstract

Advances in quantum computing have opened up new possibilities for solving complex computational problems, with quantum machine learning (QML) standing out as an alternative to classical artificial intelligence methods. Quantum generative adversarial networks (QGANs), an adaptation of GANs for quantum devices, have evolved from synthetic tests to real-world applications. This work presents an alternative for generating 28x28 pixel images on quantum devices, using datasets such as MNIST and Fashion MNIST. A QGAN architecture capable of generating images with quality comparable to classical models, using fewer parameters, is presented. The obtained results demonstrate the potential of QGANs to revolutionize image generation and pave the way for new applications in artificial intelligence.

**Keywords:** qubit, superposition, quantum computing, quantum circuits, quantum simulation, machine learning, generative networks, image generation.

# Lista de abreviaturas e siglas

CGCC	Classical Generator - Classical Critic
CUDA	Compute Unified Device Architecture
DCGAN	Deep Convolutional Generative Adversarial Network
FMNIST	Fashion Modified National Institute of Standards and Technology
GAN	Generative Adversarial Network
GP	Gradient Penalty
IQGAN	Incremental Quantum Generative Adversarial Network
JS	Jensen-Shannon
ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology
NISQ	Noisy Intermediate-Scale Quantum
PCA	Principal Component Analysis
PQC	Parameterized Quantum Circuits
PQWGAN	Patch Quantum Wasserstein Generative Adversarial Network
QGAN	Quantum Generative Adversarial Network
QGCC	Quantum Generator - Classical Critic
QGQC	Quantum Generator - Quantum Critic
QML	Quantum Machine Learning
QuGAN	Quantum State Fidelity based Generative Adversarial Network
QWGAN-GP	Quantum Wasserstein Generative Adversarial Network - Gradient Penalty
WGAN	Wasserstein Generative Adversarial Network
WGAN-GP	Wasserstein Generative Adversarial Network - Gradient Penalty

# Lista de símbolos

$\in$	Pertence
$\mathbf{z}$	Vetor amostrado
$G(\cdot)$	Função gerador
$D(\cdot)$	Função discriminador
$\mathbb{E}$	Função expectativa de acerto
$\mathcal{D}$	Conjunto D das funções 1-Lipschitz
$\lambda$	Coeficiente de penalidade do GP
$\hat{\mathbf{x}}$	Vetor interpolação de $\mathbf{x}$ e $\mathbf{z}$
$\nabla$	Função gradiente
$\mathbf{M}$	Matriz de dimensão M
$ \cdot\rangle$	Estado quântico, notação de Dirac
$\otimes$	Produto tensorial
$R_Y, \sigma_Y$	Rotação no eixo Y
$\psi$	Superposição de estado quântico
$\mathcal{U}_L$	Operação unitária ZYZ
$\phi$	1° Ângulo Z
$\omega$	2° Ângulo Z
$\theta$	Ângulo Y
$\mathbf{T}_r$	Função traço
$\mathbf{p}$	Patch
$\eta_1, \eta_2, \beta_1, \beta_2$	Hiperparâmetros Adam
$\mathcal{L}(\cdot)$	Função perda
$\partial$	Derivada

# Sumário

<b>1</b>	<b>Introdução</b>	<b>11</b>
1.1	Apresentação Geral	11
<b>2</b>	<b>Fundamentação Teórica</b>	<b>15</b>
2.1	Conhecimentos Preliminares	15
2.1.1	Circuitos Quânticos Parametrizados	15
2.1.1.1	Qubit, Superposição e Emaranhamento Quântico	15
2.1.1.2	Notação de Dirac e Portas Quânticas	16
2.1.1.3	Representação do Circuito Quântico e Esfera de Bloch	17
2.1.2	Redes Neurais Adversárias	19
2.1.3	Wasserstein GAN	19
2.2	Referencial Teórico e Trabalhos Relacionados	21
2.2.1	QGAN (Quantum Generative Adversarial Network)	21
2.2.2	Geração de Imagens em QGANs	21
2.2.3	Quantum Wasserstein GANs	22
2.2.4	Patch Quantum Wasserstein GANs	22
<b>3</b>	<b>Framework PQWGAN</b>	<b>23</b>
3.1	O Framework	23
3.1.1	Algoritmo do Gerador	23
3.1.2	Algoritmo do Discriminador e Treino	25
3.1.2.1	Estrutura do Crítico	25
3.1.2.2	Alvo do Treinamento	26
<b>4</b>	<b>Metodologia</b>	<b>28</b>
4.1	PQWGAN Metodologia de Experimentação	28
4.1.1	Dataset e Bibliotecas	28
4.1.2	Implementação e Ambiente de Execução	28
4.1.2.1	Arquivo <code>train.py</code>	30
4.1.2.2	Arquivo <code>QGCC.py</code>	30
4.1.2.3	Arquivo <code>QGQC.py</code>	31
4.1.2.4	Arquivo <code>wgan.py</code>	32
<b>5</b>	<b>Experimentação e Resultados</b>	<b>33</b>
5.1	Observações	33
5.2	Geração de Imagem com o framework PQWGAN	33

5.2.1	MNIST binário . . . . .	35
5.2.2	FMNIST binário . . . . .	36
5.2.3	MNIST triplo . . . . .	37
<b>6</b>	<b>Conclusão . . . . .</b>	<b>38</b>
	<b>Referências . . . . .</b>	<b>39</b>
	<b>Apêndices</b>	<b>43</b>
	<b>APÊNDICE A Disponibilidade de arquivos . . . . .</b>	<b>44</b>

# 1

## Introdução

### 1.1 Apresentação Geral

Considerando as Redes Neurais Adversárias (GANs) como um avanço para o aprendizado profundo na era das redes generativas [Goodfellow et al. 2014]. A estratégia das GANs consiste em um gerador e um discriminador competindo entre si, o gerador é responsável por gerar dados (no caso deste trabalho, imagens), e o discriminador distinguir entre dados reais e dados gerados, gerador e discriminador tentam sempre superar-se um ao outro. Esta arquitetura propõe que a distribuição do gerador seja semelhante à distribuição dos dados de treinamento, relacionando matematicamente a divergência de Jensen-Shannon (JS), que seria diminuir a divergência entre as distribuições [Goodfellow et al. 2014]. As redes neurais adversárias já foram implantadas em muitas áreas de aplicação, utilizadas para a geração de imagens [Radford, Metz e Chintala 2016], previsão futura na geração de vídeos [Vondrick, Pirsivash e Torralba 2016], sintetização de texto para imagem [Reed et al. 2016], e tradução de imagem para imagem [Emami et al. 2020].

Pode-se realçar os ganhos que as redes neurais adversárias têm, como a capacidade de melhorar as suas gerações a cada treinamento e por causa da estratégia adversária onde deve-se efetuar o treino, se no gerador ou discriminador. Também deve-se mencionar o que ocasionou a necessidade de desenvolver complementos e correções desde as primeiras ideias até modelos atuais conhecidos. As GANs podem sofrer de problemas de Vanish Gradient (Gradiente Desvanecente) que ocorre quando o gradiente utilizado para atualizar os pesos da rede se torna muito pequeno, o efeito das pequenas derivadas principalmente das primeiras camadas da rede tornam o gradiente muito baixo, quase nulo. Mode Collapse (modo colapso) acontece quando uma das partes da GAN começa a dominar sobre a outra, seja o discriminador ou o gerador, e mesmo que a evolução conjunta das partes estejam se satisfazendo mutuamente, devido a dominância de uma das partes os resultados coletados são degenerados e a GAN caminha para a perda de capacidade de gerar dados realistas. As GANs também sofrem com a falta de critérios

de parada de treinamento [Arjovsky e Bottou 2017] [Arjovsky, Chintala e Bottou 2017]. Também são apresentadas as alternativas desenvolvidas progressivamente até a proposta discutida neste trabalho. A intervenção clássica para os problemas das GANs de maior influência para esta discussão é o Wasserstein GAN (WGAN) [Arjovsky, Chintala e Bottou 2017], que visando diminuir a distância de Wasserstein (pode ser vista na seção 2.1.3) reformula a maneira como o treinamento é realizado. Empiricamente, o framework WGAN mostrou a capacidade de lidar com os problemas citados. Um adendo é que, semelhante a outros GANs clássicos, o treinamento em conjuntos de dados grande aumenta a complexidade das redes, e que também intui-se para um grande uso de recursos computacionais.

Com a evolução dos estudos da eletrodinâmica Feynman desenvolvia estratégias para processar informações em sistemas quânticos [Feynman 2018]. Esse campo de estudo continuou evoluindo até o que hoje se conhece como computação quântica. O desenvolvimento da computação quântica têm trazido avanços para solucionar vários problemas computacionalmente complexos da ciência da computação, como o problema da fatoração eficiente de números primos com o algoritmo de Shor [Shor 1994] [Sousa 2020]. Pensando em como a mecânica quântica pode gerar padrões contra-intuitivos nos dados, isso nos traz a ideia de que aparentemente computadores quânticos podem reconhecer padrões que para um computador clássico é uma tarefa desafiadora [Biamonte et al. 2017]. Uma das áreas que se desenvolveu com a evolução da computação quântica foi o Quantum Machine Learning (QML), que consiste em programar rotinas de aprendizado de máquina já existentes em sistemas quânticos [West et al. 2023] visando a melhora de eficiência e performance. Benefícios registrados do QML incluem aceleração de tempo de treinamento em [Rebentrost, Mohseni e Lloyd 2014], [Lloyd, Mohseni e Rebentrost 2013], [Wiebe, Kapoor e Svore 2014] ou obtém-se melhor desempenho do modelo em [Amin et al. 2018], [Havlíček et al. 2019], [West et al. 2023], [West, Sevier e Usman 2023], [West, Sevier e Usman 2023], [Du et al. 2020]. A expectativa é que futuramente a QML seja algo complementar e até mesmo uma substituta em muitos casos para as alternativas clássicas à medida que a complexidade dos problemas crescer seja cada vez mais palpável. O maior desafio atual para avançar com a tomada quântica é o de estarmos na era NISQ (Noisy Intermediate-Scale Quantum) [Preskill 2018], onde os nossos sistemas quânticos possuem um comportamento ruidoso. A tarefa de executar de forma confiável algoritmos quânticos de grande escala no hardware quântico atual é difícil pelos ainda não mitigados problemas de engenharia de sistemas quânticos para solucionar o ruído. Mesmo que isso dificulte o desenvolvimento de soluções quânticas, não impede que o desenvolvimento de QML seja baseado visando o funcionamento em dispositivos NISQ [Biamonte et al. 2017], e também existem soluções híbridas quânticas clássicas com PQCs (Parameterized Quantum Circuits), os circuitos quânticos parametrizados [Benedetti et al. 2019].

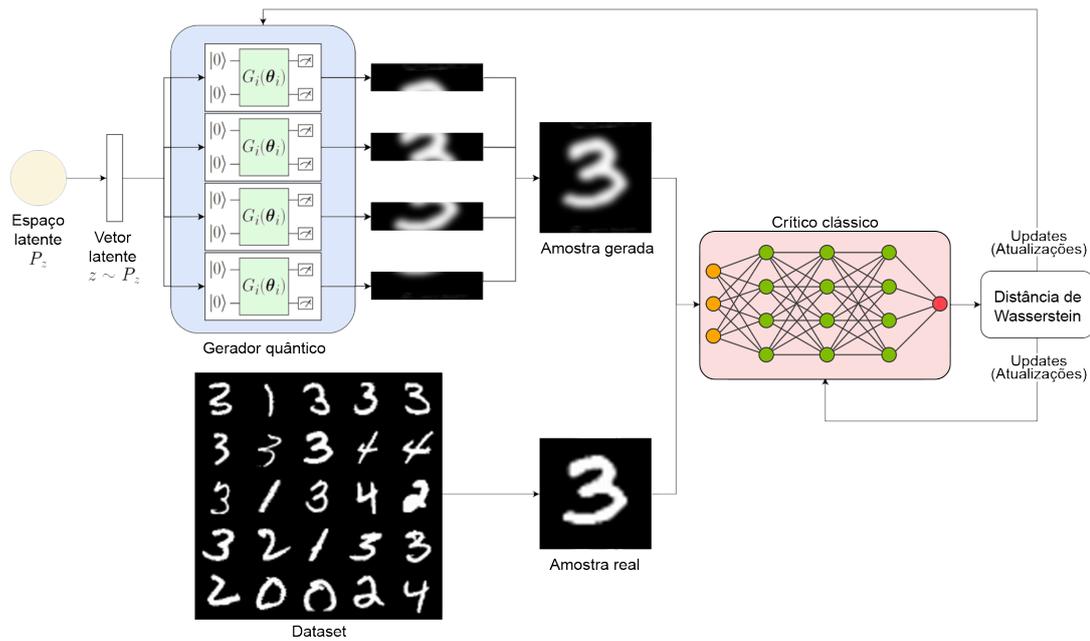


Figura 1 – Estrutura da arquitetura da PQWGAN, espaço e vetor latentes, gerador quântico demonstrado pelo bloco azul e seus sub-geradores que resultam de uma imagem gerada. O crítico clássico é demonstrado pelo bloco vermelho que avalia a amostra real e a amostra gerada, de acordo com a distância de Wasserstein ele retorna os updates necessários.

Neste ponto, na interseção das duas áreas, uma já estabelecida e em constante evolução que são as GANs e outra que está tomando forma atualmente, a junção dessas áreas originou as denominadas QGANs (Quantum Generative Adversarial Networks) as redes neurais adversárias quânticas [Lloyd e Weedbrook 2018] [Dallaire-Demers e Killoran 2018], que é uma alternativa para o aprendizado generativo. Pode-se afirmar que as QGANs ainda estão em uma fase muito inicial de sua capacidade, e mencionando alguns frameworks que foram propostos que lidam com dados de baixa dimensão entre esse modelos para distribuições com probabilidades simples temos [Zoufal, Lucchi e Woerner 2019], [Assouel, Jacquier e Kondratyev 2021], [Li, Topaloglu e Ghosh 2021]. Nos modelos de framework que trabalham com geração de imagens tem-se os que trabalham com baixa resolução [Huang et al. 2021], outros que têm estratégia de redução na dimensão das imagens com a análise de componentes principais (PCA) [Stein et al. 2021], [Chu et al. 2023]. Chegando no objeto de estudo deste trabalho, o framework que tem a estratégia de dividir em pedaços a tarefa para poder trabalhar de forma paralela com imagens de alta resolução [Tsang et al. 2023]. Ainda existe mais para se explorar o conhecimento em QGANs sobre a variação e ajustes de parâmetros fornecidos para um gerador quântico, e como isso afeta o desempenho de sua saída. Num retrospecto de trabalhos anteriores em QGANs que se baseiam em baixas dimensões, as imagens passaram por um processo de compressão em [Stein et al. 2021], [Chu et al. 2023], ou foi utilizado um conjunto sintético de dimensão 2x2 pixels em [Huang et al. 2021], e agora visando trabalhar com aplicações mais realistas e derrubando as

restrições que esses frameworks apresentavam, a alternativa é fragmentar o conjunto e trabalhar em paralelo com o conjunto de dados de dimensões maiores.

Este trabalho busca apresentar a atual trajetória da QGANs e apresentar o framework PQWGAN (Patch Quantum Wasserstein Generative Adversarial Network) [Tsang et al. 2023] desde sua estratégia para trabalhar com imagens de alta resolução para as QGANs atuais, como foi estruturado em sua arquitetura híbrida clássica-quântica mostrada na figura 1. Além de apresentá-lo, também foram propostas algumas alterações na implementação original do PQWGAN para ser possível a sua experimentação em ambientes CUDA. Ainda foram reproduzidos alguns experimentos realizados em [Tsang et al. 2023] mostrando ser tangível a simulação de dispositivos quânticos para aplicações reais.

# 2

## Fundamentação Teórica

### 2.1 Conhecimentos Preliminares

#### 2.1.1 Circuitos Quânticos Parametrizados

Com o avanço da computação quântica e seus dispositivos, e com a compreensão de que a era NISQ dos dispositivos quânticos têm forte influência na forma como são desenvolvidos métodos para testar os processadores quânticos atuais, foram idealizados os circuitos quânticos parametrizados (PQCs), capazes de fornecer não somente uma base concreta de implementar os algoritmos e mostrar a supremacia quântica sobre métodos da computação clássica mesmo na era ruidosa dos sistemas quânticos [Benedetti et al. 2019]. Os PQCs são constituídos por portas análogas às portas lógicas em sistemas digitais da computação clássica, essas portas quânticas podem ser fixas como por exemplo **CNOT**, e portas ajustáveis, como por exemplo rotações de qubit. Posteriormente na seção 3.1.1 é visto como foi utilizado um circuito quântico parametrizado para realizar o encoder de informações de uma imagem em tons de cinza, para um conjunto de qubits.

##### 2.1.1.1 Qubit, Superposição e Emaranhamento Quântico

De forma análoga à computação clássica toda informação é constituída por dois estados diferentes em um bit “0” ou “1”, na computação quântica também utiliza-se esses dois estados da informação com o acréscimo da superposição de estado, quando o bit quântico denominado *qubit* pode representar substancialmente os valores “0” e “1” ao mesmo tempo. O benefício da superposição é o de poder efetuar cálculos sobre todos os resultados possíveis para a mesma ocasião, por exemplo, num caso simples em que será realizada uma operação sobre três qubits, para calcular sobre cada entrada possível dessa operação seria feito  $2^{n_{qubits}} = 8$  cálculos diferentes. No contexto de superposição esse cálculo seria feito apenas uma vez e a projeção final induzida provavelmente retornaria o resultado correto esperado. Intrinsecamente as operações sobre qubits

não dependem do valor contido neles para serem executados, o que pode significar que alguns algoritmos quânticos com tais características podem ser exponencialmente mais baratos que algoritmos clássicos.

Além da superposição há outra propriedade da mecânica quântica utilizada na computação quântica, o emaranhamento ou entrelaçamento. Esse fenômeno acontece quando duas ou mais partículas têm seu comportamento ligado à outra instantaneamente, e como nos sistemas quânticos os qubits são constituídos de partículas, esse comportamento pode ser reproduzido e experimentado em computadores quânticos. Na literatura esses qubits emaranhados são denominados de estados de Bell [Nielsen e Chuang 2010].

### 2.1.1.2 Notação de Dirac e Portas Quânticas

Utiliza-se a notação de Dirac para a representação matemática de todas as mecânicas quânticas envolvidas no desenvolvimento de sistemas e rotinas, as representações básicas fundamentais são:

$$\text{- ket: } |a\rangle = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

$$\text{- bra: } \langle b| = |b\rangle^+ = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}^+ = (b_1^* \ b_2^*)$$

$$\text{- bra-ket: } \langle b|a\rangle = a_1 b_1^* + a_2 b_2^* = \langle a|b\rangle^* \in \mathbb{C}$$

$$\text{- ket-bra: } |a\rangle\langle b| = \begin{pmatrix} a_1 b_1^* & a_1 b_2^* \\ a_2 b_1^* & a_2 b_2^* \end{pmatrix}$$

Tomado esse conhecimento, pode-se definir os estados quânticos  $|0\rangle$  e  $|1\rangle$  como:

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ e } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \text{ que são ortogonais } \langle 0|1\rangle = 1 \cdot 0 + 0 \cdot 1 = 0$$

Também temos os estados quânticos normalizados como:

$$\langle \psi|\psi\rangle = 1, |\psi\rangle = \frac{1}{\sqrt{2}} \cdot (|0\rangle + |1\rangle) = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\langle \psi|\psi\rangle = \left(\frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}}\right) + \left(\frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}}\right) = \frac{1}{2} + \frac{1}{2} = 1.$$

As portas lógicas quânticas mais comuns em PQC's são as portas de Pauli [Barenco et al. 1995], responsáveis pelas rotações do qubit, e também as mudanças de base na representação tridimensional vistas na figura 3. A primeira porta é a porta  $\sigma_X$

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = |0\rangle\langle 1| + |1\rangle\langle 0|,$$

que promove a rotação no eixo x, a seguinte porta é a porta  $\sigma_z$

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|,$$

que promove a rotação no eixo z, e para efetuar uma rotação no eixo y, podemos fazer uma combinação de  $\sigma_x$  com  $\sigma_z$

$$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = i \cdot \sigma_x \cdot \sigma_z,$$

deve-se mencionar também a porta Hadamard, usada para causar o efeito de superposição no qubit

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|),$$

todas essas portas são unitárias e aplicadas sobre um único qubit, a próxima porta é um **NOT** controlado, denominada **CNOT**, aplicada sobre dois qubits

$$\mathbf{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = |00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 11| + |11\rangle\langle 10|.$$

### 2.1.1.3 Representação do Circuito Quântico e Esfera de Bloch

A representação convencional é demonstrada na figura 2, onde cada linha  $q[i]$  é um qubit, e cada bloco de iteração simboliza alguma operação sobre os qubits. E assim como num dispositivo real, é possível observar todas as operações sobre um ou mais qubits, porém o seu resultado final é obtido somente após a projeção realizada.

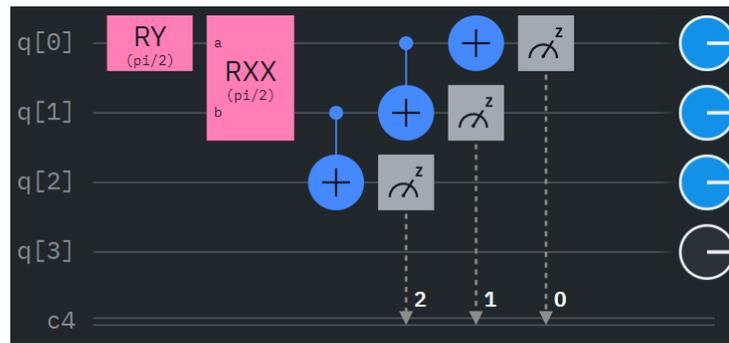


Figura 2 – Exemplo de representação de um circuito quântico, neste os blocos em rosa operações de rotação simples em Y e dupla em X, em azul as portas NOT sobre um qubit e CNOT aplicada sobre dois qubits, por fim os blocos em cinza as medições de cada qubit.

$$\begin{aligned}
 |0\rangle : \theta = 0, \varphi \text{ arbitrário} &\rightarrow \vec{r} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\
 |1\rangle : \theta = \pi, \varphi \text{ arbitrário} &\rightarrow \vec{r} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \\
 |+\rangle : \theta = \frac{\pi}{2}, \varphi = 0 &\rightarrow \vec{r} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\
 |-\rangle : \theta = \frac{\pi}{2}, \varphi = \pi &\rightarrow \vec{r} = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \\
 |+i\rangle : \theta = \frac{\pi}{2}, \varphi = \frac{\pi}{2} &\rightarrow \vec{r} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\
 |-i\rangle : \theta = \frac{\pi}{2}, \varphi = \frac{3\pi}{2} &\rightarrow \vec{r} = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}
 \end{aligned}$$

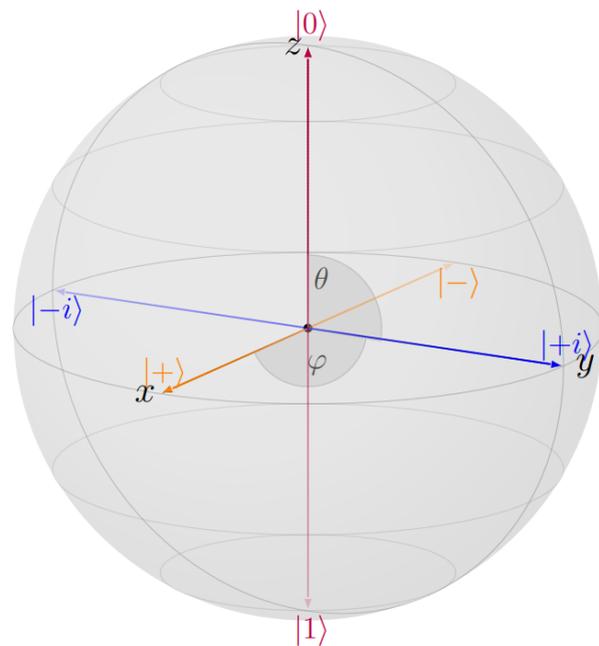


Figura 3 – Esfera de Bloch com todos os estados  $|0\rangle$ ,  $|1\rangle$ ,  $|+\rangle$ ,  $|-\rangle$ ,  $|+i\rangle$  e  $|-i\rangle$  puros representados.

Na esfera de Bloch tem-se uma representação tridimensional da fase de um qubit, que é onde está armazenada a sua informação quântica, na figura 3 pode-se observar os estados puros  $|0\rangle$ ,  $|1\rangle$ ,  $|+\rangle$ ,  $|-\rangle$ ,  $|+i\rangle$  e  $|-i\rangle$  que são as extremidades de cada dimensão. Em um qubit pode-se definir combinações desses estados puros, mas também ângulos diferentes nos eixos para realizar a combinação criando uma quantidade infinita de valores representados por um único qubit. A ideia de representar valores e de um vetor latente em um qubit é demonstrada em circuitos

quânticos de encoder [Benedetti et al. 2019], a arquitetura desse encoder pode variar em como serão representadas essas informações, mas seguem no intuito de traduzir a informação latente para a informação quântica. O encoder mencionado neste estudo na seção 3.1.1, foi a escolha dos autores da arquitetura PQWGAN [Tsang et al. 2023] para traduzir as informações.

## 2.1.2 Redes Neurais Adversárias

Na ideia original em [Goodfellow et al. 2014], temos um discriminador que é rival de um gerador, e ambos passam por treinamento com os dados reais. A estratégia é que, dado um vetor  $z$  amostrado utilizando alguma distribuição  $P_z$  (por exemplo, uma distribuição uniforme ou Gaussiana), o vetor  $z$  preenchido com ruído é então inserido em um gerador  $G$ , que produz um resultado. Objetivamente, compara-se a distribuição de  $P_{\text{dado}}$  com  $P_G$  verificando o índice de coincidência. Competitivamente, temos o discriminador  $D$ , que recebe como entrada  $x$  e, na sua saída, retorna a probabilidade de que acredita ser  $x$  um dado real ou não. Caso  $x$  seja lido como proveniente de  $P_z$  com probabilidade suficiente (previamente determinada como satisfatória) para entendê-lo como uma imagem do conjunto real de dados, então o discriminador  $D$  foi superado pelo gerador. O objetivo de  $G$  é produzir amostras que sejam aprovadas por  $D$ , e, equivalentemente, o objetivo de  $D$  é maximizar a sua probabilidade de assimilar corretamente a legenda correta para as amostras.

$$\min_G \max_D \mathbb{E}_{x \sim P_{\text{dado}}} [\log D(x)] + \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))] . \quad (2.1)$$

Na equação (2.1),  $\min_G$  representa a atuação do gerador buscando o menor valor para a equação, enquanto o discriminador em  $\max_D$  procura aumentar esse valor,  $\mathbb{E}_{x \sim P_{\text{dado}}}$  é a esperança matemática sobre a distribuição de dados reais, enquanto  $D(x)$  é a probabilidade de  $x$  ser classificado como uma imagem real, a função  $D(x)$  é colocada sobre o logaritmo  $[\log D(x)]$  para calcular o erro, na segunda parte da equação segue-se a mesma ideia para  $\mathbb{E}_{z \sim P_z}$  sobre a distribuição dos dados gerados, também será calculado o erro, mas de forma inversa com  $[\log(1 - D(G(z)))]$ , assim quanto melhor for  $G(z)$ , maior negativamente será o valor de  $\mathbb{E}_{z \sim P_z}$ , diminuindo o valor da equação e de forma adversária quanto melhor for o discriminador  $D$  maior será o valor da primeira parte da equação e menor o valor da segunda parte aumentando o valor final da equação.

## 2.1.3 Wasserstein GAN

As GAN's são acometidas de problemas durante o seu treinamento prático por problemas brevemente apresentados na seção 1.1, como gradiente desvanescente, modo colapso, e algumas deficiências em critérios de paradas [Arjovsky e Bottou 2017] [Arjovsky, Chintala e Bottou 2017] e também pode-se mencionar a divergência de Jensen-Shannon [Goodfellow et al. 2014]. A alternativa para mitigar esses problemas foi a utilização das WGANs (Wasserstein Generative

Neural Networks) reformulando a forma como o treino das GANs é feito, a WGAN é um dos frameworks que apresentam melhor sucessão, visando diminuir a distância de Wasserstein, a sua função é descrita como:

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{\mathbf{x} \sim P_{\text{dado}}} [D(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim P_z} [D(G(\mathbf{z}))]. \quad (2.2)$$

Diferente das GANs agora o valor de  $D$  pertence ao conjunto  $\mathcal{D}$  das funções 1-Lipschitz [Arjovsky, Chintala e Bottou 2017]. No entanto, a saída do discriminador de binária passa a ser a pontuação determinada como a distância de Wasserstein em  $P_G$  e  $P_{\text{dado}}$ . O discriminador sendo justamente o crítico, funciona melhor teoricamente quando a distância de Wasserstein é reduzida, resultados mostram que a técnica convergiu em muitos casos onde a divergência de JS calculando os erros falhou [Arjovsky, Chintala e Bottou 2017]. Essas estratégias implicam em resultados bem sucedidos com mais facilidade nas tentativas anteriores, ou seja, os treinamentos se tornaram mais fáceis de serem executados [Tsang et al. 2023]. A função da WGAN resulta na estimativa da distância de Wasserstein, que está empiricamente ligada à qualidade das amostras que se pode obter no gerador. Essa métrica pode então ser utilizada como condição de parada no treinamento da WGAN. Olhando novamente para as funções 1-Lipschitz, proposto por [Arjovsky, Chintala e Bottou 2017] definiram um parâmetro para cada crítico num intervalo fixo como  $[-0.01, 0.01]$ . Entretanto esse intervalo proposto cria novos problemas, por exemplo caso a magnitude do gradiente for grande, levaria um tempo muito grande até que o crítico se torne bom, e caso contrário isso poderia ocasionar o desaparecimento de gradientes. A alternativa apresentada por [Gulrajani et al. 2017] foi a de utilizar a WGAN-GP, sendo GP (gradient penalty), essa penalidade de gradiente implica então a restrição 1-Lipschitz. O valor da função d WGAN-GP é dado por:

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{\mathbf{x} \sim P_{\text{dado}}} [D(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim P_z} [D(G(\mathbf{z}))] - \lambda \mathbb{E}_{\hat{\mathbf{x}} \sim P_{\hat{\mathbf{x}}}} [(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2]. \quad (2.3)$$

Definindo as alterações temos então na terceira parte da equação (2.3), onde  $\lambda$  é o coeficiente de penalização da restrição 1-Lipschitz, o que justifica a escolha de  $D \in \mathcal{D}$  em  $\max_{D \in \mathcal{D}}$ , pois precisa-se trabalhar apenas com funções que sejam 1-Lipschitz,  $\lambda$  é o parâmetro que controla a intensidade da penalidade, a expectativa  $\mathbb{E}_{\hat{\mathbf{x}} \sim P_{\hat{\mathbf{x}}}} [(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2]$  onde  $\hat{\mathbf{x}}$  é a interpolação dos dados reais e gerados, é parte da restrição, para garantir a variação menor igual a 1, calcula-se então a norma quadrada do gradiente de  $D(\hat{\mathbf{x}})$ , e temos  $[(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2]$ , onde a subtração por 1 ao quadrado é a diferença ao quadrado completam os componentes da restrição 1-Lipschitz. O framework WGAN-GP foi aplicado com sucesso em várias aplicações de arquitetura DCGAN (Deep Convolutional Generative Adversarial Network) baseada em convoluções de profundidade, com diferentes funções de ativação, profundidade, contagem de filtros e normalização de batch. Fazendo uma comparação, os frameworks WGAN-GP tem um score mínimo de Inception [Salimans et al. 2016](métrica de quantidade, variedade e

qualidade dos resultados de uma GAN) treinando em uma quantidade muito maior em relação aos frameworks GAN para dados da ImageNet de 32x32 pixels.

## 2.2 Referencial Teórico e Trabalhos Relacionados

### 2.2.1 QGAN (Quantum Generative Adversarial Network)

Pode-se mencionar a teorização da QGAN em [Radford, Metz e Chintala 2016], e a eficiência satisfatória numérica em [Dallaire-Demers e Killoran 2018]. Uma característica já vantajosa da QGAN é a de que é possível trabalhar tanto com uma base de dados quântica quanto com uma base de dados clássica, em [Lloyd e Weedbrook 2018] é discutida a não garantia de vantagem quando se utiliza dados clássicos, mas assim como no algoritmo Harrow-Hassidim-Loyd [Harrow, Hassidim e Lloyd 2009] é esperado que de acordo com o a eficiência em resolver equações lineares a QGAN possa levar menos tempo para aprender a distribuição linear.

### 2.2.2 Geração de Imagens em QGANs

Considerando a era NISQ dos computadores quânticos, a tarefa de de um computador com  $n$  qubits para gerar um saída com dimensão  $M$ , exige algumas estratégias, [Huang et al. 2021] sugerem nessas condições estratégias de batch e patch para as QGAN's. Essa estratégia do patch favorece trabalhar com dimensões que obedecem a excedência da relação  $n < \lceil \log(M) \rceil$ , assim é possível gerar imagens em alta resolução em dispositivos NISQ. Para o patch, com  $k$  circuitos quânticos onde cada circuito quântico fica responsável por gerar um patch da imagem, isso é encapsulado como a parte geradora da QGAN, já o discriminador pode ser tanto um clássico como quântico. Somando os patches resultantes é possível formar a imagem. Neste modelo, tanto via simulações quanto em um computador quântico supercondutor, foi possível o treinamento para gerar e reconhecer imagens de escritas de um dataset [Dua e Graff 2017] contendo 0s e 1s com 8x8 pixels. Mesmo que a abordagem tenha obtido êxito na geração de imagens de escrita, 64 pixels de resolução resulta em imagens de pouca qualidade. Pode-se mencionar uma outra alternativa, denominada QuGAN [Stein et al. 2021], o framework em si funciona com swap test, medindo a fidelidade entre discriminador e gerador, e posteriormente foi incrementada uma extensão, o IQGAN [Chu et al. 2023], o framework recebeu um novo codificador de dados clássicos para quânticos e o gerador com circuitos quânticos simplificados e menos custosos, ou seja circuitos quânticos minimizados [Katsukawa 2022]. Tanto o QuGAN quanto o IQGAN rodam o conjunto de dados MNIST, que são comprimidos usando PCA. Os dois obtiveram êxito na geração de imagens, a ressalva é que na descompressão que também se usa PCA, o resultado final são imagens com mais resolução porém bastante borradas e sem muita definição. Inclusive a falta de definição nas imagens pode dificultar caso as classes geradas sejam mais variadas que duas classes “0” e “1” testadas, e que por exemplo tenham mais dígitos.

### 2.2.3 Quantum Wasserstein GANs

O framework WGAN também teve a sua versão inteiramente quântica proposta em [Chakrabarti et al. 2019] e [Kiani et al. 2022]. Seguindo a mesma ideia de utilizar a distância de Wasserstein como métrica, os trabalhos desenvolvem a estratégia utilizando circuitos quânticos parametrizados para gerar estados puros e mistos. Os trabalhos também mostram a eficiência do framework qWGAN em convergir rapidamente para o resultado ótimo. O adendo é de que o framework é desenvolvido para trabalhar diretamente com informação quântica, o que impõe uma restrição quando o objetivo é a geração de imagens, que é um dos motivos de interesse deste artigo. A extensão do framework WGAN que considera o gradient penalty, o WGAN-GP teve uma apresentação em uma versão híbrida clássica-quântica, denominada QWGAN-GP, o método de funcionamento é composto por um circuito quântico parametrizado que recebe um vetor latente como entrada, o restante do desenvolvimento é similar ao da WGAN-GP. Os experimentos feitos com o dataset de fraudes de cartão de crédito em [ULB 2016] mostram que em comparação o framework QWGAN-GP tem performance similar ao WGAN-GP na detecção de anomalias, porém, com uma quantidade menor de parâmetros de treinamento. O treinamento dataset de fraudes de cartão de crédito exige respostas simples, e como esperado, a estrutura clássica e a estrutura quântica convergem rapidamente para o resultado ótimo com um vetor latente de baixa dimensão e geradores de pouca profundidade.

### 2.2.4 Patch Quantum Wasserstein GANs

Proposto por [Tsang et al. 2023] o framework PQWGAN, constituído para gerar imagens de alta resolução em dispositivos NISQ. Herdando características dos frameworks mencionados anteriormente, como a técnica de patch para geração de imagens em dispositivos NISQ [Huang et al. 2021] e a metodologia do framework WGAN-GP. Retomando na equação (2.3), e explorando a aplicação da distância de Wasserstein aqui, o gerador  $G$  sempre procura o menor valor da distância, e, de forma antagônica o discriminador  $D$  procura sempre o maior valor. Note que é fundamental que o discriminador  $D$  classifique em  $\mathbb{E}_{x \sim P_{\text{dado}}}$  como verdadeiras as imagens pois se trata da distribuição dos dados reais, e que  $D(G(z))$  sejam classificadas como imagens falsas para que o valor de  $\mathbb{E}_{z \sim P_z}$  seja baixo, analiticamente o gerador  $G$  começa a vencer o jogo quando o valor em  $\mathbb{E}_{z \sim P_z}$  cresce, pois a distância de Wasserstein é dada pela diferença da esperança matemática de classificar como verdadeira a imagem da distribuição de dados reais com a expectativa de classificar como verdadeira a imagem da distribuição de dados gerados, assim a distância é menor quando os valores são aproximados. Como discutido na seção 2.1.3 a terceira parte da equação trata a restrição 1-Lipschitz, impedindo que o gradiente caminhe de forma discrepante durante o treino.

# 3

## Framework PQWGAN

Neste capítulo explora-se detalhadamente o framework PQWGAN com detalhes da arquitetura do framework, sua construção algorítmica e a sua rotina de treinamento e aquisição de imagens.

### 3.1 O Framework

Representada na figura 1, a arquitetura PQWGAN assim como em outras estratégias generativas utiliza o espaço latente adequado ao propósito, adequação essa que discutiremos posteriormente no procedimento de aquisição do vetor latente  $\mathbf{z}$ , esse vetor de ruídos é submetido ao gerador quântico sendo codificado para trabalhar nos sub-geradores parametrizados que formam o conjunto gerador da arquitetura, cada sub-gerador é responsável por gerar um patch da amostra gerada, assim que gerado o seu resultante é submetido ao crítico, que é proposto na forma clássica de um discriminador, o crítico resulta na distância de Wasserstein entre as expectativas de acerto discutidas na seção 2.1.3, após a avaliação o crítico envia os updates necessários para cada situação.

#### 3.1.1 Algoritmo do Gerador

A ideia algorítmica do gerador desenvolvida em [Tsang et al. 2023] é a de codificar pedaços do espaço latente, cada vetor é codificado em um estado quântico. Analisando em partes o algoritmo, para começar é pego um vetor latente de N-dimensões  $\mathbf{z} = (z_1, z_2, \dots, z_N)$  que é originado através de uma distribuição  $P_z$ , que pode ser uniforme, ou Gaussiana por exemplo. Cada elemento do vetor latente é codificado de forma parametrizada através de rotações  $R_Y$ . Começando no estado  $|0\rangle^{\otimes n}$  é obtido o estado  $|\mathbf{z}\rangle$ , a representação do circuito de encoder é dado por:

$$|z\rangle = R_Y^\dagger(z_1)R_Y^2(z_2)\dots R_Y^N(z_N)|0\rangle^{\otimes N}$$

para cada  $R_Y^\dagger(z_i)$  a porta  $R_Y$  é aplicada, com a angulatura de  $z_i$  ao  $i$ -ésimo qubit. Fazendo menção e considerando as características de eficiência da estrutura ansatz de eficiência de hardware [Kandala et al. 2017] que estipula uma estrutura baseada em PQC's para o encoder, que são idealmente rotações arbitrárias  $R(\phi, \theta, \omega)$  parametrizadas onde em cada qubit e seu adjacente é aplicada uma porta **CNOT** para adquirir o emaranhamento entre eles

$$R(\phi, \theta, \omega) = R_Z(\omega)R_Y(\theta)R_Z(\phi) = \begin{bmatrix} \exp\left(-\frac{i(\phi+\omega)}{2}\right) \cos\left(\frac{\theta}{2}\right) & \exp\left(-\frac{i(\phi-\omega)}{2}\right) \cos\left(\frac{\theta}{2}\right) \\ \exp\left(-\frac{i(\phi-\omega)}{2}\right) \cos\left(\frac{\theta}{2}\right) & \exp\left(-\frac{i(\phi+\omega)}{2}\right) \cos\left(\frac{\theta}{2}\right) \end{bmatrix}.$$

O objetivo em escolher esta porta  $R(\phi, \theta, \omega)$  e dado pela capacidade de representar qualquer rotação de um qubit-único desejado, e também a capacidade de aplicar uma mudança de fase. Além disso é intuitivo pensar na decomposição dessa porta na sequência de portas  $ZYZ$ , facilmente concebíveis em um dispositivo quântico real. Para que a escolha de  $L$  camadas seja feita, a representação parametrizada dessas mesmas  $L$  camadas podem ser traduzidas nessa grande única grande operação unitária  $U_L(\phi_i, \theta_i, \omega_i)$  que tem um comportamento linear sobre o estado  $|z\rangle$ , o seu resultante é o estado gerado pelo  $i$ -ésimo sub-gerador:

$$|\psi_{G_i}\rangle = U_L(\phi_i, \theta_i, \omega_i|z\rangle).$$

A estratégia que usa-se para o aprendizado de dados não lineares na entrada das redes neurais também foi projetada para o framework. Nos sub-geradores a não linearidade é aplicada através de uma medição parcial  $M$  sobre os qubits de ancilla, depois é feito o traço sobre os qubits de ancilla para obter o estado resultante dos qubits de dados. A medição é feita através da projeção, assim, o estado dos qubits de dados  $|\psi_D\rangle$  depois do traço nos qubits de ancilla é matematicamente dado por:

$$|\psi_D\rangle = Tr_A \left( \frac{M \otimes \mathbb{I} |\psi_{G_i}\rangle \langle \psi_{G_i}|}{\langle \psi_{G_i}| M \otimes \mathbb{I} |\psi_{G_i}\rangle} \right).$$

A medição parcial por simplicidade é setada como  $M = (|0\rangle\langle 0|)^{\otimes A}$ , assim o estado final dos qubits de dados fica:

$$\rho_D = Tr_A \left( \frac{(|0\rangle\langle 0|)^{\otimes A} \otimes \mathbb{I} |\psi_{G_i}\rangle \langle \psi_{G_i}|}{\langle \psi_{G_i}| (|0\rangle\langle 0|)^{\otimes A} \otimes \mathbb{I} |\psi_{G_i}\rangle} \right).$$

Aqui pode se observar que o estado atual depende de  $|\psi_{G_i}\rangle$  tanto no numerador quanto do numerador, e que por ordem de antecedência depende de  $|z\rangle$ . Seguindo, mede-se a probabilidade de cada base computacional do estado dos qubits de dados para saber o valor de saída de cada sub-gerador, é feito então:

$$G_i(\mathbf{z}) = [p(0), p(1), \dots, p(2^{D-1})].$$

A melhor opção após a representação de valores entre  $[0, 1]$  como valores para os pixels é identificar pesos diferentes para cada elemento, assim, a normalização não se torna problemática e podemos abdicar dos pixels indesejados fazendo o pós-processamento:

$$G'_i(\mathbf{z}) = \frac{G_i(\mathbf{z})}{\max(G_i(\mathbf{z}))}$$

seguindo essa ordem para obter somente os pixels válidos. Como segue-se a ideia de patches, as saídas dos circuitos quânticos são potências de 2, fazendo  $\frac{H \times W}{P}$  é selecionado o tamanho correto que essa saída deve ter para cada patch. Juntando a saída de cada sub-gerador podemos formar a imagem final de tamanho  $H \times W$

$$G(\mathbf{z}) = [G'_1(\mathbf{z}), G'_2(\mathbf{z}), \dots, G'_p(\mathbf{z})]^T.$$

Algumas outras maneiras de fazer as transformações lineares são discutidas em [Huang et al. 2021], mas para o caso de estudo essa foi a estratégia selecionada pelo autor do framework.

---

**Algorithm 1** Algoritmo para Gerar Imagens a Partir de um Patch Quantum Generator
 

---

**Input:** Imagens de dimensão  $H \times W$ , número de qubits de ancilla  $A$ , número de qubits de dados  $D$ , número de camadas sub-geradoras  $L$ , número de patches  $P$ , parametros do gerador  $\theta = [\theta_1, \dots, \theta_P]$ , variável latente  $\mathbf{z}$

**for**  $i = 1, \dots, P$  **do**

$|\psi_i\rangle \leftarrow U_{i,L,\theta_i}|\mathbf{z}\rangle$

$\rho_D \leftarrow \text{Tr}_A \left( \frac{|0\rangle\langle 0|^{\otimes A} \otimes \mathbb{I} |\psi_i\rangle\langle \psi_i|}{\langle \psi_i | |0\rangle\langle 0|^{\otimes A} \otimes \mathbb{I} | \psi_i \rangle} \right)$

$G_i(\mathbf{z}) \leftarrow [p(0), \dots, p(2^D - 1)]$  ▷ Medição  $\rho_D$  na base computacional

$G'_i(\mathbf{z}) \leftarrow \frac{G_i(\mathbf{z})}{\max(G_i(\mathbf{z}))}$

$G''_i(\mathbf{z}) \leftarrow G'_i(\mathbf{z}) \left[ 1 : \frac{HW}{P} \right]$  ▷ Descarta os valores excedentes dos pixels

**end for**

$G(\mathbf{z}) \leftarrow [G''_1(\mathbf{z}), \dots, G''_p(\mathbf{z})]^T$

**return**  $G(\mathbf{z})$

---

## 3.1.2 Algoritmo do Discriminador e Treino

### 3.1.2.1 Estrutura do Crítico

A escolha de usar a estrutura crítica semelhante a WGAN-GP em [Tsang et al. 2023] que é a estrutura da rede neural clássica foi feita observando que as redes neurais clássicas suportam a questão dos dispositivos NISQ, assim a estrutura clássica tem uma conformidade eficiente para a composição da arquitetura PQWGAN. Outro ponto é que o crítico recebe uma imagem e resulta um valor real para ser utilizado na distância de Wasserstein. Pensar nessa ideia para um circuito quântico traz complexidades como a quantidade de recursos necessários para efetuar essa tarefa, e

a de que o processo de aprendizado de circuitos quânticos ainda não são tão bem compreendidos em comparação com as redes neurais clássicas. Em [McClellan et al. 2018] podemos ver que se tratando de circuitos quânticos parametrizáveis a escolha de circuitos com muitos qubits faz com que o gradiente vá para qualquer direção diferente de zero em uma relação exponencialmente pequena como uma função da quantidade de qubits.

---

**Algorithm 2** PQWGAN Algoritmo de Treino
 

---

**Input:** Coeficiente de penalidade do gradiente  $\lambda$ , número de iterações do gerador  $n_C$ , número de epochs  $n_{epochs}$ , tamanho do batch  $m$ , hiperparâmetros Adam  $\eta_1, \eta_2, \beta_1, \beta_2$ .  
 inicializa os  $\omega$  parâmetros do crítico, e os  $\theta$  parâmetros dos sub-geradores

**for**  $epoch = 1, \dots, n_{epochs}$  **do**

**for**  $t = 1, \dots, n_C$  **do**

**for**  $i = 1, \dots, m$  **do**

      Amostragem de um dado real  $\mathbf{x} \sim \mathbb{P}_{data}$ , variável latente,  $\mathbf{z} \sim p_z$

$\mathbf{x}' \leftarrow quantum\_generator(\theta, \mathbf{z})$

$\hat{\mathbf{x}} \leftarrow \epsilon \mathbf{x} + (1 - \epsilon) \mathbf{x}'$

$L_D^{(i)} \leftarrow D(\mathbf{x}') - D(\mathbf{x}) + \lambda(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2$

**end for**

$\omega \leftarrow Adam\left(\nabla_{\omega} \frac{1}{m} \sum_{i=1}^m L_D^{(i)}, \omega, \eta_1, \beta_1, \beta_2\right)$

**for**  $i = 1, \dots, m$  **do**

      Amostragem da variável latente  $\mathbf{z} \sim p_z$

$\mathbf{x}' \leftarrow quantum\_generator(\theta, \mathbf{z})$

$L_G^{(i)} \leftarrow -D(\mathbf{x}')$

**end for**

**end for**

$\theta \leftarrow Adam\left(\nabla_{\theta} \frac{1}{m} \sum_{i=1}^m L_G^{(i)}, \theta, \eta_2, \beta_1, \beta_2\right)$

**end for**

---

### 3.1.2.2 Alvo do Treinamento

Algoritmicamente a diferença substancial do que é definido em (2.3) quando o framework PQWGAN é treinado em relação ao framework WGAN-GP é a de que a saída do gerador é originada de múltiplos sub-geradores, como visto e demonstrado na seção 3.1. O WGAN-GP é capaz de treinar várias arquiteturas diferentes com a necessidade apenas de alguns ajustes finos nos hiperparâmetros. Considerando a era NISQ, tendo em pauta também a escassez de recursos quânticos, a possibilidade de simular circuitos quânticos têm um impacto positivo considerável ao mesmo tempo que é muito difícil, tanto que a tarefa de encontrar hiperparâmetros se torna uma jornada demorada e exaustiva. Até que a execução de circuitos quânticos seja mais simples e eficiente, ajustar hiperparâmetros para circuitos maiores e complexos está fora do escopo. Outra diferença do framework PQWGAN com o WGAN-GP, é que o PQWGAN acrescenta os geradores quânticos que são descritos no algoritmo 2. Como a estratégia consiste em utilizar vários sub-geradores é necessário atualizar os parâmetros de cada um deles sobre uma perda que é calculada em cima de uma imagem inteira. A função de perda  $\mathcal{L}(\omega, \theta)$  depende tanto dos parâmetros do crítico quanto do gerador. Crítico e gerador são treinados de

forma alternada, mas esse processo alternado faz parte do treinamento do PQWGAN. Fixando o crítico ele é treinado, e vice-versa para o gerador, para calcular o gradiente também foi usada a “parameter shift rule” regra de deslocamento do gradiente em [Schuld et al. 2019] que permite calcular os gradientes de um circuito quântico evitando simulações repetidas vezes. Com  $N_G$  sub-geradores com  $n$  parâmetros em cada, expressados esses parâmetros em um vetor tal que  $\theta = [\theta_1, \dots, \theta_{N_G}] = [\theta_{1,1}, \dots, \theta_{1,n}, \dots, \theta_{N_G,1}, \dots, \theta_{N_G,n}]$ . Assim, o gradiente do  $j$ -ésimo parâmetro do  $i$ -ésimo sub-gerador em relação à perda é dado por:

$$\frac{\partial \langle \mathcal{L}(\omega, \theta) \rangle}{\partial \theta_{i,j}} = \frac{1}{2} \left( \langle \mathcal{L}(\omega, [\theta_{1,1}, \dots, \theta_{i,j} + \frac{\pi}{2}, \dots, \theta_{N_G,n}]) \rangle - \langle \mathcal{L}(\omega, [\theta_{1,1}, \dots, \theta_{i,j} - \frac{\pi}{2}, \dots, \theta_{N_G,n}]) \rangle \right).$$

# 4

## Metodologia

### 4.1 PQWGAN Metodologia de Experimentação

#### 4.1.1 Dataset e Bibliotecas

Foram utilizados os mesmos datasets MNIST [LeCun, Cortes e Burges n.d.] e FMNIST [Xiao, Rasul e Vollgraf 2017] para a experimentação. O MNIST é um dataset simples de amplo reconhecimento para o treinamento de GANs clássicas, constituído por imagens de dígitos de 0 à 9 (10 classes diferentes) de escrita manual com resolução de 28x28 pixels em escala de cinza e com um set de treino de 60 mil exemplos e um set de teste de 10 mil exemplos. O Fashion-MNIST é similar MNIST, a principal diferença são as suas classes que invés de dígitos, temos 10 classes de peças de roupas distintas (camisetas, calças, moletoms, vestidos, jaquetas, sandálias, camisas, tênis, bolsas, botas), com a mesma quantidade de exemplos 60 mil para treino e 10 mil para testes, na mesma escala de cinzas e na mesma resolução 28x28 pixels. Embora os dois datasets utilizados tenham quantidades satisfatórias de exemplos para o aprendizado de GAN's, devido a limitação de recursos serão utilizados apenas 1000 exemplos de cada classe escolhida para o set de treino, escolha que será justificada na seção 5.1.

#### 4.1.2 Implementação e Ambiente de Execução

O algoritmo é fornecido pelos autores em [Tsang et al. 2023], originalmente implementado em Python3 utilizando as bibliotecas PyTorch [Paszke et al. 2019] e PennyLane [Bergholm et al. 2022]. A biblioteca PyTorch é utilizada para desenvolver algoritmos de aprendizado de máquina de alta performance. Por vez a biblioteca PennyLane é utilizada para o desenvolvimento de QML (Quantum Machine Learning), que possibilita a simulação quântica de algoritmos utilizando um computador clássico. A biblioteca PennyLane também tem capacidade de comunicar como uma interface com a biblioteca PyTorch. Diferente dos autores da PQWGAN os testes feitos neste

estudo foram realizados em uma máquina com processador AMD Ryzen 5600X com 6 núcleos físicos e 12 threads, 32GB de memória RAM, e uma GPU NVIDIA RTX 3070 Ti com 8GB de VRAM. Justamente pensando no diferente cenário de execução de testes realizados pelos autores, algumas alterações foram feitas no código original para melhor aproveitamento do hardware e possibilitar também tanto a execução de testes na GPU e CPU da máquina ao mesmo tempo.

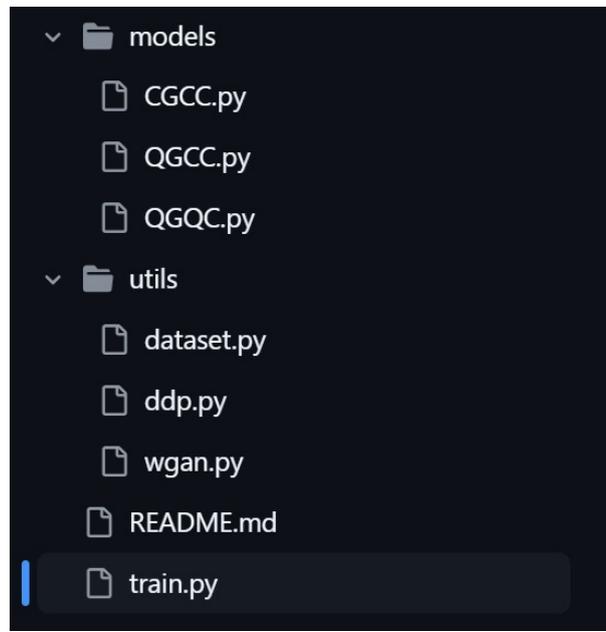


Figura 4 – Estrutura de implementação PQWGAN Master

O arquivo principal de treino `train.py` tem alguns apoios, começando pelos modelos que são *CGCC* (Classical Generator - Classical Critic), *QGCC* (Quantum Generator - Classical Critic) e *QGQC* (Quantum Generator - Quantum Critic) que são referenciados pelo treino dependendo da necessidade. Caso a chamada de prompt faça referência a alguma arquitetura específica híbrida ou pura, ela será acionada pelo treino. Os outros apoios são `ddp.py` para a distribuição paralela de processamento das rotinas e tarefas, `dataset.py` para interfacear o treino com os datasets que novamente serão selecionados entre *MNIST* ou *FMNIST* dependendo da escolha na chamada de prompt, e o último apoio `wgan.py` onde é calculado o gradiente de penalização da Wasserstein GAN. É importante mencionar as alterações feitas na implementação original, já que o objetivo aqui é executar as simulações de um computador quântico em uma máquina clássica convencional diferente do ambiente computacional adotado em [Tsang et al. 2023]. E que a intenção em alterar o código era paralelizar o máximo de tarefas durante os treinos visando o uso de todos os recursos da máquina utilizada antes que a mesma atingisse os seus limites físicos quando estivesse executando um experimento, essa estratégia de utilizar a maior quantidade possível de recursos da uma mesma máquina para experimentar diferentes cenários em paralelo foi adotada visando acelerar o processo de aquisição de resultados.

### 4.1.2.1 Arquivo train.py

Analisando a implementação do treino, viu-se que era possível alterar o local em que a execução ocorreria, no trecho:

```
16 def train(classes_str, dataset_str, patches, layers, n_data_qubits, batch_size,
17 out_folder, checkpoint, randn, patch_shape, qcritic):
18     classes = list(set([int(digit) for digit in classes_str]))
19     device = torch.device("cpu")
```

As bibliotecas PyTorch e PennyLane são compatíveis com GPU's NVIDIA que possuem a tecnologia CUDA [Nvidia 2007], assim disponibilizando a unidade de processamento gráfico para executar as simulações a máquina utilizada teria mais uma unidade de processamento para a execução dos experimentos. Foi feita a primeira alteração na implementação para adequar ao ambiente:

```
16 def train(classes_str, dataset_str, patches, layers, n_data_qubits, batch_size,
17 out_folder, checkpoint, randn, patch_shape, qcritic):
18     classes = list(set([int(digit) for digit in classes_str]))
19     device = torch.device("cuda" if torch.cuda.is_available() else "cpu") #Seleciona
    cuda como o dispositivo
```

Essa alteração permite verificar se há um dispositivo CUDA disponível na máquina, em caso afirmativo o dispositivo selecionado será o primeiro disponível. Após essa alteração durante as primeiras tentativas de execução foram constatados erros com os tensores do código implementado, onde alguns tensores de execução estavam alocados em dispositivos diferentes. O que levou a verificação de tensores que ainda não haviam sido corretamente referenciados para o funcionamento em dispositivos CUDA caso fosse possível.

### 4.1.2.2 Arquivo QGCC.py

Por ordem de testes o arquivo do modelo híbrido foi um dos que precisou de alterações, dentro da função `forward` e logo depois na função `partial_trace_and_postprocess`:

```
42 def forward(self, x):
43     x = x.to(next(self.parameters()).device) #Move x para o dispositivo correto

51 output_images = torch.Tensor(x.size(0), 0).to(x.device) #Move x para o dispositivo
    correto

53 for sub_generator_param in self.params:
54     patches = torch.Tensor(0, pixels_per_patch).to(x.device) #Move x para o
    dispositivo correto

62 if special_shape:
63     final_out = torch.zeros(x.size(0), *self.image_shape).to(x.device) #Move x para
    o dispositivo correto
```

```

70 else:
71     final_out = output_images.view(output_images.shape[0], *self.image_shape).to(x.
       device) #Move x para o dispositivo correto
72 return final_out

```

Foi necessário certificar que `x` estava sendo processado no dispositivo correto para evitar qualquer inconsistência

```

95 return post_processed_patch.to(next(self.parameters()).device) #Certifica que o
    patch processado esteja no dispositivo correto

```

No caso da função `partial_trace_and_postprocess` o tensor em questão foi o patch que estava sendo pós processado.

### 4.1.2.3 Arquivo QGQC.py

De forma semelhante às alterações do modelo híbrido, o modelo totalmente quântico também teve vários tensores certificados para executar no dispositivo selecionado lá no documento de treino em `device = torch.device("cuda" if torch.cuda.is_available() else "cpu")`, as alterações realizadas no arquivo foram:

```

25 def forward(self, x):
26     x = x.view(x.shape[0], -1).to(next(self.parameters()).device) #Move x para o
    dispositivo correto

```

```

48 def forward(self, x):
49     x = x.to(next(self.parameters()).device) #Move x para o dispositivo correto

```

```

56 output_images = torch.Tensor(x.size(0), 0).to(x.device) #Certifica que
    output_images esteja no dispositivo correto

```

```

58 for sub_generator_param in self.params:
59     patches = torch.Tensor(0, pixels_per_patch).to(x.device) #Certifica que patches
    esteja no dispositivo correto

```

```

60 for item in x:
61     sub_generator_out = self.partial_trace_and_postprocess(item, sub_generator_param
    ).float().unsqueeze(0).to(x.device) #Certifica que sub_generator_out esteja no
    dispositivo correto

```

```

67 if special_shape:
68     final_out = torch.zeros(x.size(0), *self.image_shape).to(x.device) #Certifica
    que final_out esteja no dispositivo correto

```

```

75 else:
76     final_out = output_images.view(output_images.shape[0], *self.image_shape).to(x.
       device) # Certifica que final_out esteja no dispositivo correto

```

```

99 return post_processed_patch.to(next(self.parameters()).device) #Certifica que o
    patch processado esteja no dispositivo correto

```

O modelo inteiramente clássico `CGCC.py` não precisou de nenhuma alteração, pois nenhum tensor foi alocado de forma equivocada durante os testes, ou seja, a configuração inicial no arquivo de treino foi suficiente para o funcionamento desse modelo.

#### **4.1.2.4 Arquivo `wgan.py`**

Foi feita uma reestruturação nesse trecho de código quando estava na fase de mitigação de erros para o funcionamento do algoritmo completo no ambiente de GPU's utilizando CUDA, para certificar que qualquer tensor ou cálculo estivesse sendo endereçado para o dispositivo correto assim que selecionado no arquivo `train.py`. Em avaliações posteriores, descartou-se qualquer problema estrutural em `wgan.py` na adaptação realizada. No entanto, esse e os demais arquivos adaptados para funcionar no ambiente CUDA estão disponíveis no apêndice deste trabalho.

# 5

## Experimentação e Resultados

Neste capítulo é feita uma comparação com a aquisição de resultados que foram produzidos pelo autor deste estudo com os resultados experimentados em [Tsang et al. 2023]. A ideia principal aqui é verificar a plausibilidade de reproduzir esses resultados mesmo que em um ambiente diferente, e também testar como as redes neurais generativas quânticas podem se relacionar desde resultados sintéticos em supercomputadores até aplicações reais em máquinas pessoais.

### 5.1 Observações

Na seção 4.1.1 vimos a escolha de apenas 1000 exemplos de cada classe para o treinamento com framework, devido a limitações de recursos computacionais essa opção foi tomada para evitar a demora exaustiva de aquisição de resultados, tanto pelos autores do PQWGAN quanto para o decorrer deste trabalho.

É importante reiterar que todos os resultados adquiridos durante este estudo são sinalizados em diferenciação com os resultados de comparação. Outra observação importante é a de que somente resultados do framework PQWGAN foram reproduzidos neste trabalho por questões que envolvem foco na geração quântica de imagens, tempo de experimentação e recursos computacionais. Sendo assim nenhuma aquisição de resultados do framework WGAN-GP foi realizada para esta sintetização de resultados.

### 5.2 Geração de Imagem com o framework PQWGAN

Como produto final do PQWGAN, nesta seção são expostos alguns experimentos com atenção para a capacidade de geração de imagens do framework. Os primeiros resultados visuais são encontrados na figura 5, onde encontra-se amostras coletadas pelos autores em [Tsang et al. 2023] e em seguida na “Reprodução PQWGAN” as amostras dos experimentos feitos para este

trabalho.

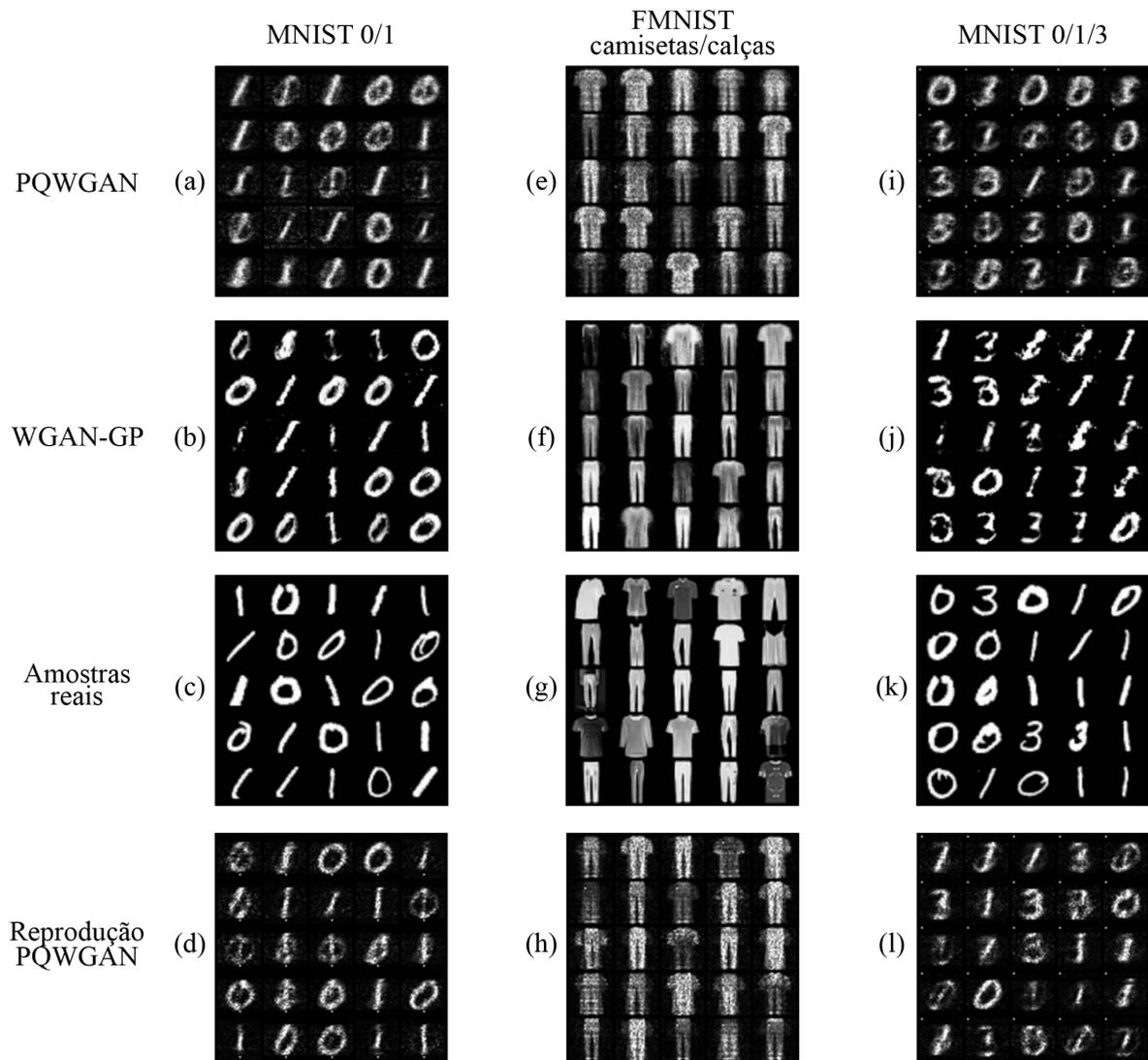


Figura 5 – *Imagens geradas de forma randômica, na primeira linha os resultados dos autores da PQWGAN, na segunda linha temos os resultados da WGAN-GP, na terceira linha as amostragens reais dos datasets e na quarta linha tem-se os resultados adquiridos neste estudo na reprodução dos experimentos.*

Também foram reproduzidas as plotagens da distância de Wasserstein na figura 6 para motivos de comparação seguindo a mesma lógica de comparar dados dos autores com a reprodução realizada na figura 5.

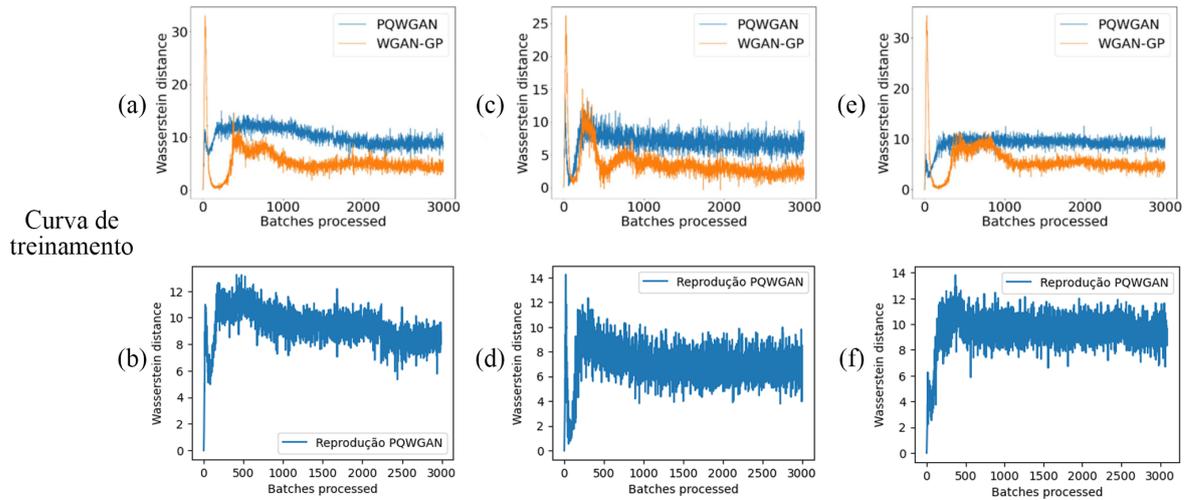


Figura 6 – Plotagem das distâncias de Wasserstein, (a), (c) e (e) resultado dos autores da PQWGAN com MNIST-0/1, FMNIST-camisetas/calças e MNIST-0/1/3 respectivamente e (b), (d) e (f) a plotagem das distâncias de Wasserstein que adquiriu-se com a reprodução dos experimentos na mesma ordem respectiva anterior.

### 5.2.1 MNIST binário

A primeira coluna da figura 5 é baseada na aquisição de amostras do dataset MNIST com as classes 0 e 1. Nesta primeira tarefa cada sub-gerador tem sete camadas, com sete qubits de dados e um qubit de ancilla com a finalidade de conservar recursos. Quando vê-se na figura 5 os itens (a), (b) e (d), a percepção de que para ambos os casos quântico e clássico, os geradores estão aprendendo com sucesso a gerar imagens de 0s e 1s. Na figura 6 os itens (a) e (b) também mostram como a é tendência da distância de Wasserstein diminuindo com o passar dos batches processados, o que se induz a pensar na melhora das imagens conforme o treinamento avança. Porém, por inspeção visual ainda é possível detectar falhas que se distinguem das amostras reais. Um comportamento em particular no caso quântico que pode ser observado no item (a) linha 3 e coluna 3 e no item (d) linha 2 coluna 5 da figura 5, de que as amostras parecem ao mesmo tempo ser 0 e 1 como se fossem uma soma. Esse tipo de falhas podem ser atribuídas ao processo incompleto de aprendizagem, quando o gerador ainda não aprendeu de forma abrangente um mapeamento para todo o espaço latente. Reiterando que a situação de simulações quânticas são limitadas, 600 iterações do gerador são uma quantidade significativamente menor que a ordem de  $10^4 \sim 10^5$  iterações de um gerador clássico, mesmo que para uma tarefa mais complexa. O comportamento de produzir amostras que parecem uma combinação de diferentes clássicas não é exclusiva do PQWGAN, esse comportamento também é presente no WGAN-GP, a sugestão é de que esse problema pode ser mitigado treinando mais o gerador para ambos os casos. Outro problema que pode ser observado no gerador quântico é uma espécie de neblina nas imagens.

Ao mesmo tempo que as imagens parecem satisfazer a sua classe, como no item (d) da figura 5 na linha 1 coluna 2, que as bordas não são nítidas. Lembrando que os valores dos pixels no gerador quântico se originam das amplitudes do estado quântico no final do circuito. Então para que um pixel seja completamente escuro, a amplitude correspondente deve ser 0, e pensando na natureza de um circuito quântico emaranhado, uma amplitude igual a 0 é um estado muito difícil de se obter na ocasião. Esse efeito da amplitude diferente de 0 ainda é amplificado na etapa de pós-processamento, quando as amplitudes são normalizadas para 0 ou 1. A vantagem de nitidez então é colocada sobre os geradores clássicos, ainda mais quando se deseja alterar um valor específico de um pixel, pois isso em informação quântica significaria mudar a angulação e fase de um qubit. Falando de performance, visualmente a alternativa quântica PQWGAN atua de forma equiparável a alternativa clássica WGAN-GP, isso reforça a ideia da força expressiva dos PQCs [Du et al. 2020] superar as redes neurais clássicas. Olhando para a arquitetura WGAN-GP, o gerador tem em torno de 1,46 milhão de parâmetros treináveis, enquanto para a arquitetura PQWGAN o gerador tem 5376 parâmetros treináveis. Pensando no produto de um gerador clássico com uma quantidade semelhante de parâmetros, não seria possível aprender nada praticável tal como o gerador quântico.

### 5.2.2 FMNIST binário

Aumentando a complexidade da tarefa, agora experimentando com imagens de camisetas e calças do dataset FMNIST. A metodologia para este experimento foi de agora onze camadas visando o aumento da complexidade dos dados, os mesmos sete qubits de dados e um qubit de ancilla, na primeira análise visual tudo parece ocorrer bem. Retornando a figura 5, nos itens (e), (f) e (h) é possível observar novamente, tanto para o caso quântico quanto para o caso clássico que os geradores podem aprender com sucesso a gerar imagens das classes de camisetas e calças. E também podemos observar a persistência de problemas como borrões e amostras incertas. Porém é possível notar como no framework quântico o gerador tenta aprender até detalhes mais sutis, variando os tons das amostras, amostrando as peças com espessuras diferentes, detalhes esses que são facilmente observados nos itens (e) e (h) da figura 5. Mais uma aderência para apoiar o potencial do futuro dos geradores quânticos. Na figura 6 itens (c) e (d), observando a curva de treinamento, a distância de Wasserstein diminui de forma quase imperceptível, e com uma taxa de variação bem alta a cada atualização. Sugestivamente o gerador está chegando bem próximo de sua capacidade e mesmo assim não está conseguindo aprender uma representação com possibilidade de capturar todas as variações em detalhes das amostras, pois chegou no seu limite de expressividade. Esse problema poderia ser corrigido com o aumento do número de camadas em cada sub-gerador para aumentar a expressividade do gerador. Também é possível aumentar o tamanho do batch do processo de aprendizado, assim obteria-se gradientes com maior estabilidade. Essas alterações, entretanto, aumentariam o custo computacional de simulação, aumentando significativamente a quantidade de recursos necessários para simular, o que tornaria inviável em tempo prático esse experimento para este trabalho.

### 5.2.3 MNIST triplo

Para este experimento utilizou-se a mesma metodologia do FMNIST binário, onze camadas, sete qubits de dados e um qubit de ancilla, já que agora em comparação com o MNIST binário foi aumentado em uma classe o aprendizado que o gerador deve seguir. No experimento do FMNIST binário foi possível observar a dificuldade que o framework quântico teve em aprender detalhes mais complexos por causa da sua capacidade expressiva. A tarefa aqui será diferenciar entre os dígitos 0, 1 e 3 que têm características distintas na sua escrita. Os resultados tanto no gerador clássico quanto no gerador quântico mostraram-se capazes de produzir imagens correspondentes aos três dígitos determinados. Pode-se observar no entanto que no framework PQWGAN existem artefatos persistentes no mesmo local em todas as imagens. O comportamento ainda não completamente conhecido é facilmente manipulado através de mudanças nos parâmetros, como número de qubits de dados no gerador quântico, reduzindo-o. Deixando de lado os artefatos, ambos os frameworks, clássico e quântico são capazes de aprender e gerar imagens de 0, 1 e 3. Como esperado, a maior complexidade do conjunto de dados em comparação com o MNIST binário também implicou na maior quantidade de imagens mistas. A distância de Wasserstein na figura 6 itens (e) e (f) está estabilizando, de forma que se sugere que a capacidade do gerador está chegando no seu limite. Assim como já discutido anteriormente, uma solução seria aumentar a quantidade de camadas e também um processo de treinamento mais longo.

# 6

## Conclusão

Boa parte do universo que se conhece sobre a geração de imagens com inteligência artificial é estabelecida por GANs clássicas. Quando comparamos as GANs clássicas com as QGANs vê-se uma promissora caminhada para o futuro das redes neurais generativas quânticas potencializado pelo desenvolvimento da computação quântica. Mesmo limitada pela era NISQ, a pesquisa de QML pode hoje impulsionar a utilização de dispositivos futuros com mais recursos quânticos, tal como o framework PQWGAN que aprimora a execução de uma tarefa de forma escalável. De forma análoga à computação paralela clássica, a ideia de separar tarefas em patches para processá-las de maneira paralela não somente aumenta a capacidade de processar como também a complexidade de processamento. Pode-se considerar que ainda existem duas barreiras a serem quebradas no desenvolvimento de QGANs, a primeira em como o ruído dos dispositivos atuais afetou a geração de imagens causando artefatos quando principalmente quando a complexidade dos PQCs era incrementada, a segunda de que por mais que seja possível simular um dispositivo quântico com grande poder computacional para efetivar as simulações, a quantidade de recursos em dispositivos quânticos reais atuais ainda está e seu processo inicial de estruturação. Como a complexidade em simular um dispositivo quântico em uma máquina clássica é bem alta, algoritmos que conseguem escalar sua robustez à medida que os recursos quânticos se tornam palpáveis apresentam-se de forma muito bem vinda ao cenário atual e futuro do estudo de redes neurais quânticas. Apesar das adversidades a serem superadas, o favoritismo dos PQC e das QGANs em relação a sua tomada clássica são dados pela sua força de aprendizado, como na seção 5.2.1 quando comparado o framework WGAN-GP com o framework PQWGAN, onde respectivamente os parâmetros treináveis eram de aproximadamente 1,46 milhão e 5376, diferença essa dada para a tarefa no qual os dois frameworks apresentaram resultados equiparáveis. Deste estudo é possível concluir que as redes neurais generativas quânticas oferecem um potencial gigantesco para a evolução da GANs num geral. Pensar na evolução aprendizado de máquinas quântico é também pensar em como o cenário do aprendizado de máquinas pode se tornar mais robusto e em como produzir redes neurais pode se tornar muito mais energeticamente eficiente.

# Referências

AMIN, M. H. et al. Quantum boltzmann machine. *Phys. Rev. X*, American Physical Society, v. 8, p. 021050, May 2018. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevX.8.021050>>. Citado na página 12.

ARJOVSKY, M.; BOTTOU, L. *Towards Principled Methods for Training Generative Adversarial Networks*. 2017. Disponível em: <<https://arxiv.org/abs/1701.04862>>. Citado 2 vezes nas páginas 12 e 19.

ARJOVSKY, M.; CHINTALA, S.; BOTTOU, L. *Wasserstein GAN*. 2017. Disponível em: <<https://arxiv.org/abs/1701.07875>>. Citado 3 vezes nas páginas 12, 19 e 20.

ASSOUEL, A.; JACQUIER, A.; KONDRATYEV, A. *A Quantum Generative Adversarial Network for distributions*. 2021. Disponível em: <<https://arxiv.org/abs/2110.02742>>. Citado na página 13.

BARENCO, A. et al. Elementary gates for quantum computation. *Physical Review A*, American Physical Society (APS), v. 52, n. 5, p. 3457–3467, nov. 1995. ISSN 1094-1622. Disponível em: <<http://dx.doi.org/10.1103/PhysRevA.52.3457>>. Citado na página 16.

BENEDETTI, M. et al. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, IOP Publishing, v. 4, n. 4, p. 043001, nov 2019. Disponível em: <<https://dx.doi.org/10.1088/2058-9565/ab4eb5>>. Citado 3 vezes nas páginas 12, 15 e 19.

BERGHOLM, V. et al. *PennyLane: Automatic differentiation of hybrid quantum-classical computations*. 2022. Disponível em: <<https://arxiv.org/abs/1811.04968>>. Citado na página 28.

BIAMONTE, J. et al. Quantum machine learning. *Nature*, Nature Publishing Group UK London, v. 549, n. 7671, p. 195–202, 2017. Citado na página 12.

CHAKRABARTI, S. et al. *Quantum Wasserstein Generative Adversarial Networks*. 2019. Disponível em: <<https://arxiv.org/abs/1911.00111>>. Citado na página 22.

CHU, C. et al. *IQGAN: Robust Quantum Generative Adversarial Network for Image Synthesis On NISQ Devices*. 2023. Disponível em: <<https://arxiv.org/abs/2210.16857>>. Citado 2 vezes nas páginas 13 e 21.

DALLAIRE-DEMERS, P.-L.; KILLORAN, N. Quantum generative adversarial networks. *Phys. Rev. A*, American Physical Society, v. 98, p. 012324, Jul 2018. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevA.98.012324>>. Citado 2 vezes nas páginas 13 e 21.

DU, Y. et al. Expressive power of parametrized quantum circuits. *Phys. Rev. Res.*, American Physical Society, v. 2, p. 033125, Jul 2020. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevResearch.2.033125>>. Citado 2 vezes nas páginas 12 e 36.

DUA, D.; GRAFF, C. *UCI machine learning repository*. 2017. <<http://archive.ics.uci.edu/ml>>. Citado na página 21.

EMAMI, H. et al. *SPA-GAN: Spatial Attention GAN for Image-to-Image Translation*. 2020. Disponível em: <<https://arxiv.org/abs/1908.06616>>. Citado na página 11.

- FEYNMAN, R. P. Simulating physics with computers. In: *Feynman and computation*. [S.l.]: cRc Press, 2018. p. 133–153. Citado na página 12.
- GOODFELLOW, I. J. et al. *Generative Adversarial Networks*. 2014. Disponível em: <<https://arxiv.org/abs/1406.2661>>. Citado 2 vezes nas páginas 11 e 19.
- GULRAJANI, I. et al. *Improved Training of Wasserstein GANs*. 2017. Disponível em: <<https://arxiv.org/abs/1704.00028>>. Citado na página 20.
- HARROW, A. W.; HASSIDIM, A.; LLOYD, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, American Physical Society, v. 103, p. 150502, Oct 2009. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.103.150502>>. Citado na página 21.
- HAVLÍČEK, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature*, Nature Publishing Group, v. 567, n. 7747, p. 209–212, 2019. Citado na página 12.
- HUANG, H.-L. et al. Experimental quantum generative adversarial networks for image generation. *Phys. Rev. Appl.*, American Physical Society, v. 16, p. 024051, Aug 2021. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevApplied.16.024051>>. Citado 4 vezes nas páginas 13, 21, 22 e 25.
- KANDALA, A. et al. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *nature*, Nature Publishing Group, v. 549, n. 7671, p. 242–246, 2017. Disponível em: <<https://doi.org/10.1038/nature23879>>. Citado na página 24.
- KATSUKAWA, M. Computação quântica: Minimização de circuitos. UFMS, 2022. Citado na página 21.
- KIANI, B. T. et al. Learning quantum data with the quantum earth mover’s distance. *Quantum Science and Technology*, IOP Publishing, v. 7, n. 4, p. 045002, jul 2022. Disponível em: <<https://dx.doi.org/10.1088/2058-9565/ac79c9>>. Citado na página 22.
- LECUN, Y.; CORTES, C.; BURGESS, C. J. C. *The MNIST database*. n.d. Disponível em: <<http://yann.lecun.com/exdb/mnist/>>. Citado na página 28.
- LI, J.; TOPALOGLU, R.; GHOSH, S. *Quantum Generative Models for Small Molecule Drug Discovery*. 2021. Disponível em: <<https://arxiv.org/abs/2101.03438>>. Citado na página 13.
- LLOYD, S.; MOHSENI, M.; REBENTROST, P. *Quantum algorithms for supervised and unsupervised machine learning*. 2013. Disponível em: <<https://arxiv.org/abs/1307.0411>>. Citado na página 12.
- LLOYD, S.; WEEDBROOK, C. Quantum generative adversarial learning. *Phys. Rev. Lett.*, American Physical Society, v. 121, p. 040502, Jul 2018. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.121.040502>>. Citado 2 vezes nas páginas 13 e 21.
- MCCLEAN, J. R. et al. Barren plateaus in quantum neural network training landscapes. *Nature communications*, Nature Publishing Group UK London, v. 9, n. 1, p. 4812, 2018. Disponível em: <<https://doi.org/10.1038/s41467-018-07090-4>>. Citado na página 26.
- NIELSEN, M. A.; CHUANG, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. [S.l.]: Cambridge University Press, 2010. Citado na página 16.

NVIDIA, C. Compute unified device architecture programming guide. 2007. Citado na página 30.

PASZKE, A. et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. Disponível em: <<https://arxiv.org/abs/1912.01703>>. Citado na página 28.

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, ago. 2018. ISSN 2521-327X. Disponível em: <<https://doi.org/10.22331/q-2018-08-06-79>>. Citado na página 12.

RADFORD, A.; METZ, L.; CHINTALA, S. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016. Disponível em: <<https://arxiv.org/abs/1511.06434>>. Citado 2 vezes nas páginas 11 e 21.

REBENTROST, P.; MOHSENI, M.; LLOYD, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.*, American Physical Society, v. 113, p. 130503, Sep 2014. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.113.130503>>. Citado na página 12.

REED, S. et al. *Generative Adversarial Text to Image Synthesis*. 2016. Disponível em: <<https://arxiv.org/abs/1605.05396>>. Citado na página 11.

SALIMANS, T. et al. *Improved Techniques for Training GANs*. 2016. Disponível em: <<https://arxiv.org/abs/1606.03498>>. Citado na página 20.

SCHULD, M. et al. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A*, American Physical Society, v. 99, p. 032331, Mar 2019. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevA.99.032331>>. Citado na página 27.

SHOR, P. Algorithms for quantum computation: discrete logarithms and factoring. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. [S.l.: s.n.], 1994. p. 124–134. Citado na página 12.

SOUSA, R. S. Computação quântica - algoritmo de shor. UFMS, 2020. Citado na página 12.

STEIN, S. A. et al. Qugan: A quantum state fidelity based generative adversarial network. In: *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. [S.l.: s.n.], 2021. p. 71–81. Citado 2 vezes nas páginas 13 e 21.

TSANG, S. L. et al. Hybrid quantum-classical generative adversarial network for high-resolution image generation. *IEEE Transactions on Quantum Engineering*, Institute of Electrical and Electronics Engineers (IEEE), v. 4, p. 1–19, 2023. ISSN 2689-1808. Disponível em: <<http://dx.doi.org/10.1109/TQE.2023.3319319>>. Citado 10 vezes nas páginas 13, 14, 19, 20, 22, 23, 25, 28, 29 e 33.

ULB, M. L. G. *Credit card fraud detection*. 2016. <<https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>>. Citado na página 22.

VONDRICK, C.; PIRSIYAVASH, H.; TORRALBA, A. *Generating Videos with Scene Dynamics*. 2016. Disponível em: <<https://arxiv.org/abs/1609.02612>>. Citado na página 11.

- WEST, M. T. et al. Benchmarking adversarially robust quantum machine learning at scale. *Phys. Rev. Res.*, American Physical Society, v. 5, p. 023186, Jun 2023. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevResearch.5.023186>. Citado na página 12.
- WEST, M. T.; SEVIOR, M.; USMAN, M. Boosted ensembles of qubit and continuous variable quantum support vector machines for b meson flavor tagging. *Advanced Quantum Technologies*, v. 6, n. 10, p. 2300130, 2023. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/qute.202300130>. Citado na página 12.
- WEST, M. T.; SEVIOR, M.; USMAN, M. Reflection equivariant quantum neural networks for enhanced image classification. *Machine Learning: Science and Technology*, IOP Publishing, v. 4, n. 3, p. 035027, aug 2023. Disponível em: <https://dx.doi.org/10.1088/2632-2153/acf096>. Citado na página 12.
- WEST, M. T. et al. Towards quantum enhanced adversarial robustness in machine learning. *Nature Machine Intelligence*, Nature Publishing Group UK London, v. 5, n. 6, p. 581–589, 2023. Citado na página 12.
- WIEBE, N.; KAPOOR, A.; SVORE, K. *Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning*. 2014. Disponível em: <https://arxiv.org/abs/1401.2142>. Citado na página 12.
- XIAO, H.; RASUL, K.; VOLLGRAF, R. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 2017. Disponível em: <https://arxiv.org/abs/1708.07747>. Citado na página 28.
- ZOUFAL, C.; LUCCHI, A.; WOERNER, S. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Information*, Nature Publishing Group UK London, v. 5, n. 1, p. 103, 2019. Citado na página 13.

# **Apêndices**

# APÊNDICE A – Disponibilidade de arquivos

*Disponibilidade do Código:* O código alterado para funcionar em dispositivos CUDA caso estejam disponíveis para executar as simulações pode ser encontrado em <<https://github.com/gusta-reis/PQWGAN-modificado>>.

*Conflitos de Interesse Financeiro:* Assim como os autores do framework experimentado declaram não ter nenhum conflito de interesse financeiro ou não financeiro segue-se a mesma intuição qualquer alteração para contribuição com os estudos.