UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Dipartimento di Fisica ”Giuseppe Occhialini”
Corso di Laurea Magistrale in Fisica

Tesi di Laurea Magistrale

Study of Feynman Diagrams with Quantum Graph Neural Network

Candidato:
Lorenzo Ballerio
Matricola 838676

Relatori:
Prof. Andrea Giachero
Dr. Michele Grossi

Correlatori:
Dr. Roberto Moretti
Prof. Pietro Govoni
Dr. Danilo Labranca

Anno Accademico 2022-2023

Sintesi

Nell'ultimo decennio, Machine Learning (ML) e Deep Learning (DL) hanno acquisito un ruolo sempre più rilevante in diverse aree di ricerca, tra cui la fisica fondamentale. Molti modelli di apprendimento automatico trovano impiego nel campo dell'analisi dati e nella modellizzazione di problemi complessi, sia in fisica fondamentale che applicata, grazie all'efficienza nell'estrarre informazione direttamente dai dati di partenza. Uno dei rami di ML più innovativi e di maggior attrattiva nell'ultimo decennio è quello del Quantum Machine Learning (QML), il quale coniuga Machine Learning e Quantum Computing con l'obiettivo di sfruttare i principi della meccanica quantistica, dunque la manipolazione di stati multi-qubit, per sviluppare nuovi modelli di apprendimento che possano integrare o sostituire quelli classici.

Questo lavoro di tesi si pone come obiettivo la caratterizzazione di modelli di QML applicati alla Fisica Teorica, in particolare il calcolo del modulo quadro delle ampiezze di scattering $|\mathcal{M}|^2$ per processi di elettrodinamica quantistica (QED). Dallo sviluppo perturbativo in teoria quantistica dei Campi (QFT) è noto come all'aumentare della complessità del processo da studiare (es. scattering adronici) o dal grado di precisione nel calcolo della sezione d'urto (correzioni ad ordini perturbativi superiori) il numero di diagrammi di Feynman cresce molto rapidamente; da ciò deriva un'evidente difficoltà nell'individuare tutti i diagrammi di Feynman per un determinato scattering e soprattutto calcolarne le relative ampiezze modulo quadro. Per far fronte a questo problema, tecniche di regressione possono essere utilizzate per stimare le ampiezze modulo quadro associate ai diversi diagrammi di Feynman.

In questo lavoro sono stati sviluppati e testati modelli di regressione quantistici, ossia modelli che hanno l'obiettivo di apprendere, tramite un processo di addestramento supervisionato, una mappa tra i dati in input ed una funzione continua in output. Esempi di modelli quantistici di regressione sono le reti neurali quantistiche, definite come circuiti quantistici composti da una *Feature Map*, che codifica le features del dato di input tramite trasformazioni che agiscono su stati multi-qubit, e da un *Ansatz Parametrico* (ovvero un *Circuito Quantistico Parametrico*) i cui parametri vengono ottimizzati tramite algoritmi classici (ad esempio algoritmi di discesa del gradiente).

In particolare, il modello implementato in questo lavoro di tesi prende il nome di *Quantum Graph Neural Network*, ossia una rete neurale quantistica specializzata nell'elaborare strutture a grafo. La scelta di questo tipo di QNN prende ispirazione dalla struttura dei diagrammi di Feynman, la cui topologia è informazione essenziale per determinare l'ampiezza finale del processo.

In questa tesi si affronta il problema di codificare i differenti canali di scattering tramite la scelta di Feature Map appropriate. In particolare, a ciascun nodo del grafo viene assegnato un qubit, mentre ai vertici vengono assegnate connessioni tra i qubit corrispondenti.

L'innovativo approccio alla regressione di ampiezze di scattering presentata in questo studio è di carattere tuttora preliminare. Ci si limita pertanto allo scattering al secondo ordine della teoria perturbativa in Elettrodinamica Quantistica (QED), per scopi dimostrativi.

Si è dunque passati a valutare le performance della QGNN, la quale è stata allenata su tre differenti diagrammi di Feynman: il canale-s e il canale-t dello scattering Bhabha e lo scattering $e^-e^+ \longrightarrow \mu^+\mu^-$, sia nel limite di QED ad alte energie (nel quale è possibile trascurare la massa delle particelle) che nel limite a basse energie (nel quale la massa va considerata).

Il dataset consiste in un set di variabili cinematiche corrispondenti ad un certo processo e l'ampiezza di scattering modulo quadro ad esso relativa, ottenuta tramite calcolo esatto. La QGNN sviluppata risulta essere molto performante nella predizione di ampiezze di scattering con andamento sinusoidale, per cui si trova per il canale-s dello scattering Bhabha un errore quadratico medio (MSE) pari a 1.830×10^{-9} (errore relativo del 0.30%) nel caso ad alte energie, e pari a 1.686×10^{-7} (errore relativo medio del 2.35%) nel caso di basse energie; per quanto riguarda invece lo scattering $e^+e^- \longrightarrow \mu^+\mu^-$, studiato considerando i leptoni massivi, l'MSE è pari a 4.331×10^{-7} (errore relativo medio del 3.97%).
In seguito è stata studiata la capacità predittiva della QGNN per un dataset misto contenente entrambi i canali dello scattering Bhabha nel limite ad alte energie; in particolare si è analizzato come la rete non sia in grado di distinguere correttamente le differenti topologie dei due diagrammi.
Nonostante l'ottima performance della QGNN per il canale-s infatti si è analizzato come la QGNN non sia in grado di imparare $|\mathcal{M}|^2$ del canale-t; tale difficoltà è dovuta alla presenza di un asintoto, presente sia nel limite ad alte energie che per basse energie; inoltre sembra che il modello non migliori di performance nonostante il tentativo di ottimizzare la sua struttura, aumenando ulteriormente i gradi di libertà del modello.
A conclusione del lavoro di tesi si è affrontato infine il problema dell'estrapolazione di termini di interferenza per processi a due canali, costruendo un circuito quantistico che combinasse due QGNN addestrate separatamente. Si dimostra però che dalla QGNN non si può ricavare l'ampiezza di scattering $\mathcal{M}$ in modo univoco, il che impedisce di calcolare il termine di interferenza fra due diagrammi di Feynman.
Questo progetto di tesi ha mostrato come una Quantum Graph Neural Network sia in grado di predire $|\mathcal{M}|^2$ partendo dal corrispondente diagramma di Feynman; tale modello però non è stato in grado di combinare efficacemente più diagrammi simultaneamente per descrivere un processo che ammettesse più di un diagramma. Nella parte finale dell'elaborato si mostra l'incapacità del modello di calcolare il termine di interferenza di due diagrammi di Feynman, sfruttando le proprietà dei circuiti quantistici.
Questo lavoro costituisce un punto di partenza per lo studio delle interazioni fondamentali utilizzando tecniche di Quantum Machine Learning.
L'attività di tesi è stata svolta parzialmente all'estero presso il gruppo *CERN Quantum Technologies Initiative (QTI)* di Givevra (CH) durante uno periodo di Mobilità internazionale avvenuta tramite un programma Exchange Extra UE.

Abstract

In the last decade, Machine Learning (ML) and Deep Learning (DL) have gained an increasingly significant role in various research areas, including fundamental physics. Many machine learning models are employed in the field of data analysis and modeling complex problems, both in fundamental and applied physics, due to their efficiency in extracting information directly from the initial data. One of the most innovative and attractive branches of ML in the last decade is Quantum Machine Learning (QML), which combines Machine Learning and Quantum Computing with the aim of exploiting the principles of quantum mechanics, such as the manipulation of multi-qubit states, to develop new learning models that can integrate or replace classical ones.

The objective of this thesis work is to characterize QML models applied to Theoretical Physics, particularly the calculation of the modulus squared of scattering amplitudes $|\mathcal{M}|^2$ for quantum electrodynamics (QED) processes. In quantum field theory (QFT), it is known that as the complexity of the process under study (e.g., hadronic scattering) or the degree of precision in cross-section calculations (higher-order corrections) increases, the number of Feynman diagrams grows very rapidly. This leads to an evident difficulty in identifying all the Feynman diagrams for a specific scattering and, especially, calculating their respective modulus squared. To face this problem, regression techniques can be used to estimate $|\mathcal{M}|^2$ associated with different Feynman diagram.

In this work, quantum supervised regression models have been developed and tested; these are models that, through a supervised training process, learn a mapping between input data and a continuous output function. Examples of quantum regression models include quantum neural networks, defined as quantum circuits composed of a *Feature Map*, that encodes input data features through transformations acting on multi-qubit states, and a *Parametric Ansatz* (a Parametric Quantum Circuit) whose parameters are optimized using classical algorithms (e.g., gradient descent algorithms).

In particular the model implemented in this thesis is called *Quantum Graph Neural Network*, which is a quantum neural network specialized in processing graph-structured data. The choice of this type of QNN is inspired by the structure of Feynman diagrams, whose topology is essential information for determining the final amplitude squared of the scattering process.

In this thesis, encoding different scattering channels is faced through the definition of appropriate Feature Maps. In particular, in our model it is exploited a correspondence bewteen graph's nodes and qubits and graph's edges and qubits connnections.

The innovative approach to scattering amplitude regression presented in this study is still in a preliminary stage. Therefore, it is limited to second-order scattering in perturbative Quantum Electrodynamics (QED) for demonstrative purposes.

The performance of the Quantum Graph Neural Network (QGNN) was then evaluated; the model was trained on three different QED Feynman diagrams: the Bhabha scattering s-channel and t-channel and the $e^+e^- \longrightarrow \mu^-\mu^+$ scattering, both in the high-energy limit (where particle masses can be neglected) and in the low-energy limit (where mass must be considered).

The dataset consists of a set of kinematic variables corresponding to a specific process, while the modulus squared of the associated scattering amplitude is obtained through exact calculation.

The developed QGNN proves to be very performing in predicting scattering amplitudes with a

sinusoidal behavior. For the s-channel of Bhabha scattering at high energies, it achieves a mean squared error (MSE) of 1.83010^{-9} (and a relative error of 0.30%), while it predicts $|\mathcal{M}|^2$ for low energies with a MSE about 1.686×10^{-7} (corresponding to a relative error f 2.35%); on the other hand for the $e^-e^+ \longrightarrow \mu^+\mu^-$ scattering dataset, which was studied considering massive leptons, the MSE is 4.331×10^{-7} (average relative error equal to 3.97%).

Afterwards, the predictive capacity of the QGNN was studied for a mixed dataset containing both Bhabha scattering channels in the high-energy limit. In particular, it was analyzed how the network is unable to correctly distinguish the different topologies of the two diagrams.

Despite the excellent performance of the QGNN for the s-channel, it was observed that the QGNN is unable to learn $|\mathcal{M}|^2$ for the t-channel. This difficulty is due to the presence of an asymptote in both the high-energy and low-energy limits. Furthermore, it appears that the model does not improve in performance despite attempts to optimize its structure by increasing the degrees of freedom.

In conclusion of the thesis, it was studied a quantum circuit that, by combining two separately trained QGNNs, it was able to extrapolate the interference term of two Feynman diagrams. However, it was demonstrated that the QGNN cannot uniquely retrieve the scattering amplitude $\mathcal{M}$, preventing the calculation of the interference term between the two diagrams.

This thesis project has shown how a Quantum Graph Neural Network is capable of predicting $|\mathcal{M}|^2$ based on the corresponding Feynman diagram. However, the model was unable to effectively combine multiple diagrams simultaneously to describe a process that allows for multiple diagrams. In the final part of the work it was demonstrated the model's inability to calculate the interference term between two Feynman diagrams using quantum properties.

This work is as a starting point for the study of fundamental interactions using Quantum Machine Learning techniques. The thesis work was conducted partly abroad at the CERN Quantum Technologies Initiative (QTI) group in Geneva (CH) during an international mobility period of the Exchange Extra UE program.

Contents

Chapter 1

Introduction to Quantum Computing

During this chapter we are introducing the basic concepts of quantum computing and quantum information theory and their applications for real-world problems. One of the most promising applications of quantum computing are undoubtedly artificial intelligence (A.I.) techniques, namely *Quantum Machine Learning* algorithms, which will be our topic of interest during all the thesis. The main goal of the project is indeed the realization of *Quantum Neural Network (QNN)* for Feynman diagram analysis.

1.1 Basic Concepts

While in classical computing the elementary piece of information consist of a *bit*, a logical deterministic state that can either assume value 0 or 1, it is possible to define its quantum counterpart, namely *qubit*; it is formally defined as a two-level quantum system defined in a Hilbert space $\mathcal{H} \simeq \mathbb{C}^2$; the most general state of such a system is described by the superposition of the so called computational basis $\{|0\rangle, |1\rangle\}$ of $\mathcal{H}$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad with \quad \alpha, \beta \in \mathbb{C} \quad and \quad |\alpha|^2 + |\beta|^2 = 1 \tag{1.1}$$

Such a constraint on both α and β derives from the probabilistic interpretation of quantum mechanics, which state that the coefficients α and β of the state (1.1) are related to the probabilities of obtaining state $|0\rangle$ or state $|1\rangle$ after a measure:

$$\begin{aligned} P(|0\rangle) = |\alpha|^2 \quad P(|1\rangle) = |\beta|^2 \\ \Longrightarrow P_{tot} = P(|0\rangle) + P(|1\rangle) = |\alpha|^2 + |\beta|^2 = 1 \end{aligned} \tag{1.2}$$

for definition of probability. This constraint is called normalization of the quantum state, and it is a fundamental principle of all quantum systems.

It is worth noting that more than one elementary piece of information exists: in fact we can also define a three-level quantum system, called *qutrit* as the superposition of the corresponding computational basis $\{|-1\rangle, |0\rangle, |1\rangle\}$. Nowadays the most common way to encode information into quantum computers are qubits, which represent a good analogy with classical way to store information.

One of the first and biggest differences between bits and qubits is the access to information we have in both cases; in fact in classical computers we can always have full access to the information

stored in the state of the bit, either state 0 or 1, and that makes very easy to store any kind of data (from single numbers to more complex structures such as strings, matrices, photos etc. ...) in a large set of bits and manipulate it with multi-bit operations.
The qubit as defined in (1.1) can exist in a continuum of states between $|0\rangle$ and $|1\rangle$, which means it contains, in principle, infinite information but, even though this is in principle true, this assertion is misleading, in fact we have access only to a fraction of it because of the behavior of a qubit when observed [1].
From the fundamental principles of quantum mechanics we know that it is impossible to get the full-knowledge of a quantum state (which means to determine both $|\alpha|^2$ and $|\beta|^2$) because of the Born's rule, which states that after a measurement the system collapses in the measured state, which means a loss of information of the initial state. A measurement of qubit (1.1) will give only either 0 or 1 and, in the final analysis, the accessible information amounts to only a single bit, and all the information about α and β is lost; since α and β are related to the probability of measuring the corresponding state, we can in principle recover $|\alpha|^2$ and $|\beta|^2$ by repeating $N \to \infty$ times the same measurement over N identical states, but this is not always possible because of the $no-cloning$ theorem:

Theorem 1. *Given a generic unknown quantum state of unit norm $|\psi\rangle$, there is no unitary operator U such that:*

$$|\psi\rangle \otimes |\phi\rangle \longrightarrow U(|\psi\rangle \otimes |\phi\rangle) = |\psi\rangle \otimes |\psi\rangle \quad \forall|\psi\rangle \tag{1.3}$$

Proof. Aassume there exists such a unitary operator U and two different unknown states $|\psi_1\rangle, |\psi_2\rangle$ for which holds that:

$$\begin{aligned} |\psi_1\rangle \otimes |\phi\rangle \xrightarrow{U} |\phi_1\rangle \otimes |\phi_1\rangle \\ |\psi_2\rangle \otimes |\phi\rangle \xrightarrow{U} |\phi_2\rangle \otimes |\phi_1\rangle \end{aligned} \tag{1.4}$$

then if I take the product state:

$$\begin{aligned} &\big(\langle\psi_2| \otimes \langle\phi|\big)\big(|\psi_1\rangle \otimes |\phi\rangle\big) \xrightarrow{U} \big(\langle\psi_2| \otimes \langle\psi_2|\big)\big(|\psi_1\rangle \otimes |\psi_1\rangle\big) \\ &\langle\psi_2|\psi_1\rangle\langle\phi|\phi\rangle \xrightarrow{U} \big(\langle\psi_2|\psi_1\rangle\big)^2 \end{aligned} \tag{1.5}$$

recalling that $\langle\phi|\phi\rangle = 1$ (normalized state) we find the following equation:

$$\begin{aligned} \langle\psi_2|\psi_1\rangle = \big(\langle\psi_2|\psi_1\rangle\big)^2 \\ \langle\psi_2|\psi_1\rangle\big(1 - \langle\psi_2|\psi_1\rangle\big) = 0 \end{aligned} \tag{1.6}$$

The solutions of such an equation are for $\langle\psi_2|\psi_1\rangle = 0, 1$ which means that either the two stases are orthogonal or $|\psi_2\rangle = e^{i\alpha}|\psi_1\rangle$, but in both cases $|\psi_1\rangle$ and $|\psi_2\rangle$ are not indipendent. □

All these considerations make extracting information from quantum systems a very tricky issue.

To fully introduce the concept of measurement is necessary to introduce the concept of observable in quantum mechanics.

Definition 1. *An observable in quantum mechanics is an hermitian operator $A : \mathcal{H} \longrightarrow \mathcal{H}$, such that $A^\dagger = A$, where $\mathcal{H}$ is the Hilbert space of the system.*

[1] so when we perform a measurement of the system

Since the observable by definition is an hermitian operator it has a set of real eigenvalues $\{\lambda_i\}$ and has a set of orthonormal eigenstates $\{|i\rangle\}$, with $i = 1, ..., dim(\mathcal{H})$. Then it is always possible to rewrite a generic state as a superposition of eigenstates of A: $|\psi\rangle = \sum_{i=1}^{dim(\mathcal{H})} \alpha_i |i\rangle, \quad \alpha_i \in \mathbb{C}$. For a single qubit the most generic observable is defined as:

$$A = \alpha_0 \mathbb{1} + \vec{\alpha} \cdot \vec{\sigma}, \quad \alpha_0, \vec{\alpha} \in \mathbb{R} \tag{1.7}$$

as a linear combination of the identity matrix and the Pauli matrices:

$$\begin{aligned}
\sigma_z &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
\sigma_x &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad |+\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad |-\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\
\sigma_y &= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad |+i\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}, \quad |-i\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}
\end{aligned} \tag{1.8}$$

1.1.1 Bloch Sphere and Multi Qubit Systems

In this subsection, we are presenting further explanations about the qubit and how systems of more qubits behave.

Recalling the most general definition of a qubit (1.1):

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \alpha, \beta \in \mathbb{C} \quad s.t. \quad |\alpha|^2 + |\beta|^2 = 1 \tag{1.9}$$

From group theory it is known that $\mathbb{C}^2 \simeq \mathbb{R}^4$, so it is possible to replace two complex degrees of freedom with four real ones, which reduce to three after the normalization constraint; so we can rewrite the state $|\psi\rangle$ as:

$$|\psi\rangle = e^{i\gamma} \left[cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\rho} sin\left(\frac{\theta}{2}\right)|1\rangle \right], \quad with \quad \gamma, \theta, \rho \in \mathbb{R} \tag{1.10}$$

In particular we note that angle γ acts as a global phase, which is irrelevant in quantum mechanics because we always have to deal with expectation values of observables over states, so this factor cancels out; we then say that quantum mechanics is invariant under global $U(1)$ transformations, and such an invariance, from Noether's theorem, implies the existence of a conserved quantity P such that $\frac{dP}{dt} = 0$, which is the probability of the system. Such a definition of a qubit has a geometrical interpretation, because we can think of the angles ρ and θ as respectively the radial and polar coordinates of a unitary sphere in $3D$ called *Bloch sphere* 1.1: all the *pure* states of a qubit are hence points of the sphere' surface $S^2 \subset \mathbb{R}^3$.

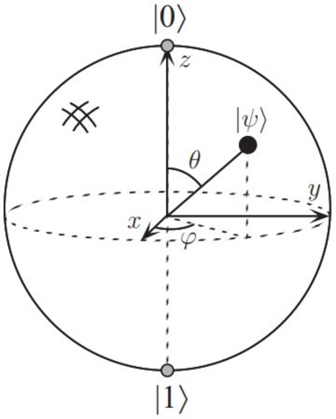


Figure 1.1: Definition of Bloch' sphere: a pure quantum state is represented of a point in S^3

We previously introduced, without any further investigation, the concept of a pure quantum state, which is defined as followed:

Definition 2. *a pure state $|\psi\rangle$ is a quantum system which is known exactly (with classical probability $p = 1$).*

A state which is a collection of pure states in the ensable $\{p_i, |\psi_i\rangle\}$, with p_i classical probability of the state $|\psi_i\rangle$, is called a mixed state.

While dealing with mixed states or interacting physical systems, instead of working with state vectors representing the quantum system (quantum-state picture), it is more convenient to introduce the concept of *density matrix*:

Definition 3. *Given an ensemble of quantum pure states and their associated probabilities $\{p_i, |\psi_i\rangle\}$, we can define the density matrix operator as:*

$$\rho \equiv \sum_i p_i |\psi_i\rangle\langle\psi_i| \tag{1.11}$$

for a pure state we have $p_j = 1$ and $p_i = 0$ $\forall i \neq j$ the density matrix operator is simply $\rho = |\psi_j\rangle\langle\psi_j|$.

Any matrix operator ρ must satisfy the following properties:

- *ρ is hermitian and positive*
- *$Tr[\rho] = 1$*
- *$Tr[\rho^2] < 1$ for mixed states, $Tr[\rho^2] = 1$ for pure states*

Here follows the proofs of the properties for a density matrix operator ρ; here we will use a mixed state without loss of generality.

Proof.

- hermitianity and positiveness:

$$\begin{aligned}\rho^\dagger = \Big(\sum_i p_i|\psi_i\rangle\langle\psi_i|\Big)^\dagger = \sum_i p_i\Big(|\psi_i\rangle\langle\psi_i|\Big)^\dagger = \sum_i p_i|\psi_i\rangle\langle\psi_i| = \rho \\ \langle\phi|\rho|\phi\rangle = \sum_i p_i|\langle\phi|\psi_i\rangle|^2 \geq 0 \quad \forall|\phi\rangle\end{aligned} \tag{1.12}$$

- non-unitary trace of ρ:

$$Tr[\rho] = \sum_i p_i Tr[|\psi\rangle\langle\psi_i|] = \sum_i p_i Tr[\langle\psi_i|\psi_i\rangle] = \sum_i p_i = 1 \tag{1.13}$$

- unitary trace of ρ^2:

$$Tr[\rho^2] = \sum_i p_i^2 Tr[|\psi_i\rangle\langle\psi_i|] = \sum_i p_i^2 Tr[\langle\psi_i|\psi_i\rangle] = \sum_i p_i^2 < 1 \tag{1.14}$$

for pure states we have that $p_j = 1$, $p_i = 0 \quad \forall i \neq j$, then $\sum_i p_i^2 = p_j^2 = 1$.

□

By this definition it is possible to write the most generic form of a density matrix ρ in terms of Pauli matrices:

$$\rho = a_0 \mathbb{1} + \vec{a}\vec{\sigma}, \tag{1.15}$$

If we impose to (1.15) all the conditions a density matrix must satisfy we will come up with:

$$\begin{aligned} &Tr[\rho] = 2a_0 = 1 \Longrightarrow a_0 = \frac{1}{2} \quad and \quad \vec{a} = \frac{\vec{r}}{2} \\ &\rho = \frac{\mathbb{1} + \vec{r}\cdot\vec{\sigma}}{2}, \quad \vec{r} \in \mathbb{R} \\ &Tr[\rho^2] = \frac{1}{4}Tr\left[(\mathbb{1} + \vec{r}\cdot\vec{\sigma})^2\right] = \frac{1}{4}Tr[\mathbb{1} + 2\vec{r}\cdot\vec{\sigma} + |\vec{r}|^2] = \frac{1+|\vec{r}|^2}{2} \leq 1 \end{aligned} \tag{1.16}$$

from which follows that in general for a pure state we have to impose $|\vec{r}| = 1$, whereas for a mixed state it holds $|\vec{r}| < 1$. The geometrical interpretation on the Bloch sphere is then straightforward: a pure state lies on the surface S^2 of the sphere, whereas a mixed state lives in the internal points of the sphere.

In the *Schrödinger* picture we know that a pure state evolves in time with respect to Schrodinger equation: $\psi(t)\rangle = \hat{U}(t)|\psi(0)\rangle$, where $\hat{U}(t) = e^{-\frac{i}{\hbar}\hat{H}t}$[2] and $\hat{H}$ Hamiltonian of the system. So we can define the evolution law for a density matrix ρ as:

$$\rho(t) = \sum_i p_i |\psi_i(t)\rangle\langle\psi_i(t)| = \sum_i p_i U(t)|\psi_i(0)\rangle\langle\psi_i|U^\dagger(t) = U(t)\rho(0)U^\dagger(t) \tag{1.17}$$

Density matrix and state-vector are different formalisms of the same theory, and whether to use one or the other depends on the problem itself (indeed as we said density matrix formalism suits well for mixed states or real gates problems). For our purposes we will deal with only pure states and ideal gates, so we are working on state-vector simulation of quantum circuit.

Quantum computers are made of a collection of n qubits, so it is very important to understand how to deal with multi-qubits systems.
A system of n qubits lives in an Hilbert space which is the tensor product between all the n single qubits' Hilbert spaces, namely $\mathcal{H}_{tot} = \mathcal{H}^n \simeq \mathbb{C}^{2n}$, so the generic state of $\mathcal{H}_{tot}$ is a linear combination of the computational basis $\{|i\rangle\}$, with $i = 0, ..., 2^n - 1$. A n-qubit quantum state can then encode information for all the eigenstates of the computational basis at once; this can seem a big advantage over the classical case, but it is misleading because we always have to take into account that by performing a measurement of the system in order to extract information, the system collapse and most of the information is lost.
In multi qubit systems some interesting quantum properties emerge, such as quantum entanglement:

[2] the system evolves with respect to a unitary operator U because of the global $U(1)$ symmetry and its related probability conservation; in fact by definition unitary operators preserve the norm of vectors, hence they preserve probability.

Definition 4. *a n-qubit state is called entangled if it can not be decomposed as the tensor product of n single qubit states; namely:*

$$|\psi\rangle \neq |\phi_1\rangle \otimes |\phi_2\rangle \otimes ... \otimes |\phi_{n-1}\rangle \otimes |\phi_n\rangle \tag{1.18}$$

Entanglement is one of the most interesting and important consequences of a quantum theory, which leads to unique properties of quantum systems and quantum information protocols that can be exploited in quantum information theory.
An interesting consequence of entanglement is the EPR paradox [1], formulated in 1935 by *Einstein*, *Podolsky* and *Rosen*, in which they stated that quantum mechanics was an incomplete theory because it violates the concept of *reality*[3].

In this paper E-P-R stated that the two main properties a complete theory must satisfy are the principle of *reality* and *locality*, defined as follow:

Definition 5. *Reality: If, without in any way disturbing a system, we can predict with certainty (i.e., with probability equal to unity) the value of a physical quantity, then there exists an element of physical reality corresponding lo this physical quantity.*

Locality: an object is influenced directly only by its immediate surroundings. A theory that includes the principle of locality is said to be a local theory.

In the following lines is presented the core of the EPR paradox in the Bohm's variant[2]. Assume an electron-positron pair in the singlet state:

$$|\psi\rangle = \frac{|0_A 1_B\rangle - |1_A 0_B\rangle}{\sqrt{2}}, \tag{1.19}$$

where $|0\rangle, |1\rangle$ are eigenstates of Pauli-Z matrix $\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, with eigenvalues respectively $\sigma_z|0\rangle = +1|0\rangle$ and $\sigma_z|1\rangle = -1|1\rangle$.

We take these two particles very far one from the other and we perform one measure, for example of σ_z^A, on particle A. Suppose that the result of such a measurement is the state $|0_A\rangle$; from the Born's rule we know that the system right after the measurement falls into the state:

$$|\psi\rangle \xrightarrow{\sigma_z^A} |0_A 1_B\rangle \tag{1.20}$$

what we see is that by determining a measurement over particle A is then determined with certainty the state (and then the result of the measurement σ_z^B) of particle B. At first look it might seems a violation of *locality*, because of a sort of superluminal signal propagating from the two particles, but it is a false paradox: in fact the $no-communication$ theorem states that no classical bits of information is transmitted instantaneously right after the collapse of the entangled-state function after a measurement.
The kernel of the paradox is indeed an apparent violation of reality, so the fact that it is impossible to know *a priori* the values of all the physical quantities of a system[4] without ambiguity.

We take again the initial state (1.19) and perform the same procedure shown before, where the global state falls into $|\psi\rangle \xrightarrow{\sigma_z^A} |0_A 1_B\rangle$. It was shown that if we measure σ_z^B we know with certainty the z-spin state of B, but if we measure σ_x^B we can get either $|+\rangle$ or $|-\rangle$ (eigenstates of σ_x) with probability 50%: so the values for the observable σ_x is not a priori exactly determinable (because $[\sigma_z, \sigma_x] \neq 0$), and that is a violation of reality. E-P-R suggested that this sort of violation was due to the fact that QM was an incomplete theory and supposed that there

[3] In [1] E-P-R stated *reality* is one of the two properties a complete theory must satisfy together with *locality*.

[4] we note that this is strictly related to the Heisenberg uncertainty principle due to non commuting observables.

were some missing variables to the theory that, if they were taken into account (the so called $hidden-variables\ theory$), would make quantum mechanics a complete theory.
All the discussion about EPR paradox was solved in 1964 by Bell [3], who defined a set of inequalities (the so called Bell's inequalities) to verify whether the interpretation of the real world of quantum mechanics was right or not; many different experiments (i.e. [4, 5, 6]) and many other variants of the Bell's inequalities were formulated, and was proven that quantum mechanics is indeed a complete theory which violates the principle of *local realism*.

Entanglement is the key of very interesting protocols that can be exploited in quantum information and computing theory such as *superdense coding* and *teleportation*.

- **superdense coding**
  superdense coding is the protocol where two classical bits of information are encoded into two entangled qubits: it is possible to prove that to send the whole information is necessary to send only one of the two qubits.
  A two qubit state is defined in an Hilbert space $\mathcal{H} \simeq \mathcal{H}_A \otimes \mathcal{H}_B \simeq \mathbb{C}^4$, which has complex dimension $dim(\mathcal{H}) = 4$, so there is a 4-states basis spanning the entire Hilbert space. We define the Bell's basis and we associate it to a string of 2 bits:

$$\begin{aligned} 00: \quad & |\beta_{00}\rangle \equiv \frac{|0_A 0_B\rangle + |1_A 1_B\rangle}{\sqrt{2}} \\ 11: \quad & |\beta_{11}\rangle \equiv \sigma_z^A |\beta_{00}\rangle = \frac{|0_A 0_B\rangle - |1_A 1_B\rangle}{\sqrt{2}} \\ 01: \quad & |\beta_{01}\rangle \equiv \sigma_x^A |\beta_{00}\rangle = \frac{|1_A 0_B\rangle + |0_A 1_B\rangle}{\sqrt{2}} \\ 10: \quad & |\beta_{10}\rangle \equiv \sigma_x^A \sigma_z^A |\beta_{00}\rangle = \frac{|1_A 0_B\rangle - |0_A 1_B\rangle}{\sqrt{2}} \end{aligned} \tag{1.21}$$

  Consider the scenario where the initial state is $|\beta_{00}\rangle$ and two observers, A and B, attempt to communicate; A sends a qubit to B: if A wants to send message 00 B simply measures the state without further operations, if A wants to send 11 A apply a σ_z operation to $|\beta_{00}\rangle$ in order to bring it to $|\beta_{11}\rangle$, then B measure his qubit; the same procedure for the remaining strings 10 and 01. It is possible to see that after B measures his single qubit he with certainty reconstruct the two bits message sent by A.

- **teleportation**
  This protocol demonstrates that is possible to transfer a generic single-qubit state with the help of two entangles qubits. Assume a generic single qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and an entangled state $|\beta_{00}\rangle = \frac{|0_A 0_B\rangle + |1_A 1_B\rangle}{\sqrt{2}}$. So the total Hilbert space is $\mathcal{H}_{tot} \simeq \mathcal{H} \otimes \mathcal{H}_{\beta 00} \simeq \mathbb{C}^8$ with the total state equal to:

$$\begin{aligned} & |\Psi\rangle = |\psi\rangle \otimes \frac{|0_A 0_B\rangle + |1_A 1_B\rangle}{\sqrt{2}} = \big(\alpha|0\rangle + \beta|1\rangle\big) \otimes \frac{|0_A 0_B\rangle + |1_A 1_B\rangle}{\sqrt{2}} \xrightarrow{CNOT(\psi,A)} \\ & \alpha|0\rangle \otimes \frac{|0_A 0_B\rangle + |1_A 1_B\rangle}{\sqrt{2}} + \beta|1\rangle \otimes \frac{|1_A 0_B\rangle + |0_A 1_B\rangle}{\sqrt{2}} \xrightarrow{H(\psi)} \\ & \xrightarrow{H(\psi)} \frac{1}{2}(\alpha|0\rangle + \beta|1\rangle)|00\rangle + \frac{1}{2}(\alpha|1\rangle + \beta|0\rangle)|01\rangle + \frac{1}{2}(\alpha|0\rangle - \beta|1\rangle)|10\rangle + \frac{1}{2}(\alpha|1\rangle - \beta|0\rangle)|11\rangle \end{aligned} \tag{1.22}$$

  So after these operations A measures his qubits and depending on the measure results B perform specific operations:

– if A measures $|00\rangle$ B recovers the original qubit without further operations;

– if A measures $|01\rangle$ B applies σ_x to his qubit;

– if A measures $|10\rangle$ B applies a σ_z;

– if A measures $|11\rangle$ B applies $\sigma_x \sigma_z$ operation;

Then after this protocol the state initial state $|\psi\rangle$ has moved to state B's qubit state $|\phi_B\rangle$ without sending of information nor cloning of states.

1.2 Quantum Computing

Definition 6. *Quantum computing is the paradigm that exploits quantum mechanical properties of matter in order to do calculations.*

We recall from section 1.1.1 that given a quantum system with Hamiltonian $\hat{H}$, the state $|\psi\rangle$ that describes such a system evolves as:

$$|\psi(t)\rangle = e^{\frac{-i}{\hbar}\hat{H}t}|\psi(0)\rangle \tag{1.23}$$

Since $\hat{H}$ is by definition hermitian the evolution is *unitary*; a consequence of unitary evolution in quantum mechanics is the conservation of probability current and hence of probability. We recall that an operator U is unitary iff $UU^{\dagger} = UU^{-1} = \mathbb{1}$, from which we find that $U^{\dagger} = U^{-1}$.

In quantum computing hence performing calculations means apply specific unitary operations $\hat{\hat{U}} = \prod_i^n \hat{U}_i$ (or *gates*) to the n qubits system, which is equivalent to let evolve the system according to a specific Hamiltonian $\hat{\hat{H}}$ which is the generator of such a string of unitary gates, and finally perform some measurement M[5] over the final total state; such gates could be either acting on single or multiple qubits, either fixed (i.e. as Pauli matrices) or parametrized (i.e. qubits rotations). Formally we can define a function $f(x)$ as:

$$f(x) = M\left(\hat{\hat{U}}(x)|\psi(0)\rangle\right) = M\left(e^{-\frac{i}{\hbar}\hat{\hat{H}}(x)t}|\psi(0)\rangle\right) \tag{1.24}$$

A set of finite coherent unitary operations that terminates with an ensemble of measurements can be considered a quantum algorithm. An efficient way to visualize the sequence of the transformations consists in drawing *quantum circuit diagrams*, whose principal elements are registers (horizontal lines) that follows the state of each qubit, and quantum gates (blocks), placed on the qubit lines in sequential order. It is possible to draw operations in (1.22) in the following quantum circuit:

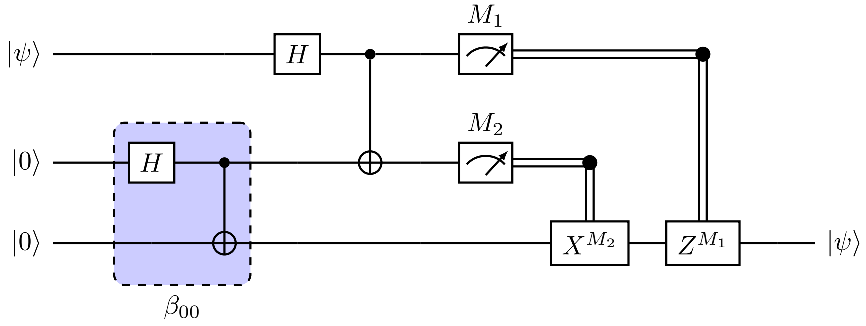


Figure 1.2: scheme of the quantum circuit used for teleportation protocol.

[5] here the only constraint is that the observed operator must be hermitian.

It is possible to give the definition of *logic gate* [7]:

Definition 7. *Given a set of k input qubits(bits) and l output qubits(bits), a logic gate is a function $f : \{0;1\}^k \longrightarrow \{0;1\}^l$.*

It is possible to demonstrate that classical computing is *universal*, so it is possible to define a finite set of *fixed elementary* logical gates from which one can build *any* possible function f; for example OR, AND, NOT, $CROSSOVER$, $FANOUT$ is a universal set of gates for classical computing. In particular the first three gates are one or two bits gates and can be represented as truth tables:

$$NOT = \begin{array}{c|c} 0 & 1 \\ 1 & 0 \end{array} \qquad OR = \begin{array}{cc|c} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{array} \qquad AND = \begin{array}{cc|c} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{array} \tag{1.25}$$

From (1.25) is clear that in general classical operation are $non-unitary$, hence classical computing is *irreversible*: in fact for OR and AND gates are not one-to-one functions, so it is no possible to retrieve the exact initial state after applying those gates.
However any classical circuit can always be replaced by an equivalent circuit containing only reversible elements, by making use of a classical reversible gate known as the Toffoli gate, which has three input bits and three output bits; it represent this kind of operation: $(a, b, c) \longrightarrow (a, b, c \oplus ab)$ and it is easy to show that it is reversible (in fact acting twice means computing $(a, b, c \oplus ab \oplus ab) = (a, b, c)$).

On the other hand, due to unitary evolution (1.23) of a quantum system, quantum computing is by definition *reversible*, hence for a generic $n-qubit$ gate U (which must be unitary) can always be defined its hermitian $U^\dagger = U^{-1}$, so that it is possible to bring back the system to its initial state:

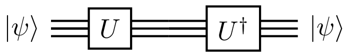


Figure 1.3: scheme of reversibility of quantum computation.

Since the generic group of unitary operator $U \in U(2)^{\otimes n}$ is generated by hermitian operators (1.7), we can define the most generic one-qubit gate:

$$\begin{aligned} &A = \alpha_0 \mathbb{1} + \vec{\alpha} \cdot \vec{\sigma} \\ &U = e^{-iA} = e^{-\frac{i\lambda}{2}(\vec{n}\cdot\vec{\sigma})} = \mathbb{1} cos\left(\frac{\lambda}{2}\right) - isin\left(\frac{\lambda}{2}\right)(\vec{n}\cdot\vec{\sigma}) \equiv R_{\vec{n}}(\lambda) \end{aligned} \tag{1.26}$$

that correspond to a generic rotation of angle λ along direction $\vec{n}$; eq.(1.26) can be rewritten by decomposing $R_{\vec{n}}$ as rotation of Euler's angles along x and z axis plus a global phase:

$$U \equiv R_{\vec{n}}(\lambda) = e^{i\alpha} R_z(\beta) R_x(\gamma) R_z(\delta) = e^{i\alpha} \begin{pmatrix} e^{-\frac{i\beta}{2}} & 0 \\ 0 & e^{\frac{i\beta}{2}} \end{pmatrix} \begin{pmatrix} cos\left(\frac{\gamma}{2}\right) & -sin\left(\frac{\gamma}{2}\right) \\ sin\left(\frac{\gamma}{2}\right) & cos\left(\frac{\gamma}{2}\right) \end{pmatrix} \begin{pmatrix} e^{-i\frac{\delta}{2}} & 0 \\ 0 & e^{i\frac{\delta}{2}} \end{pmatrix} \tag{1.27}$$

which means that a generic gate U can be defined by using four real parameters.

In principle simulating classical circuits with quantum ones could be a problem because of the non-reversibility of classical computing, but a quantum $Toffoli$ gate can be implemented, then is possible to simulate even non-unitary classical circuits; in fact it can be demonstrated that even quantum computing is *universal*, hence can be defined a set of gates that can approximate to arbitrary accuracy any unitary operation; such a universal set is: $\{H, CNOT, S, T\}$, which are represented as matrices:

$$H \equiv \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \qquad S \equiv \begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix} \qquad T \equiv \begin{bmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{bmatrix} \qquad CNOT \equiv \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix} \tag{1.28}$$

We previously said that any quantum gate can be approximated *to arbitrary accuracy* using Hadamard, S, CNOT , and T gates: this is an exact correspondence only in the limit of infinite string of universal quantum gates.
Given a generic n-qubit gate U it is possible to prove that it can *exactly* be expressed as a product of 2-qubits gates; this is due to algebraic properties of quantum gates, however this step is not so efficient because it requires a number $\mathcal{O}(2^{2n})$ of 2-qubits unitary operations. Moreover it can also be shown that any 2-qubits quantum gate $\tilde{U}$ can be decomposed *exactly* in terms of $CNOT$ and single qubit gates, which decomposition requires a number $\mathcal{O}(n^2)$ single qubit and $CNOT$ gates. In total such decomposition of a generic n-qubit gate U into single qubit and $CNOT$ gates requires $\mathcal{O}(n^2 2^{2n})$ and is an exact result.

Obviously, a discrete set of gates can't be used to implement an arbitrary unitary operation as in (1.26) exactly, since the set of unitary operations is continuous. Rather, it turns out that a discrete set can be used to *approximate* any unitary operation. To explain the concept of *gate approximation*, suppose U and V are two unitary operators that act on the same single qubit Hilbert space $\mathcal{H}$, where U is the target unitary operator that we wish to implement, and V is the unitary operator that is actually implemented in practice. We define the error of V with respect to U as:

$$E(U, V) = \max_{|\psi\rangle} ||(U - V)|\psi\rangle|| \tag{1.29}$$

where the maximum is over all normalized quantum states $|\psi\rangle$ in $\mathcal{H}$. This definition of approximation error represents how similar is the statistic of any measurement performed on the state $V|\psi\rangle$ to the one expected for $U|\psi\rangle$, with $|\psi\rangle$ is a generic initial state[6].
Then, given a set of universal gate, a single qubit gate can be approximated with error ϵ by a string of such universal gate, with the length N of the string given by the $Solvay - Kitaev$ theorem [7], which states that $N = O(log^c(\frac{1}{\epsilon}))$[7], with $c \sim 2$ constant.

Proving quantum computing is (up to an error ϵ) universal is a crucial part of this work because it guarantees us that, without further investigations, our computational problem is well defined in quantum computing and can be effectively be coded in a quantum algorithm.

Quantum computing is a very wide research field because it is thought that for a lot of tasks quantum computing may achieve *quantum advantage* over classical systems, i.e. proving that either quantum processors or ad-hoc quantum algorithms over-perform any classical ones for specific tasks, by setting new benchmarks or achieving speed-ups.
On the other hand the term *quantum supremacy* indicates that a quantum computer is able to solve a problem that classical hardware could never solve given a realistic amount of time or resources.

This thesis is about QML, which is the application of quantum computing to machine learning, hence before introducing and discussing about QML formalism it is useful to show in sec.1.3

[6] to check the proof please go to (A.1)

[7] We remark that, because the behaviour of N is logarithmic, it is always $finite$.

the basic concept of its classical counterpart; indeed we have a quite profound knowledge about classical deep learning, whereas many topics in QML still lack of explanation, so talking about classical deep learning will give us the opportunity to introduce techniques (such as the concept of graph transformers) and problems that are classically well known but are still open questions in the quantum theory.

1.3 Classical Graph Deep Learning

In this section we will introduce the basic knowledge of deep learning in a classical fashion, in particular related to graph learning, according to the kernel of this work. In fact, even though in this thesis we did not make use of classical machine learning the starting point of this work was the article [8], which was the first one to extend the usage of graph neural network for computing matrix elements $|M|^2$ for a given Feynman diagram. Moveover, discussing about classical deep learning will help us to introduce in a more exhaustive manner some crucial problems that come up both in ML and QML especially when dealing with graph-structured datas.

1.3.1 Graph Theory

Before starting to talk about graph learning it is necessary to give some fundamental definition about graphs; from [9, 10] a graph can be defined in the following way:

Definition 8. *A graph $G = (V, E)$ is an object consisting of two sets $V = V(G)$ and $E = E(G)$ called respectively the vertex set and edge set. V is a finite non empty set of elements (called vertices), whether E may be empty, but otherwise its elements (called edges) are non-ordered two-element subsets of V.*

Given two vertices $u, v \in V$, if their non-ordered connection $\{u, v\} \in E$, we say that u, v are adjacent vertices in G, whether two edges $e, g \in E$ incident to the same vertex $v \in V$ are called adjacent.

We say that $G' = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subset V$ and $E' \subset E$.

Two graphs $G = (V, E)$ and $H = (W, F)$ are said to be isomorphic if there is a permutation p of V such that u, v is in the set of graph edges $E(G)$ iff $p(u), p(v)$ is in the set of graph edges $F(H)$.

Two graphs G and H are isomorphic if one may be obtained from the other by renaming the vertices.

A vertex $v \in V$ with $deg_G(v) = 0$ is called isolated.

Given a graph G, the degree $deg_G(v)$ of a vertex $v \in G$ is equal to the number of vertices incident with v.

We highlight here that the definition of isomorphism between two graphs is strictly related to the concept of *topologically independent* Feynman diagrams in QFT perturbation theory and, since graphically speaking a Feynman diagram is basically a *labeled* graph, it become natural to work on graph learning. A more exhaustive presentation of *Quantum Field Theory* will be discussed in chapter 2.

Definition 9. *A graph $G = (V, E)$ is called directed if the edge set E is a set of ordered pairs of vertices called directed edges.*

In our use-case we will deal with directed graphs to give the idea of time flow along the edge orientation and to be in agreement with QFT considerations, whether in [8] the authors used undirected ones.

Definition 10. *A graph $G = (V, E, L)$ is called a labeled graph, where L is an assigned string of generic datas.*
A vertex labelling is a function of V to a set of labels (that could be different data types); a graph with such a function is called a vertex-labeled graph.
Likewise, an edge labelling is a function of E to a set of labels (of different data types). In this case, the graph is called an edge-labeled graph.

Definition 11. *a graph $G = (V, E, L)$ is called heterogeneous if the elements $v \in V$ or $e \in E$ have different features. If a graph is not heterogeneous is called homogeneous.*

In graph learning one of the biggest problems consists in encoding efficiently all the features in the dataset, so depending on it we will have to work on a specific kind of graphs.

1.3.2 Graph Learning

It is well recognized that graph data is often sophisticated and thus is challenging to deal with. To process graph data effectively, the first critical challenge is to find effective graph data representation, that is, how to represent graphs concisely so that advanced analytic tasks, such as pattern discovery, analysis, and prediction, can be conducted efficiently in both time and space.

In fact most of the machine learning methods can not be applied to graph data represented in the traditional way; those methods indeed usually assume that data samples can be represented by independent vectors in a vector space, while the components fo graph data are dependant to each other. Although it is possible to represent a node by its corresponding row vector in the adjacency matrix it is not an efficient representation because of its very high dimensionality so computationally expensive [11]. To tackle these challenges, substantial effort has been committed to develop novel graph representation learning [12, 13], i.e., learning the dense and continuous low-dimensional vector representations for nodes and labels, so that the noise or redundant information can be reduced and the intrinsic structure information can be preserved [14]. In order to make the representation space well supporting graph analysis tasks, there are two goals for graph representation learning. First, the original graph can be reconstructed from the learned representation space. Second, the learned representation space can effectively support graph inference, such as predicting unseen links, identifying important nodes, and inferring node labels.

Most graph neural network methods are based on or generalized from deep learning techniques (i.e. convolutional layers for image processing), because they can encode and represent efficiently graph data into vectors. Thus, it is convenient so simply define a proper representation of graph that can be used for downstream tasks[8] without an explicit embedding process of the raw data. There are three main representations for graphs:

- Adjacency Matrix: is a $|V| \times |V|$ matrix representation of graph $G = (V, E)$, where $|V|$ is the order od the vertex set. Element (i, j) in this matrix is 1 if and only if the couple $(v_i, v_j) \in E$.Thus, an adjacency matrix takes up $\mathcal{O}(|V|^2)$ storage. The adjacency matrix for the undirected graph is always symmetric, whether the one of directed graph is not.
- Adjacency List: In the adjacency list representation, a graph is represented as an array of linked list. The index of the array represents a vertex and each element in its linked list represents the vertices that form an edge with the vertex.
- Edge list: list of all the edges in a graph. Each edge is represented by a pair of vertices. The edge list can also include the weight or cost of each edge.

[8] i.e. node classification, link prediction or regression

Obviously the choice of the graph representation depends on the algorithm to be used.

This works well in very simple cases such as unlabeled graph datas, while in other cases the graph structure is much more complex that is necessary to preprocess all pieces of information (such as node or edges labels, weights etc.) to obtain a reliable embedding of the data on a vector to work with.

Given a generic labeled graph $G = (V, E, L)$, the essential idea of graph neural networks is to iteratively update the node representations by combining the representations of their neighbors and their own representations; by varying the way these nodes' representations are combined we are defining different kind of GNN layers, and this is because it is easy to see that each of these ways to process information exploits specific properties (such as invariance under a specific transformation) of the dataset and will then have different performance.

Generally the framework for any kind of graph neural network is composed by two different steps: *aggregation* and *combination* of the input data.

- *AGGREGATE*, which tries to aggregate the information from the neighbors of each node.
- *COMBINE*, which tries to update the node representations by combining the aggregated information from neighbors with the current node representations.

Mathematically it means that a single GNN layer performs the following transformation' steps:

$$\begin{aligned} a_v &= AGGREGATE\{H_u^{in} : u \in N(v)\} \\ H_v^{out} &= COMBINE\{H^{in} a_v\} \end{aligned} \tag{1.30}$$

Where $H^{in} \in \mathbb{R}^{|V| \times d}$ is the input node representation and $N(v)$ is the set of the neighbors of the node v; finally the output H^{out} can be used for downstream tasks in standard neural networks and trained.

Vanilla Graph Layer

The simplest graph layer that could be defined is the so called *Vanilla Graph Layer*, which simply a graph linear layer, where the information about the structure of the graph is given by its adjacency matrix. Given a node-labeled graph $G = (V, E, L)$ with adjacency matrix A_{ij} we can define the Vanilla Graph Layer as:

$$\begin{aligned} H &= \tilde{A}^T L W^T \\ h_i &= \sum_{j \in N(i)} l_j W^T \end{aligned} \tag{1.31}$$

which is a matrix multiplication between the trainable weights matrix W, the node features matrix L and $\tilde{A} = A + \mathbb{1}$; this is equivalent to sum over the neighbors nodes Ni of node i the product between the $j - th$ node features vector l_j and the weight matrix W.

for a network of K Vanilla Graph Layers the message is propagating as:

$$\begin{aligned} H^{(k+1)} &= \tilde{A}^T H^{(k)} (W^{(k)})^T \\ h_i^{(k+1)} &= \sum_{j \in N(i)} h_j^{(k)} (W^{(k)})^T \end{aligned} \tag{1.32}$$

with $k = 0, ..., K-1$ and $H^{(0)} \equiv L$, or $h_i^{(0)} \equiv l_i$. In this example, from eq. (1.31),(1.32), we clearly see that the node message propagates only in the neighbors nodes, and this is due to the fact that we are multiplying the linear transformation with $\tilde{A}$, which contains information about the graph topology.

Graph Convolutional Layer

In a graph convolutional layer the node representations in each layer l is updated according to the following propagation rule:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{1.33}$$

where $\tilde{A} = A + \mathbb{1}, \quad \mathbb{1} \in \mathbb{R}^{|V|\times|V|}$ is the adjacency matrix of the given undirected graph $G = (V, E, L)$ with selfconnections and $H^{(0)} = L$ list of node features. $\tilde{D}$ is the diagonal degree matrix, which collects the degree for each vertex v, with $\tilde{D}_{ij} \equiv \sum_j \tilde{A}_{ij}$, $\sigma(\cdot)$ is a proper activation function and $W_{ij} \in \mathbb{R}^{d\times d'}$ is the matrix of trainable parameters that takes an object with dimension d and brings it to the output dimension d'.

We can rewrite eq.(1.33) as:

$$H_i^{(l+1)} = \sigma\left(\sum_{j\in N(i)} \frac{A_{ij}}{\sqrt{\tilde{D}_{ii}\tilde{D}_{jj}}}H_j^{(l)}W^{(l)} + \frac{1}{\tilde{D}_i}H_i^{(l)}W^{(l)}\right) \tag{1.34}$$

and interpret what the layer is doing: the first term represents the *AGGREGATION* step, which consist on a weighted average of all neighbor representations $H_j^{(l)}$ times the adjacency matrix A_{ij}; the weights are $\sqrt{\tilde{D}_{ii}\tilde{D}_{jj}}$, which are the degrees of vertices i and j. On the other hand the *COMBINE* step is simply the summation of the aggregrated messages plus the term $H_i^{(l)}W^{(l)}/\tilde{D}_i$, which is the weighted representation of node i.

It is important to notice that in formulas (1.33),(1.34) the information about node features is aggregated with respect to the adjacency matrix A_{ij}, which is a good method to give explicitly to the network information about graph topology, but on the other hand could be pretty limiting when dealing with heterogeneous graphs, which need more complicated aggregation functions that aggregate two unlinked vertices for a more complete information flow in the entire graph.

Graph Transformer Layer

As we previously said tipically GNNs works well for fixed and homogeneous graphs, but they often get worse with noisy/dynamical and heterogeneous graphs, and this is due to the fact that convolutional layers are determined by the fixed graph structure.

In deep learning transformers layers are currently the best performing neural network architectures for handling long-term sequential datasets such as sentences in Natural Language Processing, and their framework can be generalized even for graph-structured datas. Different kind of graph transformers layers can be defined, from simple attention layer mechanism to more complex meta-path generating algorithms[15]. In this section we will review only the first one, the graph attention layer (or *GAT*), which is used to learn graph representation. The reason why GTs improve upon GNNs (for some analytic tasks) is that, unlikely for example GCNs, they assume an all-to-all graph connectivity and can learn hidden links which are not present in the original graph structure but are essential to better learn the task.

Graph attention layer: There are some cases where GCN layers doesn't work well, for example in an edge graph may occur that the edge weights doesn't reflect the true correlation between two nodes, or maybe two unlinked nodes are correlated; as we previously said GCNs can not handle this problems, so can be useful to define a layer which automatically learns the importance of each neighbor. Graph Attention Networks, introduced by [16] is built on this idea and try to learn the importance of each neighbor based on the Attention mechanism. For graph neural networks an attention layer defines how to transfer the hidden node representations $H^{(l)} \in \mathbb{R}^{|V| \times d}$ in layer l to layer $l+1$ and its corresponding node representations $H^{(l+1)} \in \mathbb{R}^{|V| \times d'}$. The input of the layer l is a set of node features, $H^{(l)} = \{H_1^{(l)}, H_2^{(l)}, ..., H_{|V|}^{(l)}\}$, where $H_i^{(l)} \in \mathbb{R}^d$, where d is the dimension of the feature space in each node The layer produces a new set of node. Analogously this layer will return as out a brand new representation of the node features $H^{(l+1)} = \{H_1^{(l+1)}, H_2^{(l+1)}, ..., H_{|V|}^{(l+1)}\}, \quad H_i^{(l+1)} \in \mathbb{R}^{d'} \quad \forall i = 0, ..., |V|$, that in general lives in a different feature space ($d \neq d'$).

As an initial step for an attention layer, a shared linear transformation, parameterized by a weight matrix, $W^{(l)} \in \mathbb{R}^{d' \times d}$, is applied to every node; we then perform self-attention, a function $a : \mathbb{R}^{d'} \times \mathbb{R}^{d'} \longrightarrow \mathbb{R}$ on the nodes, in order to compute the attention coefficients:

$$e_{ij}^{(l)} = a\big(W^{(l)} H_i^{(l)}, W^{(l)} H_j^{(l)}\big) \tag{1.35}$$

$e_{ij}^{(l)}$ indicates the importance of node j's features to node i at layer l.
As it is possible to see there is no dependence on the structural information of the graph, since (1.35) allows every node to attend on every other node (even though they're unlinked). However it is always possible to put some constraints to the system and allow only to compute $e_{ij}^{(l)}$ for nodes $j \in N(i)$, where $N(i)$ is the set of neighbors of node i (masked attention mechanism). finally to make the coeffcients comparable across different nodes, the attention coeffcients $e_{ij}^{(l)}$ are usually normalized with the SoftMax function:

$$\alpha_{ij}^{(l)} = SoftMax(\{e_{ij}^{(l)}\}) = \frac{exp(e_{ij}^{(l)})}{\sum_{k \in V} exp(e_{ik}^{(l)})} \tag{1.36}$$

which correspond to the probability of the node i to jump to the node j.
Finally the $i-th$ output node representation will be given by:

$$H_i^{(l+1)} = \sigma\bigg(\sum_{j \in V} \alpha_{ij}^{(l)} W^{(l)} H_j^{(l)}\bigg) \tag{1.37}$$

where σ is an activation function.
Usually, instead of only using one single attention mechanism, multi-head attention can be used, each of which determines a different *similarity function* over the nodes. For each attention head, we can independently obtain a new node representation, so that The final node representation $H^{(l+1)}$ will be for example either a concatenation of the node representations or the mean value of the T learned attention heads:

$$\begin{aligned} H_i^{(l+1)} &= \Big|\Big|_{t=1}^{T} \sigma\bigg(\sum_{j \in V} \alpha_{ij}^{(l)(t)} W^{(l)(t)} H_j^{(l)}\bigg) \\ H_i^{(l+1)} &= \sigma\bigg(\frac{1}{T}\sum_{t=1}^{T}\sum_{j \in V} \alpha_{ij}^{(l)(t)} W^{(l)(t)} H_j^{(l)}\bigg) \end{aligned} \tag{1.38}$$

where l runs over the number of layers of the network and $t = 0, ..., T$ is the number of attention heads we've used.

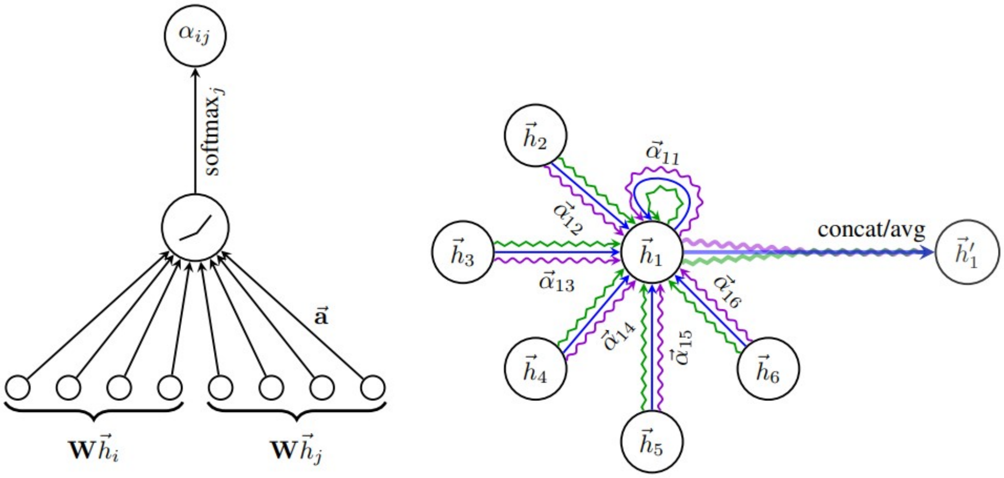


Figure 1.4: Scheme of the GAT layer mechanism; on the left is shown the process by which the layer computes the attention score α_{ij}; on the right it is represented the multi-head case: each attention score $\vec{\alpha_1}_i, \quad \forall i = 1, ..., 6$ is computed for each $\vec{h}_i$ three times; each time the features gets aggregated and finally the three attention heads are grouped together by a concatenation or an average, in order to obtain output representation $\vec{h}'_1$

1.4 Quantum Machine Learning

Quantum Machine Learning (QML) is a field of research that combines quantum computing and machine learning techniques with the hope of achieving improvements in both fields. QML is a wide and diversified topic, and it is due to the fact that still doesn't exist a consolidated approach as in classical ML, but there are plenty of workframes that can be defined.

First QML can be split in four different categories, depending on how data is generated and processed:

- The data can be generated by a quantum (Q) or classical system (C).
- The processing device can be quantum (Q) or classical (C).

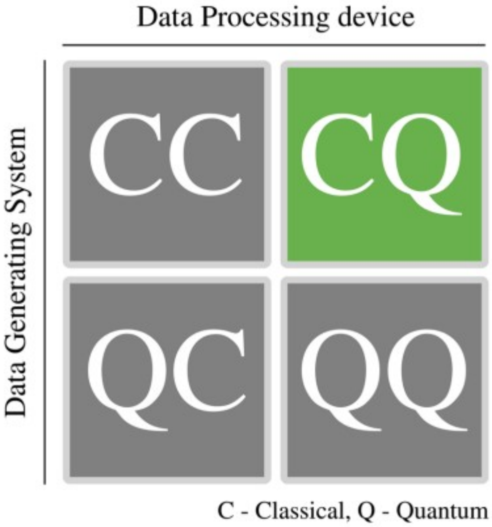


Figure 1.5: scheme representing the four categories of quantum machine learning, depending on the kind of processed data (either classical or quantum) and on the used processors (either classical or quantum too).

- **CC**: in this case classical data are processed classically, which is the conventional approach to machine learning. In the context of quantum machine learning, it refers to classical machine learning methods (for example tensor networks) which share some quantum-like properties in order to enhance the accuracy, the expressiveness and the computational cost of the network [17].
- **QC**: The case of quantum data processed by classical ML algorithm; this techniques study how deep a ML algorithm can apprehend about quantum systems.
- **CQ**: in this framework quantum computing is used to process classical datasets and constitutes the main topic of this thesis, so in the next sections we will focus only on this. The data can be constituted of any kind of observations from classical systems, and try to design a good quantum algorithm to look for any sort of *quantum advantage* (from efficiency and computational cost to accuracy or new benchmarks) with respect to its classical counterparts.
- **QQ**: in this approach quantum data are used, which are then processed by quantum computers. The data could either be derived from measuring a quantum system in a physical experiment and then passed to the quantum device, or could be derived from a quantum computer that simulate the dynamics of a quantum system. Nowadays this is the best performing approach, which shows better all the potentiality of QML in reaching quantum advantage. Obviously that is because we are studying quantum system with quantum devices, which is the most straightforward way [18, 19, 20, 21].

As in conventional machine learning the fundamental steps of quantum machine learning are *training* and *validation*, *testing* over different subsets of the dataset[9]. Generally quantum machine learning can be either *fully-quantum*, which means that all the computational operations are executed by quantum devices, or a *classical-quantum hybrid*, which means that the global

[9] in fact each separate process has to be agnostic to the other, which means that we can not evaluate the learning capability of a system with data used for training, because by doing this we are not evaluating its generalization and learning power of the task since we are studying already known information.

architecture of the system is composed by both classical and quantum layers and operations (especially classical layers are used for pre-processing, embedding and post-processing of data in order to have better performances). Properly speaking even fully quantum architectures are not composed exclusively by quantum operations since the optimization step of the learnable parameters in the training process, is computed with classical methods (such as gradient computing and gradient descend) because of a lack of efficent quantum optimization and gradient routines; so in their whole structure they are hybrid methods too, but such a distinguish is referred to the *forward* step of the computation, which is indeed fully quantum.
The two most used methods used in QML are *Quantum Kernels* and *Quantum Neural Network*.

1.4.1 Feature Map

In this project we will work on classical data processed by quantum devices, so we are going to to focus on CQ approach.
One critical aspect in this approach is the choice of encoding the information into quantum states. In fact, while quantum data are naturally encoded into quantum states, there exist a huge number of different ways to encode classic information, and the performance of the system is crucially determined by this choice. Another game-changing aspect is the right choice of the *ansatz* part of the circuit, where different pipelines can give completely different performances. A explanation of the building blocks of a quantum machine learning model is postponed in the following sections. As we previously said it is fundamental for a good performing system to define the best way to encode effectively classical pieces of information into quantum states, which otherwise would be incompatible objects. We can then introduce the concept of *quantum feature map.*

Definition 12. *Given a generic dataset $S = \{(\vec{x}_i, y_i)\}_{i=0}^{l}$ with $\vec{x}_i \in \mathcal{X}$ with m feature dimension and an n-qubit state $|\psi\rangle \in \mathcal{H} \equiv \mathbb{C}^{2^n}$ initialized as the all-zero state $|0\rangle^{\otimes n}$, we call* ***feature map*** *a map that takes an instance from the data space $\mathcal{X}$ to the feature space $\Omega \subset \mathcal{H}$:*

$$\begin{aligned} \Phi &: \mathcal{X} \longrightarrow \Omega \subset \mathcal{H} \\ \Phi &: \vec{x}_i \xrightarrow{\Phi} |\Phi(\vec{x}_i)\rangle \end{aligned} \tag{1.39}$$

In practice, data is encoded via a state preparation circuit $U(\vec{x}_i)$ that depends on the instance which acts on an initial state $|\psi\rangle$ (conventionally set to $|0\rangle^{\otimes n}$):

$$U(\vec{x}_i)|0\rangle^{\otimes n} = |\Phi(\vec{x}_i)\rangle \tag{1.40}$$

For $U(\vec{x}_i)$ to be useful for a data encoding, it should have several desirable properties. First it should depend non linearly on the instance $\vec{x}_i$, have a number of gates which is at most polynomial in the number of qubits as well as the size of the data set, dimension of the data to be encoded and other input parameters. Moreover, for machine-learning applications, we want the feature map to have enough free parameters such that there is a unique quantum state $\psi(\vec{x}_i)$ for each feature vector $\vec{x}_i$, so it has to be a bijective function [22]. Below some examples of the most used feature maps are shown:

- *Basis encoding*:
  Basis encoding is the most intuitive way to encode classical information into a quantum state. It encodes an n-bit binary string x to an n-qubit[10] quantum state in the computational basis, therefore, classical data has to be in the form of binary strings.
  Cosider a dataset $S = \{\vec{x}_i\}_{i=0}^{m}$, with $\vec{x}_i = (b_1, ..., b_n)$ with $b_j, \quad j = 0, ..., n$ a classical bit of information, it will be embedded in the following way:

[10] Hence, one bit of classical information is represented by one quantum subsystem.

$$|S\rangle = \frac{1}{\sqrt{M}} \sum_{i=0}^{m} |\vec{x}_i\rangle \tag{1.41}$$

- *Amplitude encoding*:
  Amplitude encoding encodes a vector $\vec{x}_i$ of length m into amplitudes of an n-qubit quantum state with $n = \lceil log_2(N) \rceil$:

  $$|\vec{x}_i\rangle = \sum_{j=0}^{m} x_i^j |j\rangle \tag{1.42}$$

  where $|j\rangle$ is the computational basis and the amplitudes are normalized: $|\vec{x}_i|^2 = 1 \quad \forall i$.

- *Angle encoding*:
  Angle encoding makes use of rotation gates to encode classical information $\vec{x}_i$. The classical information determines angles of single qubit rotation gates:

  $$|\vec{x}_i\rangle = \otimes_{j=0}^{n} R(x_i^j)|0\rangle^{\otimes n} \tag{1.43}$$

  where $|j\rangle$ is the computational basis again and operator R is a single qubit rotation gate choosen between rotations along x,y or z axis. Usually, the number of qubits used for encoding is equal to the dimension of the instance.

- *Product state encoding*:
  given an instance $\vec{x}_i$ this feature map encodes each feature $x_i^j, \quad j = 0, ..., m$ through a set of single-qubit gates $U_j(x_i^j)$; the state then becomes:

  $$\begin{aligned} U(\vec{x}_i) &\equiv \otimes_{i=1}^{m} U_j(x_i^j) \\ U(\vec{x}_i)|0\rangle &= \otimes_{i=1}^{m} U_j(x_i^j)|0\rangle = |\Phi(\vec{x}_i)\rangle \end{aligned} \tag{1.44}$$

- *Hamiltonian encoding*:
  Since feature map is expressed as a multi-qubit quantum gate that depends on single features, it can always be written as the exponential map of a specific Hamiltonian $\tilde{H}$ of the system; we can then encode information of a datapoint $\vec{x}_i$ by defining a suitable Hamiltonian that depends on $\vec{x}_i$ and let evolve the state according to it:

  $$\begin{aligned} U(\vec{x}_i) &\equiv e^{-i\tilde{H}(\vec{x}_i)} \\ U(\vec{x}_i)|0\rangle^{\otimes n} &= |\Phi(\vec{x}_i)\rangle \end{aligned} \tag{1.45}$$

  where the Hamiltonian in general can be defined as a combination of single-qubit gates and an entangling part of two-qubit gates[11]:

  $$\tilde{H}(\vec{x}_i) = \sum_{j,k \in D} J_{ij}(\vec{x}_i)\hat{Z}_j\hat{Z}_k(\vec{x}_i) + \sum_j Q_j(\vec{x}_i)\hat{X}_j \tag{1.46}$$

  In this case we've defined the Hamiltonian as the combination of $\hat{Z}_j\hat{Z}_k$ operation, which entangles the state, and a series of Pauli $\hat{X}$ gates, respectively weighted by a weight matrix J_{jk} and Q_j that depend both on the input data $\vec{x}_i$. This approach is a more complex way to encode information, but often is very useful to define a customized Hamiltonian that acts on the circuit.

[11] remind that any multi qubit gate can be exactly be expressed as combination of single and two qubit gates [7].

1.4.2 Quantum Neural Networks

Most of the time machine learning algorithms are quite complex and they require to perform a big number of computations, i.e. non-convex optimisation when training neural networks, that are computationally hard in general. This situation creates challenges for both fault-tolerant and NISQ (noisy intermediate scale quantum) devices approaches to machine learning. Even though near-term quantum computing is well suited for quantum machine learning algorithms, they suffer from several limitations; in fact to perform even toy-model benchmarks on simulators and early-stage quantum hardware, quantum machine learning algorithms underlie strict design limitations and, most importantly, nowadays, both in simulations and real quantum hardware, we have access only to a few qubits and relatively shallow circuits, since large routines are not simulable on classical hardware and quickly become drowned in noise on real quantum devices.

As we previously said, usually such problems are faced in two different manners, that are either use hybrid quantum-classical algorithms, which consider brief quantum computations as parts of more complex classical ones, or a fully quantum approach, in order to investigate "quantum models" as alternatives to classical models. In both approaches solid candidates to exploit for machine learning purposes are Parametrized Quantum Circuit (PQC) [23]. It is possible to define the concept of *quantum neural network* as follows:

Definition 13. *Given a data input domain $\mathcal{X}$ with $\vec{x}_i \in \mathcal{X}$ and an n-qubit circuit defined in Hilbert space $\mathcal{H} \sim \mathbb{C}^{2^n}$,**quantum neural network** is a circuit composed by:*

- *feature map $U(\vec{x}_i)$ that embeds the instance $\vec{x}_i$ in a quantum state;*
- *a PQC called **ansatz** $W(\vec{\theta})$ that depends on the vector of trainable parameters $\vec{\theta} \in \mathbb{R}^N$;*
- *an hermitian operator $\mathcal{M} \in \mathcal{H}$[12] that corresponds to the observable we measure at the end of the circuit to extract the output of the QNN.*

The output of the network is then computed by performing a measurement over $\mathcal{M}$:

$$f_{\vec{\theta}}(\vec{x}_i) = \langle\psi(\vec{x}_i, \vec{\theta})|\mathcal{M}|\psi(\vec{x}_i, \vec{\theta})\rangle = \langle 0|U^\dagger(\vec{x}_i)W^\dagger(\vec{\theta})\mathcal{M}W(\vec{\theta})U(\vec{x}_i)|0\rangle \equiv \langle\mathcal{M}\rangle_{\vec{x}_i\vec{\theta}} \tag{1.47}$$

The optimization of the vector parameter is then executed by a classical algorithm. Schematically:

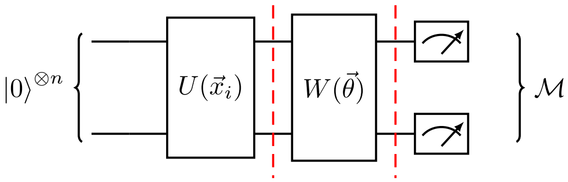


Figure 1.6: scheme of a generic quantum neural network.

where the ansatz gate $W(\theta)$ can be repeated L times, with L defined as the depth of the circuit. Obviously, since both feature map $U(\vec{x}_i)$ and ansatz $W(\vec{\theta})$ are unitary, they can be written in terms of their generators, which are hermitian operators and can be identified as the *feature Hamiltonian* $H_F(\vec{x}_i)$ and the *ansatz Hamiltonian* $H_A(\vec{\theta})$. It is noteworthy that the way that parameters and inputs enter the circuit is essentially the same.
This was the general pipeline, but can be defined much more architectures for a QNN, for example

[12] that can be seen as $2^n \times 2^n$ hermitian matrix

we can define a feature map $U(\vec{x}_i, \vec{\alpha})$ that mixes the embedding and parametrised block, so that the embedding itself trainable. or both feature map and ansatz can be repeated several times, a process called "data re-uploading" [24] Also, the input or output features could be the result of some classical pre-processing or post-processing of a set of features. Finally optimisation is done via a classical feedback loop that iteratively improves the parameters.

As we previously said a crucial aspect in quantum machine learning for a VQA to be successful is the choices of a good feature map and a good ansatz, because we want the model to have enough *expressibility.* For a QNN to be successful the solution (then the final gate configuration we want to achieve after training) needs to be contained within the ensemble of unitaries generated by the ansatz, formally:

Definition 14. *Given a QNN with feature map $U(\vec{x}_i)$ and ansatz $W(\theta)$ that generates the set of unitaries $\tilde{\mathbb{U}}$, and given $\mathbb{U}_S$ the set of solution unitaries, we say that the QNN is complete for the given problem iff:*

$$\mathbb{U}_S \cap \mathbb{U} \neq \emptyset \tag{1.48}$$

where both $\mathbb{U}, \mathbb{U}_S \in U(2^n)$, group of unitary transformation of dimension 2^n, with n number of qubits in the circuit.

Usually in quantum machine learning the solution set $\mathbb{U}_S$ is unknown, so in principle can not be built an ansatz which for sure would be complete, so in general the pipelines are to either use a problem-inspired ansatz, where properties and information about the problem are encoded in the ansatz such that I'm confident on the fact that $W(\vec{\theta})$ is complete, or build a problem-agnostic ansatz, which covers an big enough region of the unitary group $U(2^n)$ that it contains or overlap $\mathbb{U}_S$, such that $W'(\theta)$ is complete too. The latter ansatz $W'(\vec{\theta})$ is said to be *expressive*, where expressibility of an ansatz denotes how much the set $\mathbb{U}$ explores the unitary group $U(2^n)$ [25].

Definition 15. *Given the ansatz $W(\vec{\theta})$, let us consider the deviation of the state distribution generated by the circuit $|\psi(\vec{x}_i\vec{\theta}) \equiv W(\vec{\theta})U(\vec{x}_i)|0\rangle$ from the solution state generated by $\mathbb{U}_S$ $|\tilde{\psi}(\vec{x}_i)\rangle$; the expressibility of $W(\vec{\theta})$ is defined as [26]:*

$$\mathcal{A}^{(t)}(W) = \left|\left| \int_{U(2^n)} (|\tilde{\psi}\rangle\langle\tilde{\psi}|)^{\otimes t} d\psi - \int_{\Theta} (|\psi(\vec{x}_i, \vec{\theta})\rangle\langle\psi(\vec{x}_i, \theta)|)^{\otimes t} d\vec{\theta} \right|\right|_{HS} \tag{1.49}$$

where $\int_{U(2^n)}$ is the integration over the Haar measure of the solution state $\tilde{\psi}$, Θ is the set of all possible parameter vectors $\vec{\theta}$ and $||\cdot||_{HS}$ is the Hilbert-Schmidt distance. This can be easily computed as:

$$\mathcal{A}^{(t)}(W) = \mathcal{F}^{(t)}(W) - \mathcal{F}^{(t)}_{Haar} \equiv \int_{\Phi}\int_{\Theta} |\langle\psi(\vec{\theta})|\psi(\vec{\phi})\rangle|^{2t} d\vec{\theta} d\vec{\phi} - \int_{\mathbb{U}_S}\int_{\mathbb{U}_S} |\langle\tilde{\psi}|\tilde{\psi}'\rangle|^{2t} d\tilde{\psi} d\tilde{\psi}' \geq 0 \tag{1.50}$$

where Θ and Φ are the same set of parameters of the ansatz. So in general it is crucial to choose wisely the best framework between a more problem-agnostic or a problem-inspired ansatz in order to improve the performances of the network.

Quantum Graph Neural Network

Applications of quantum machine learning over graph-based tasks are thought to be probably ones of the most promising research fields where quantum computing may significantly outperform all the classical graph learning methods also in NISQ era.
As it was explained in the previous section, simple graph learning techniques, such as graph

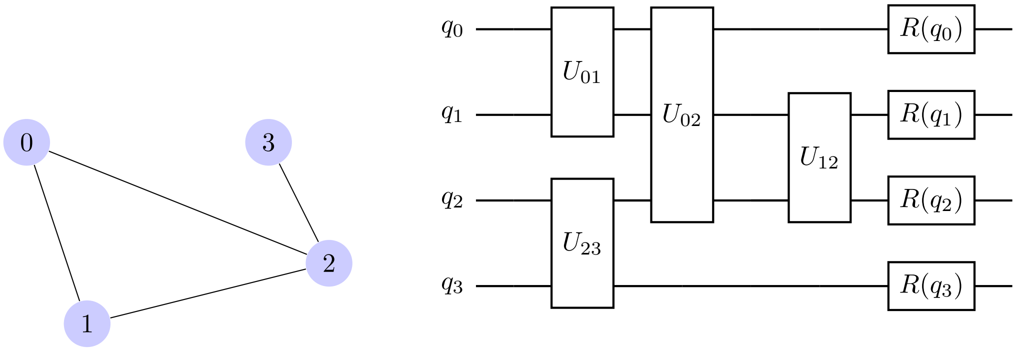
convolutional layers, cannot aggregate information over long paths, then when the downstream task relies on or hides remote information interactions, information from more distant nodes cannot be propagated, with consequently lack of performance; Though some more complex architectures such as graph transformers need to be introduced, that manage to integrate global information.

Here will be shown that graph learning fits naturally in a quantum computing regime because of specific properties of the quantum world; in fact we can exploit some of the quantum effects shown in 1.1.1, such as teleportation or entanglement. Thanks to these non-local effects of quantum computing global information can be transferred efficiently (short paths) in graph-structured data, enhancing in this sense conventional message passing methods.
Moreover the way graph data are encoded in the circuit is very intuitive, since it corresponds to evolve the initial state with respect to a specific Hamiltonian that links the qubits in the same way the nodes are linked in the graphs, creating this correspondence between qubits and nodes.

Given the labeled graph $G = (V, E, X)$, the most general ansatz for a quantum graph neural network is a parametrized circuit that can be written as the unitary evolution of one or more generic Hamiltonians [27]; such ansatz can also be repeated a number M of times, with M that denotes the depth of the circuit, so the number of layers of the network. Then we will have:

$$\begin{gathered} W_{\vec{\theta}} = \prod_{l=1}^{M} \left[\prod_{q=1}^{Q} e^{-i\hat{H}_q(\vec{\theta})} \right] \\ with \quad \hat{H}_q(\vec{\theta}) = \sum_{v \in V} A_{qv}(\vec{\theta}) \hat{P}_{qv} + \sum_{\{i,j\} \in E} B_{qij}(\vec{\theta}) \hat{O}_{qi} \otimes \hat{R}_{qj} \quad , \end{gathered} \tag{1.51}$$

where the operators $\hat{O}_{qi}, \hat{P}_{qv}$ and $\hat{R}_{qj}$ are hermitian single qubit operators; The crucial part in eq.(1.51) is that we are entangling the qubit with respect to the edges of G: in this way we are constraining the expressibility of the ansatz in order to have a problem-inspired ansatz with a good performance.
In some sense what we are actually doing is *encoding the topological information* of the graph G in the circuit thanks to the Hamiltonian in (1.51); that is why such a framework can also be useful to define a feature map for graph-structure data encoding. What we will do in our project is to define a feature map which encode the topology of the single data through entangling qubits with respect to E and then an ansatz gate which consider different graph topologies.



(a) (b)

Figure 1.7: Here it is shown how the quantum graph layer works: in fig.(a) is defined a simple generic graph $G = (V, E)$ with $V = \{0; 1; 2; 3\}$ and $E = \{(0, 1); (0, 2); (1, 2); (2, 3)\}$. In fig.(b) there is represented the quantum graph layer associated to this graph: it is possible to note that it is divided into two parts, the first *entangling part* composed by two-qubit operators which connectivity is E itself (so operation U_{ij} exists if and only if $(i, j) \in E$), and a single qubit row of operations. In this picture for simplicity we took the case where all the two-qubit gates commutes.

More complex architectures are being developed nowadays, like the quantum graph convolutional neural network, which is made of a succession of quantum convolutional and quantum pooling layers [28].
Just like the classical case, in order to define a quantum graph convolutional layer we should impose to the ansatz to be invariant under permutation of the nodes. By (1.51) it is clear that such a symmetry manifests itself as a constraint on the Hamiltonian, in which, for every layer of the circuit, the parameter vector $\vec{\theta}$ must have the same value for each node. The $\vec{\theta}$ parameters thus depends only on the number of layer and not on the qubits, so they become tied over indices of the graph: $A^{(l)_{qv}} \longrightarrow A_q^{(l)}$ and $B_{qij}^{(l)} \longrightarrow B_q^{(l)}$. After a quantum graph convolutional layer is usually used a quantum pooling layer, which reduces the feature dimension by introducing quantum measurements and achieves the same effect as the classical pooling layer. The model first measures a portion of qubits and then determines whether to implement unitary transformations on the neighbor qubits based on the measurement results.

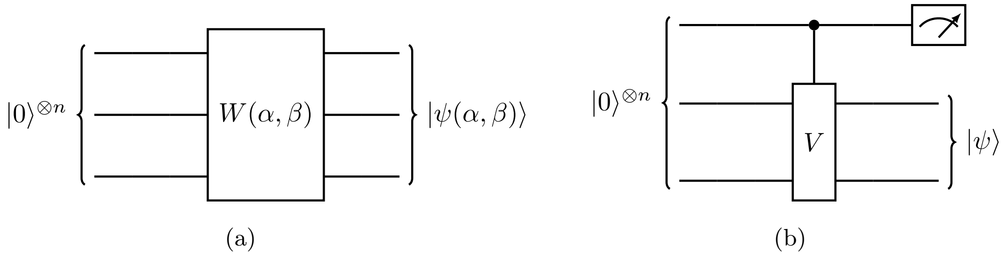


Figure 1.8: In fig.(a) it is shown the basic property of a quantum convolutional layer: as it is possible to see actually the ansatz in (1.51) depends only on two real values α and β, which are respectively singe-qubit parameter and two-qubit parameter, and that follows from the requirement that invariance under nodes permutation must be preserved. In fig.(b) on the other hand it is represented a simple pooling layer, which reduces the dimension of features thanks to a control-qubit operator $Control - V$; right after an intermediate measurement is performed on the control qubit, which then is discarded. Quantum pooling layer must be permutation invariant too (check equivariant QNN section for further details).

Like their classical counterparts, the previous two architectures explicitly encode the topology of the graph into the circuit and constrain the expressibility of the ansatz in order to build a cheaper and problem-inspired circuits. In general this framework can be very limiting because we are neglecting information flow and mixture between unconnected nodes, which in principle may be correlated. This could lead to the same problems we discussed in section 1.3.2. In order to overcome this limitation it is possible to introduce *Quantum Transformers*, which are nothing but defining a more expressive ansatz that includes more qubit connections.

A first simple example of quantum transformer is given by [29], where the authors encoded the graph topology as in (1.51) but then the trainable PQC[13] was defined as fully-connected ansatz; in this way the circuit allows information flow among all nodes, making it as amore expressive circuit.

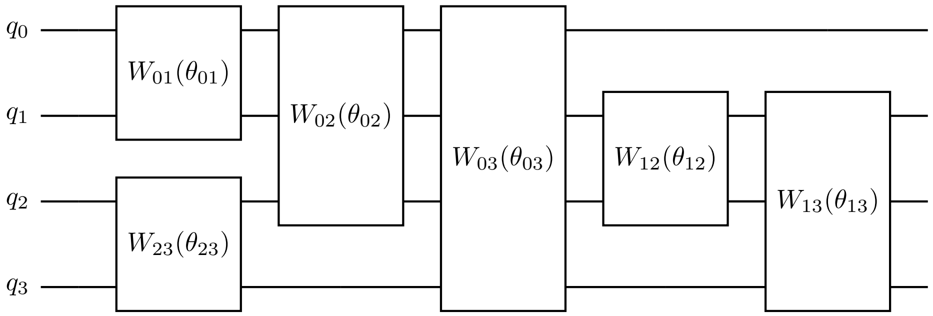


Figure 1.9: fully connected ansatz used in [29] for the graph in fig.1.7a; first, the graph is encoded by the feature map in fig.1.7b, then in the ansatz all-to-all connections between qubits (nodes) is allowed for a more expressibility of the circuit.

Another interesting work is the architecture presented in [30]; in this paper is not discussed a quantum circuit, but is presented a hybrid routine which computes a parameterized transition matrix or quantum attention matrix for a graph from the correlations of a quantum dynamic, whether the rest of the network is purely classical. Given an initial state $|\psi(0)\rangle$, we let it evolve:

[13] this stands for: *Parametrized Quantum Circuit*

$$|\psi(t)\rangle = \prod_{k=1}^{K}\left[e^{-i\hat{H}_M\theta_k}e^{-i\hat{H}_G t_k}\right]e^{-i\hat{H}_M\theta_0}|\psi(0)\rangle \quad , \tag{1.52}$$

where H_M is the ansatz Hamiltonian and H_G is the graph-encoding Hamiltonian, which can both be defined using (1.51) and repeated K-times, and with $\vec{\theta} = (\theta_0, t_1, \theta_1, ..., t_k, \theta_k)$ for $k = 0, ..., K$ are trainable parameters.
At the end of the circuit is then computed for each $(i, j) \in E$ the vector of two-bodies observables $C_{ij} = [\langle Z_iZ_j\rangle, \langle X_iX_j\rangle, \langle Y_iY_j\rangle, \langle Z_iX_j\rangle, \langle Z_iY_j\rangle, \langle X_iY_j\rangle, \langle Z_jX_i\rangle, \langle Z_jY_i\rangle, \langle X_jY_i\rangle]$. Finally the quantum attention matrix can be defined as:

$$\begin{aligned} A(\theta)_{ij} &= \gamma^T C_{ij} \\ or \quad A(\theta)_{ij} &= softmax(\gamma^T C_{ij}) \end{aligned} \tag{1.53}$$

where γ is a size-9 trainable parameter. Even though this ends the quantum part of the whole architecture, this paper can be seen as a further starting point to enrich the knowledge of quantum transformers for quantum graph lerning.

Equivariant Neural Networks

Quantum graph convolutional layers and their classical counterparts are constrained to be node-permutation invariant; this can lead us to introduce the concept of *Equivariant Neural Networks*, architectures that exploit a certain symmetry of the problem in order to optimize all the computation and decrease the complexity of the system. In fact is not so rare that the dataset reflects some symmetries, especially in physics, where the concept of symmetry plays a central role in any theory.

Definition 16. *Consider a dataset $S = (x_i, y_i)_{i=0}^N$ and a task where a prediction $y_i \in \mathcal{Y}$ should be associated with data points $x_i \in \mathcal{X}$ via an underlying function $f : \mathcal{X} \longrightarrow \mathcal{Y}$, with $f(x_i) = y_i$. We say that the dataset is invariant under a symmetry group $\mathcal{G}$ with representation $V_g : \mathcal{G} \longrightarrow Aut(\mathcal{X})$ if it holds:*

$$y_i = f(x_i) = f(V_g[x_i]) \quad \forall x_i \in \mathcal{X}, \quad \forall g \in \mathcal{G} \tag{1.54}$$

In quantum machine learning the output is extracted through the final state $|\psi\rangle$, so we can rewrite eq.(1.54) in quantum mechanical formalism [31, 32]:

$$y_i = f(\rho_i) = f(U_g\rho_iU_g^\dagger) \quad \forall \rho_i, \quad \forall g \in \mathcal{G} \tag{1.55}$$

The goal of QML models is to find an invariant hypothesis $h_{\vec{\theta}} : \mathcal{X} \longrightarrow \mathcal{Y}$ that best approximates the underlying function f in a way that we explicitly encode the system' symmetry in the building blocks of the circuit.
In order to define an equivariant quantum circuit all of its components need to be equivariant or invariant under such a symmetry group $\mathcal{G}$, from the initial state peparation to the data encoding, the PQC and the final measurement itself. Moreover, since we are dealing with quantum states and quantum operators, the representation U_g of the symmetry group $\mathcal{G}$ must be unitary.

- *initial state*
  When defining an equivariant circuit we have to prepare an initial state $|\psi(0)\rangle$ that is invariant under the symmetry group of the problem.

  Definition 17. *Given a generic state $|\psi\rangle$, it is called invariant under the symmetry group $\mathcal{G}$ if:*

  $$U_g|\psi\rangle = |\psi\rangle \quad \forall g \in \mathcal{G} \tag{1.56}$$

- *data encoding*

  Definition 18. *A quantum feature map $U(\vec{x}_i)$ is said to be equivariant with respect to the data symmetry V_g if:*
$$U(V_g[\vec{x}_i]) = U_g U(\vec{x}_i) U_g^\dagger \tag{1.57}$$
  $\forall \vec{x}_i \in \mathcal{X}$ and $\forall g \in \mathcal{G}$, where U_g is a unitary induced representation of the embedding $U(\vec{x}_i)$.

- *ansatz and observable* Recall that the ansatz gate is a collection by single or multi-qubit gates, typically parametrized, that can always be wrtitten as the exponential map of some generators $\Sigma = \{\sigma\}$:
$$R(\alpha) = e^{-i\sigma\alpha} \tag{1.58}$$

  Definition 19. *Hence, given a unitary representation U_g of the symmetry group $\mathcal{G}$, the quantum gate is equivariant with respect to $\mathcal{G}$ if:*
$$[R(\alpha), U_g] = 0 \quad \forall \alpha \in \mathbb{R}, \quad \forall g \in \mathcal{G} \tag{1.59}$$

  This statement is true if and only if the generator σ commutes with $U_g \quad \forall g \in \mathcal{G}$.

  Proof.
$$\begin{aligned} 0 &= [R(\alpha), U_g] = [e^{-i\alpha\sigma}, U_g] = [\mathbb{1} - i\alpha\sigma + \mathcal{O}(\alpha^2), U_g] \\ &= -i\alpha[\sigma, U_g] + \mathcal{O}(\alpha^2) \longleftrightarrow [\sigma, U_g] = 0 \quad \forall g \in \mathcal{G} \end{aligned} \tag{1.60}$$

  □

  Hence to build equivariant gates it is sufficient to define a generating set of commuting gates $\Gamma = \{\sigma \quad s.t. \quad [\sigma, U_g] = 0 \quad \forall g \in \mathcal{G}\}$. To define such a gateset it is possible to make use of the *twirling formula*

  Definition 20. *Let U_g be a unitary representation of $\mathcal{G}$, then [33, 34]:*
$$\mathcal{T}_U[O] = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} U_g O U_g^\dagger \tag{1.61}$$
  defines a projector onto the set of operators commuting with all the elements of the unitary representation, i.e. $[\mathcal{T}_U[O], U_g] = 0 \quad \forall O, \quad \forall g \in \mathcal{G}$.

  This definition takes into account only finite groups, so $|\mathcal{G}|$ is the order of the group; on the other hand for infinite groups such as Lie groups eq.(1.61) becomes an integral over the Haar measure of the group $\mathcal{G}$:
$$\mathcal{T}_U[O] \int d\mu(g) U_g O U_g^\dagger \tag{1.62}$$

  Hence by using eq.(1.61) on a generic gateset Σ it is possible to build its associated equivariant gateset $\Gamma \equiv \mathcal{T}_U[\Sigma] = \{\mathcal{T}_U[\sigma] \quad s.t. \quad \sigma \in \Sigma\}$. Also the final observable O must commute with any element of the unitary representation U_g, so the same arguments hold for it.
  A simple example on how to symmetrize the gates is shown in appendix A.2.

Following all the steps ensures an equivariant circuit, hence also the hypotesis function $h_{\vec{\theta}}$ of the circuit is invariant under symmetry group $\mathcal{G}$. Given the hypotesis $h_{\vec{\theta}}$ of the form:

$$\begin{aligned} h_{\vec{\theta}}(V_g[\vec{x}_i]) &\equiv \langle\psi|U^\dagger(V_g[\vec{x}_i])W^\dagger(\vec{\theta})\tilde{O}W(\vec{\theta})U(V_g[\vec{x}_i])|\psi\rangle \\ &= \langle\psi|U_g^\dagger U_g U^\dagger(\vec{x}_i)U_g^\dagger W^\dagger(\vec{\theta})\tilde{O}W(\vec{\theta})U_g U(\vec{x}_i)U_g^\dagger U_g|\psi\rangle \\ &= \langle\psi|U^\dagger(\vec{x}_i)W^\dagger(\vec{\theta})\tilde{O}W(\vec{\theta})U(\vec{x}_i)|\psi\rangle = h_{\vec{\theta}}(\vec{x}_i) \end{aligned} \tag{1.63}$$

so the hypotesis is proved to be invariant under $\mathcal{G}$.

An example of application of an equivariant QNN is the work described in [35], where the authors introduced an ansatz for learning tasks on weighted graphs that respects equivariance under node permutations. the authors evaluated the performance of this ansatz on a complex learning task, namely neural combinatorial optimization, where a machine learning model is used to learn a heuristic for a combinatorial optimization problem. It is also worthy to point out that both in [35, 33] managed to reduce the complexity of the system while reaching better performances than the ones found for non-equivariant architectures.

Chapter 2

Introduction to QFT

In this chapter we will discuss the basic concepts of Quantum Field Theories; this chapter does not aim to give a fulfilling overview of QFT, but it will briefly introduce only the topics that are strictly related to the goals of this thesis, which are Quantum Electrodynamics simple scattering processes.

Quantum Field Theory (QFT) is a theoretical framework of quantum physics that combines the principles of quantum mechanics with the ones of special relativity in order to describe the fundamental particles and their interactions.

In QFT, the fundamental particles are described as excitations of quantum fields with definite quantum numbers that permeate all space-time structure; moreover, some of those fields are associated to various fundamental forces, like the electromagnetic, weak nuclear, and strong nuclear forces, hence in QFT an interaction is interpreted as an exchange of particles. These interactions are described mathematically by Feynman diagrams, which provide a pictorial representation of particle interactions. QFT has been highly successful in explaining and predicting the behavior of fundamental particles and their interactions. Standard Model of particle physics is a widely accepted QFT that describes the electromagnetic, weak, and strong nuclear forces and the particles that mediate them. Nowadays it is consolidated that every quantum field theory is an *effective theory*, which means a low energy approximation of a deeper and more complete theory [36].

2.1 Scattering Theory

In Quantum Field Theory, observables are quantities that can be measured or calculated to describe the physical properties and behavior of the system. These observables play a fundamental role in understanding and predicting the outcomes of scattering interactions; in this sense the main observables are *cross sections σ, S-matrix elements $\langle f|\, S\,|i\rangle$ and probability amplitudes* $|\,\langle f|\,\mathcal{M}\,|i\rangle\,|^2$; as we will show, such observables are not independent from each other.
Moreover, since quantum field theories are relativistic theories, it follows that any observable must be Lorentz invariant.

The typical scattering processes investigated by colliders are 2-particles $\longrightarrow$ n-particles; without loss of generality, given an initial state $|i\rangle$ and a final one $|f\rangle$, then we can define the differential cross section for such a process:

$$d\sigma = \frac{1}{(2E_1)(2E_2)|\vec{v}_1 - \vec{v}_2|}|\,\langle f|\,\mathcal{M}\,|i\rangle\,|^2 \prod_{\mathit{final\ particles}\ j} \frac{1}{2E_{p_j}}\frac{d^3p_j}{(2\pi)^3}(2\pi)^4\delta^4\Big(\sum_{\mathit{ini}\ i} p_i - \sum_{\mathit{final}\ k} p_k\Big) \tag{2.1}$$

Now we analyze each term of (2.1):

- $\frac{1}{(2E_1)(2E_2)|\vec{v}_1-\vec{v}_2|}$ is the flux term, where $|\vec{v}_1 - \vec{v}_2|$ is the relative velocity between the two particles in the initial state, whereas E_1 and E_2 are their energy;
- $d\phi_n \equiv \frac{1}{2E_{p_j}}\frac{d^3p_j}{(2\pi)^3}(2\pi)^4\delta^4\Bigg(\sum_{ini\ i} p_i - \sum_{final\ j} p_j\Bigg)$ is the phase space of the process, which contains the conservation of the total four-momentum; it can be proven that this term is Lorentz-invariant
- $|\langle f|\,\mathcal{M}\,|i\rangle|^2$ is the scattering amplitude of the process, which corresponds to the probability of the scattering process to occur; this term encodes the dynamical description of the scattering. Scattering amplitudes are represented by Feynman diagrams and are Lorentz invariant too.

In QFT we are interested in the computation of the dynamical term $|\langle f|\,\mathcal{M}\,|i\rangle|^2$ of (2.1).

In the following section we will introduce the theoretical principles that describes the evolution of a quantum state with respect to an interacting Hamiltonian.

2.1.1 Interacting Fields

In field theories is common to use Lagrangian formalism, which define the fields as solutions of the e.o.m. (equations of motion) of specific Lagrangian densities. Given a Lagrangian density $\mathcal{L}$, we can find the e.o.m. from the variational principle of the fields action and compute the classical fields in term of normal modes of harmonic oscillators.
Non-interacting (free) field theory, where $\mathcal{L} = \mathcal{L}_0$, are exactly solvable, and it is then possible to canonically quantize the fields that are solution of the free-theory e.o.m.. In covariant quantization, fields themselves are defined as a linear combination of creation and annihilation operators $a^\dagger(\vec{k}), a(\vec{k})$ of normal modes of independent quantum harmonic oscillators with energy $\hbar\omega_{\vec{k}}$, which corresponds to the creation or destruction of a particle with four-momentum $P^\mu = \hbar k^\mu = \hbar(\omega_{\vec{k}}, \vec{k})$ and the same quantum numbers of the field itself; moreover, it is possible to define the occupation number $N(k) \equiv a^\dagger(k)a(k)$, which corresponds to the number of particles with momentum k; then it is possible to write a collection of n identical particles as the action of n creation operators $a^\dagger(k)$ for different momenta k on the vacuum state, namely:

$$\begin{aligned} |\psi\rangle &= \bigotimes_{\vec{k}} \Big|N(\vec{k})\Big\rangle \\ \text{where} \quad \Big|N(\vec{k})\Big\rangle &\equiv \frac{[a^\dagger(\vec{k})]^{N(\vec{k})}}{\sqrt{N(\vec{k})!}}\,|0\rangle \end{aligned} \tag{2.2}$$

In QFTs the concept of *gauge* symmetry acts an essential role since, by imposing the invariance of the Lagrangian of the system under *local* transformations of the fields, we recover interaction terms $\mathcal{L}_{int}$. As we will see for QED, typically the e.o.m. for an interacting theory with Lagrangian $\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_{int}$ can not be exactly solved, so we have to proceed by exploiting *perturbation theory* and take the interaction contributions small and consider them as perturbation of the free-theory solution.

Interaction Picture

As it is known in quantum theories we have freedom in the choice of the state picture to adopt, and it is due to the $U(1)$ global symmetry in quantum mechanics, that allows us to redefine

the evolution of states and operators in time in different manners. In a quantum field theory formalism the operators corresponds to the quantized fields $\varphi(\vec{x},t), \psi^{\alpha}(x^{\mu}), A_{\nu}(x^{\mu})$, which are written as combination of creation and annihilation of particles within the state $|\psi\rangle$.
In the following dissertation we will focus without loss of generality only on the real scalar field operator $\varphi(\vec{x},t)$. As it was said above, when we switch on an interaction between fields, we can define the total Lagrangian of the system as $\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_I$, and from it we can define the total Hamiltonian of the system $H = H_0 + H_I$.
Then, instead of studying the dynamics of a quantum system via imposing the quantum field to satisfy the Euler-Lagrange equations, we instead assume that it is determined by a Hamiltonian.

The two most used pictures in quantum mechanics are *Schrödinger* and *Heisenberg* pictures. In *Schrödinger* picture the state $|\psi\rangle_s$ and the quantum field operator $\varphi_s(x^{\mu})$ are defined such that, given a time independent Hamiltonian H, they evolve according to:

$$\begin{cases} i\hbar\frac{d}{dt}|\psi(t)\rangle_s = H|\psi(0)\rangle_s \Longrightarrow |\psi(t)\rangle_s = e^{-\frac{i}{\hbar}Ht}|\psi(0)\rangle_s \\ \varphi_s(\vec{x},t) = \varphi_s(\vec{x},0) \end{cases} \tag{2.3}$$

so the unitary evolution operator $U(t) = e^{-iHt/\hbar}$ acts exclusively to the state $|\psi\rangle_s$.
Usually in free quantum field theories we work in the *Heisenberg* picture, which means imposing the following definitions for $|\psi\rangle_h$ and $\varphi_h(x^{\mu})$:

$$\begin{cases} |\psi(t)\rangle_h = |\psi(0)\rangle \equiv U^{\dagger}(t)|\psi(t)\rangle_s \\ \varphi_h(\vec{x},t) = U^{\dagger}(t)\varphi(\vec{x})U(t) \\ \frac{d}{dt}\varphi_h(\vec{x},t) = \frac{i}{\hbar}[H,\varphi_h(\vec{x},t)] \end{cases} \tag{2.4}$$

In this picture the unitary evolution operator acts only on the field $\varphi(\vec{x},t)$, then for $\mathcal{L} = \mathcal{L}_0$ it evolves as a free field, which is exactly solvable: that is why this picture is often used in free theories [37]; on the other hand the state $|\psi\rangle_h$ does not evolve in time.

It is possible to check that these two pictures are equivalent since the expectation value of an observable is the same in both pictures.

Proof. Given a system in the quantum state $|\psi\rangle$ with Hamiltonian H and an observable O we find that:

$$\langle O\rangle_h = \langle\psi(0)|_h O_h(t)|\psi(0)\rangle_h = \langle\psi|_s U(t)U^{\dagger}(t)O_sU(t)U^{\dagger}(t)|\psi\rangle_s = \langle O\rangle_s \tag{2.5}$$

□

In the case of interacting systems the Hamiltonian can be written as the sum of a non-interacting Hamiltonian and an interaction term, that in general might depend on time, $H_{tot}(t) = H_0 + H_{int}(t)$; then it is more convenient to work in another picture called *interaction picture*. In this scheme the evolution gate is defined as $U_0(t) = e^{iH_0t}$, hence the state $|\psi\rangle_I$ and the field operator $\varphi_I(x^{\mu})$ are defined as:

$$\begin{cases} |\psi(t)\rangle_I = U_0^{\dagger}|\psi(t)\rangle_s \\ \varphi_I(\vec{x},t) = U_0^{\dagger}(t)\varphi(\vec{x})U_0(t) \end{cases} \tag{2.6}$$

from (2.6) it follows that $(H_0)_I = H_0$ and that $(H_{int})_I = U_0^{\dagger}(t)H_{int}U_0(t)$, then also the interaction Hamiltonian evolves in time.
We find the following evolution equations:

$$\begin{cases} i\hbar\frac{d}{dt}|\psi(t)\rangle_I = (H_{int})_I|\psi(t)\rangle_I \\ i\hbar\frac{d}{dt}\varphi_I(\vec{x},t) = [\varphi_I(\vec{x},t),H_0] \end{cases} \tag{2.7}$$

From (2.6) and (2.7) it follows two crucial consequences:

- in interaction picture the quantum state $|\psi\rangle_I$ evolves only according to the interaction Hamiltonian $(H_{int})_I = e^{iH_0t/\hbar}H_{int}e^{-iH_0t/\hbar}$; especially with the free theory, with $H_{int} = 0$, the state in the interaction picture is constant in time: $|\psi(t)\rangle_I = |\psi(0)\rangle_I$;
- in interaction picture any quantum field evolves exactly as in free theory, hence they follows the same formalism of the free theory (also the way we canonically quantize them).

The last conclusion is extremely important because, as we previously said, since often the e.o.m. derived from the Lagrangian of an interacting system $\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_{int}$ are not exactly solvable, we are not able to compute and quantize the fields which are solution of the e.o.m..

S-matrix

Thanks to the interaction picture we know that a quantum states evolves according to the *Schrödinger* equation (which implies a unitary evolution) (2.7), with $(H_{int})_I = e^{iH_0t/\hbar}H_{int}e^{-iH_0t/\hbar}$ interaction term (in interaction picture) of the total Hamiltonian of the system. With the formalism discussed above we can then deal with scattering processes.

Given an fully characterized[1] initial state $|\psi(t_i)\rangle = |i\rangle$, which is usually taken long before the scattering occurs (for convention $t_i = -\infty$) so that we can take the initial particles as free fields, we let the particles interact in a period of time $[-T, T]$, with $T < \infty$, and then fly apart again; after a final time $t_f > T$ we can consider the final state $|f\rangle$ as made up of free particles.
From (2.7) we can write that $|i\rangle$ evolved as:

$$|\psi(t_f = \infty)\rangle = S\,|i\rangle \tag{2.8}$$

In general a collision can lead to many different final states $|f\rangle$, and all these possibilities are contained within $S\,|i\rangle$; then the transition probability that after the collision (i.e. at $t_f = +\infty$) the system is in the state $|f\rangle$ is given by:

$$P(|f\rangle) = |\,\langle f|\psi(+\infty)\rangle\,|^2 = |\,\langle f|\,S\,|i\rangle\,|^2 \equiv |S_{if}|^2 \tag{2.9}$$

S represents the unitary evolution of the initial states during a scattering process and is called *S-matrix*.

We can expand the final state $|\psi(+\infty)\rangle$ in term of a complete orthonormal basis:

$$|\psi(+\infty)\rangle = \sum_f |f\rangle\,\langle f|\psi(+\infty)\rangle = \sum_f S_{fi}\,|f\rangle \tag{2.10}$$

Moreover we recall the normalization of the state $|\psi(+\infty)\rangle$, which implies the unitarity of the S-matrix:

$$\begin{aligned}1 = \langle\psi(+\infty)|\psi(+\infty)\rangle &= \sum_f \langle\psi(+\infty)|f\rangle\,\langle f|\psi(+\infty)\rangle = \sum_f \langle i|\,S\,|f\rangle\,\langle f|\,S\,|i\rangle\\ &= \sum_f |S_{if}|^2 \Longrightarrow SS^\dagger = S^\dagger S = \mathbb{1}\end{aligned} \tag{2.11}$$

From (2.7) for the total Hamiltonian $H_{tot} = H_0 + H_{int}$, where the interaction term a perturbative contribution to the free Hamiltonian, we can get that the final state is:

[1] it means that we know how many and which particles are present before the scattering process and what is their momentum, spins, polarization etc... .

$$|\psi(t)\rangle = |i\rangle + (-i)\int_{-\infty}^{t} dt'(H_{int})_I(t')\,|\psi(t')\rangle \tag{2.12}$$

(2.12) can only be solved iteratively as a perturbation series (only for small enough interacting Hamiltonians) in powers of $(H_{int})_I$:

$$|\psi(t)\rangle = \left[\sum_{n=0}^{\infty}(-i)^n\int_{-\infty}^{t}dt_1\int_{-\infty}^{t_1}dt_2...\int_{-\infty}^{t_{n-1}}dt_nH_{int}(t_1)H_{int}(t2)...H_{int}(tn)\right]|i\rangle \tag{2.13}$$

that for $t \longrightarrow \infty$ becomes:

$$\begin{aligned}|\psi(\infty)\rangle &= \left[\sum_{n=0}^{\infty}(-i)^n\int_{-\infty}^{\infty}dt_1\int_{-\infty}^{t_1}dt_2...\int_{-\infty}^{t_{n-1}}dt_nH_{int}(t_1)H_{int}(t2)...H_{int}(tn)\right]|i\rangle \\ &= \left[\sum_{n=0}^{\infty}\frac{(-i)^n}{n!}\int_{-\infty}^{+\infty}dt_1\int_{-\infty}^{+\infty}dt_2...\int_{-\infty}^{+\infty}dt_nT\Big(H_{int}(t_1)H_{int}(t2)...H_{int}(tn)\Big)\right]|i\rangle \\ &= T\Big(e^{-i\int_{-\infty}^{+\infty}dtH_{int}(t)}\Big)\,|i\rangle\end{aligned} \tag{2.14}$$

where $T(A(t_A)B(t_B))$ is the time-ordered product between two operators: $\theta(t_A-t_B)A(t_A)B(t_B)+\theta(t_B-t_A)B(t_B)A(t_A)$. We note that the first term of the perturbation series ($n = 0$) the S-matrix is simply the identity, hence the free theory where no interaction occur. In a covariant manner we can rewrite S as:

$$S = \sum_{n=0}^{\infty}\frac{(-i)^n}{n!}\int\int...\int d^4x_1d^4x_2d^4x_nT\Big\{\mathcal{H}_{int}(x_1)\mathcal{H}_{int}(x_2)...\mathcal{H}_{int}(x_n)\Big\} \tag{2.15}$$

this definition is called Dyson expansion of the S-matrix [38] as time–ordering product of the interaction Hamiltonian density for different spacetime coordinates x^μ. As it is discussed in [37], since the S-matrix is computed perturbatuvely, it is possible to rewrite (2.15) as the identity operator, which corresponds to no interactions between quantum fields, plus a term that describes the deviation from the free theory:

$$\begin{cases}S = \mathbb{1} + i(2\pi)^4\delta^4(\sum p_i^\mu - \sum p_f^\mu)\mathcal{M} \\ \langle f|\,S-\mathbb{1}\,|i\rangle = i(2\pi)^4\delta^4(\sum p_i^\mu - \sum p_f^\mu)\,\langle f|\,\mathcal{M}\,|i\rangle\end{cases} \tag{2.16}$$

where the $\delta^4(\sum p_i^\mu - \sum p_f^\mu)$ imposes the conservation of the total four-momentum during and after the scattering process, and $\mathcal{M}$ is defined as the *scattering amplitude* or simply *matrix element* [37]. The information carried by the scattering amplitude (precisely its modulus squared) describes the dynamical contribution in the cross section (2.1), so in the study of fundamental interactions it is the most interesting observable to compute; that is why its computation through quantum machine learning models will be the central topic of this thesis.

Usually scattering amplitudes are complex to compute, so it is convenient to exploit *Feynman diagrams*, which correspond to a graphical representation of the different terms of the perturbative expansion of the S-matrix. They then correspond to a particular perturbative contribution and are associated to the corresponding mathematical expression, and the complete result is obtained by summing or integrating over all contributing diagrams. This graphical approach simplifies complex calculations since it is broken down into simpler graphical component, called *Feynman rules*.

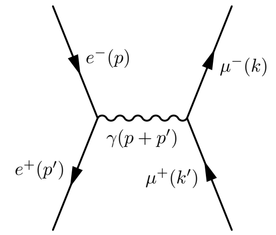


Figure 2.1: Example of Feynman diagram: this diagram corresponds to a QED scattering evaluated at the second order of perturbation theory; the initial states is made of an electron and a positron, which interacts and produce a photon field; the photon then produce a muon and anti-muon in the final state.

Such an expansion of the S-matrix can be used if and only if $\mathcal{L}_{int}$ does not contain terms proportional to the derivative of fields[2]. Because of this constraint it follows that the interacting Hamiltonian density as polynomial in the fields and their adjoints:

$$\mathcal{H}(x) = \sum_i g_i \mathcal{H}_i(x), \tag{2.18}$$

where each term $\mathcal{H}_i$ corresponds to a product of definite numbers of fields and fields adjoints and g_i is the specific *coupling constant* for each term. Then we can rewrite the Dyson expansion as the time-ordering product of fields operator. For example if we take a QFT for only real scalar fields and $\mathcal{H}_{int} = \frac{\lambda}{4!}\varphi^4$ we will have:

$$\begin{cases} \varphi(x) = \varphi^-(x) + \varphi^+(x) \\ T\Big\{\mathcal{H}_{int}(x_1)\mathcal{H}_{int}(x_2)...\mathcal{H}_{int}(x_n)\Big\} = \Big(\frac{\lambda}{4!}\Big)^n T\Big\{\Big(\varphi(x_1)^- + \varphi(x_1)^+\Big)^4 ... \Big(\varphi(x_n)^- + \varphi(x_n)^+\Big)^4\Big\} \end{cases} \tag{2.19}$$

where φ^- and φ^+ are related to creation and annihilation operators a and $a^\dagger$, that will interact with the creation/annihilation operators of the initial and final states from (2.2).

Adiabatic Hypotesis

In the previous section we've recovered that the probability of a specific transition $|i\rangle \longrightarrow |f\rangle$ is given by $\langle f| S |i\rangle$ (2.9), where S can be computed iteratively in perturbation theory according to (2.15). In particular the initial and final states $|i\rangle_I, |f\rangle_I$ are eigenstates of the free Hamiltonian H_0 (hence are *bare* particle states), but when dealing with physical particles it might be a misleading formalism. In fact a physical particle, even though it is not interacting with other particles, but always self-interactions will occur, which means there not exist *bare* particles, but only *dressed* ones. One possible procedure is to exploit the adiabatic approximation:

Definition 21. *Given an Hamiltonian that can be written as: $H_{tot}(t) = H_0 + H_{int}(t)$, where H_0 is a time-independent Hamiltonian and $H_{int}(t)$ is a second time-dependent Hamiltonian that varies slowly in time, we take an initial state $|\psi(t_0)\rangle = |n(t_0)\rangle$, where $|n(t_0)\rangle$ is the n-th instantaneous eigenstate of H_{tot} at time t_0.*
Hence if we let the system slowly evolve from instant t_0 to instant T, the adiabatic approximation states that at time T the state will be in $|n(T)\rangle$, n-th eigenstate of $H_{tot}(T)$.

[2] in fact if $\frac{\partial}{\partial\dot\varphi}\mathcal{L}_-int = 0$ when we compute the Hamiltonian density ad the Legendre transformation of $\mathcal{L}_{int}$ we find:

$$\mathcal{H}_{int} = \frac{\partial}{\partial\dot\varphi}\mathcal{L}_{int}\dot\varphi - \mathcal{L}_{int} = -\mathcal{L}_{int}. \tag{2.17}$$

This approximation can be used for scattering theory since we usually take the initial and final states at times $t_{i,f} \longrightarrow \pm\infty$, very distant from the period of time $[-T, T]$ in which the interaction term is switched on. Hence we can take the interaction Hamiltonian of the form: $H_{int}(t) = H_{int}(t)f(t)$, where $f(t)$ is such that:

$$\begin{cases} f(t) = 1 \quad for \quad t \in [-T, T] \quad \text{with T sufficiently big to let the interaction occur} \\ f(t) \longrightarrow 0 \quad \text{monotonically} \quad \text{for t} \longrightarrow \pm\infty \end{cases} \tag{2.20}$$

In this working scheme the particles in the initial state are taken bare (nonphysical), then for $-\infty < t \leq -T$ the state evolves according to $H_{int}(t)f(t)$ and here the particles become physical (dressed), for $-T \leq t \leq T$ the state evolves with respect to the full interaction Hamiltonian $H_{int}(t)$, while in the interval $-\tau \leq t \leq \tau$ (with $T >> \tau$) the scattering occurs; finally for $T < t < \infty$ the interaction is gradually switched off and the particles get undressed and turn back into bare particles. After all the computaiton is then taken the limit for $T \longrightarrow \infty$.

We still have to figure out how to compute the S-matrix expansion in (2.15) for a particular transition $|i\rangle \longrightarrow |f\rangle$. It is possible to exploit Wick's theorem, which allows us to rewrite the time-ordering operators of Hamiltonian densities as in terms of normal-ordered products and contractions, which are defined as the expectation value on the vacuum state of two operator:

$$\overset{\sqcap}{A(x)B(y)} \equiv \langle 0| T\{A(x)B(y)\} |0\rangle \tag{2.21}$$

The theorem and some computations are exhaustively shown in [39]. In QFT contracted fields corresponds to *propagators* or *virtual particles*, which are unphysical particles that connects two interactions vertices of the Feynman diagrams; those particles are unphysical since they are not on-shell, so $E^2 \neq p^2 + m^2$, where E is the total energy of the particle, p is the momentum modulus and m is the mass of the particle.

2.2 Quantum Electrodynamics

The concept of gauge symmetries, as discussed in sec.2.1.1, play a central role in QFT since can be proven that all the fundamental interactions (also gravity) can be fully explained by gauge theories, field theories that imposes specific local group symmetry on the Lagrangian of the system. In order to introduce *Quantum Electrodynamics (QED)* we can work on the Dirac Lagrangian for a spinor field ψ with mass m:

$$\mathcal{L}_D = \bar{\psi}(i\partial\!\!\!/ - m)\psi \tag{2.22}$$

it is easy to check that this Lagrangian density is invariant under $U(1)$ global transformation of the spinorial field: $\psi \longrightarrow e^{i\alpha}\psi$ and $\bar{\psi} \longrightarrow e^{-i\alpha}\bar{\psi}$; according to Noether theorem, there exist a conserved quantity associated to such a symmetry, and this quantity corresponds to the electric charge of the total electric charge of the system. We now want to impose a $U(1)$ local symmetry on the Dirac Lagrangian; it is also easy to prove that the Dirac Lagrangian as defined in 2.22 is not invariant under local $U(1)$ fields transformations of the form:

$$\begin{cases} \psi \longrightarrow \psi' = e^{i\alpha(x)}\psi \\ \bar{\psi} \longrightarrow \bar{\psi}' = e^{-i\alpha(x)}\bar{\psi} \end{cases} \tag{2.23}$$

where the phase $\alpha(x)$ is a function of the spacetime coordinate x^μ.

Proof. We compute the Dirac Lagrangian density $\mathcal{L}'_D$ for the fields $\psi' = e^{i\alpha(x)}\psi$ and $\bar{\psi}' = e^{-i\alpha(x)}$ with mass m:

$$\begin{aligned}\mathcal{L}'_D(\psi', \bar{\psi}') &= \bar{\psi}'(i\not{\partial} - m)\psi' = \bar{\psi}e^{-i\alpha(x)}(i\not{\partial} - m)e^{i\alpha(x)}\psi \\ &= \bar{\psi}e^{-i\alpha(x)}i\gamma^\mu[\partial_\mu(e^{i\alpha(x)}) + ie^{i\alpha(x)}\gamma^\mu\partial_\mu]\psi - m\bar{\psi}\psi \\ &= \bar{\psi}(i\not{\partial} - m)\psi - \partial_\mu(\alpha(x))\bar{\psi}\gamma^\mu\psi \neq \mathcal{L}_D(\bar{\psi}, \psi)\end{aligned} \tag{2.24}$$

□

In order to build an invariant $\mathcal{L}_D$ under $U(1)$ local transformations, we can redefine the derivative ∂_μ by adding a second term proportional to a bosonic field A_μ; hence we can define the covariant derivative as: $D_\mu = \partial_\mu - ieA_\mu$. If we replace $\partial_\mu \longrightarrow D_\mu$ in (2.22) we then get:

$$\mathcal{L} = \bar{\psi}(i\not{D} - m)\psi = \mathcal{L}_D + e\bar{\psi}\gamma^\mu\psi A_\mu \tag{2.25}$$

We also have to change the $U(1)$ gauge transformation laws in:

$$\begin{cases}\psi \longrightarrow \psi' = e^{i\alpha(x)}\psi \\ \bar{\psi} \longrightarrow \bar{\psi}' = e^{-i\alpha(x)}\bar{\psi} \\ A_\mu \longrightarrow A'_\mu = A_\mu + \frac{1}{e}\partial_\mu\alpha(x)\end{cases} \tag{2.26}$$

Then it is easy to prove that (2.28) is invariant under field transformation in (2.26):

Proof.

$$\begin{aligned}\mathcal{L}'(\psi', \bar{\psi}') &= \bar{\psi}'(i\not{D}' - m)\psi' = \bar{\psi}e^{-i\alpha(x)}i\gamma_\mu[\partial_\mu - ieA'_\mu]e^{i\alpha(x)}\psi - m\bar{\psi}\psi \\ &= \bar{\psi}e^{-i\alpha(x)}i\gamma^\mu[\partial_\mu(e^{i\alpha(x)}) + e^{i\alpha(x)}\partial_\mu - iee^{i\alpha(x)}A_\mu - ie^{i\alpha(x)}\partial_\mu(\alpha(x))]\psi - m\bar{\psi}\psi \\ &= \bar{\psi}i\gamma_\mu[i\partial_\mu(\alpha(x)) - ieA_\mu - i\partial_\mu(\alpha(x))]\psi - m\bar{\psi}\psi \\ &= \bar{\psi}[i\gamma_\mu(\partial_\mu - ieA_\mu) - m]\psi = \bar{\psi}[i\not{D} - m]\psi = \mathcal{L}(\psi, \bar{\psi})\end{aligned} \tag{2.27}$$

□

We recognize in (2.26) the same gauge transformation laws of the electromagnetic field; hence it is possible to write the full Lagrangian density for a $U(1)$ gauge symmetry theory, that corresponds to electrodynamics, as:

$$\mathcal{L}_{QED} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + \bar{\psi}(i\not{D} - m)\psi = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + e\bar{\psi}\gamma^\mu\psi A_\mu \tag{2.28}$$

where $F^{\mu\nu} \equiv \partial_\mu A_\nu - \partial_\nu A_\mu$ is the *strength field tensor* of the electromagnetic field A_μ. The term $e\bar{\psi}\gamma^\mu\psi A_\mu$ is the interacting Lagrangian $\mathcal{L}_{int}$. The e.o.m of the system are:

$$\begin{cases}\partial_\mu F^{\mu\nu} = e\bar{\psi}\gamma^\nu\psi \\ (i\gamma^\mu D_\mu - m)\psi = 0 \Longrightarrow (i\gamma^\mu\partial_\mu - m)\psi + e\gamma^\mu A_\mu\psi = 0 \\ -i\partial_\mu\bar{\psi}\gamma^\mu - m\bar{\psi} + e\bar{\psi}A_\mu\gamma^\mu = 0\end{cases} \tag{2.29}$$

The e.o.m. in (2.29) are not exactly solvable, hence we have to work with perturbation theory and its formalism, that was exposed in sec.2.1.1. For the Lagrangian $\mathcal{L}_{int} =: e\bar{\psi}\not{A}\psi :$ we can write the Dyson expansion of the S-matrix in (2.15) as:

$$S = \mathbb{1} + ie\int d^4x : \bar{\psi}\not{A}\psi : + \frac{(ie)^2}{2!}\int d^4x d^4y T\Big\{ : \bar{\psi}\not{A}\psi :_x : \bar{\psi}\not{A}\psi :_y \Big\} + ... \tag{2.30}$$

where : ⁝ is the normal ordering of the operator, which sorts the operators so that the creation operators are on the left, and the annihilation operators are on the right.

Before retrieving the Feynman rules for QED we want to show an interesting property of QED, which namely is the fact that the first order of perturbation theory of the S-matrix is null, namely $S^{(1)} = 0$:

$$S^{(1)} = ie \int d^4x : \bar{\psi} \not{A} \psi :_x \tag{2.31}$$

we can develop each field in positive and negative frequencies:

$$: (\bar{\psi}^{(+)} + \bar{\psi}^{(-)})(\not{A}^{(+)} + \not{A}^{(-)})(\psi^{(+)} + \psi^{(-)}) : \tag{2.32}$$

where in general the positive frequencies annihilate a particle and the negative ones create a particle. By expanding (2.32) we find the following 8 possible diagrams, which correspond to any combination of the interaction vertex.

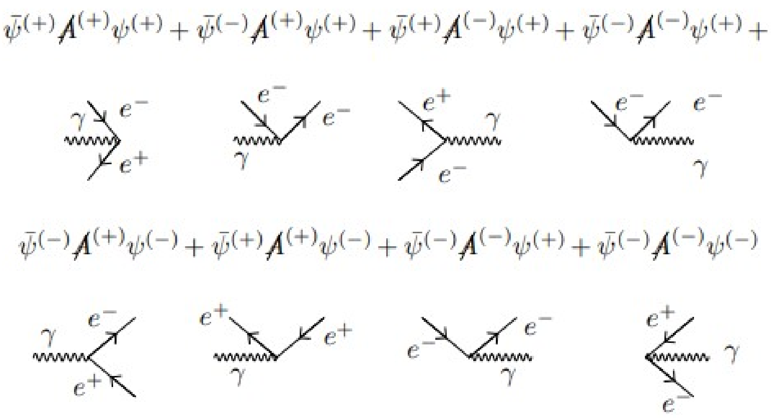

$\bar{\psi}^{(+)}\not{A}^{(+)}\psi^{(+)} + \bar{\psi}^{(-)}\not{A}^{(+)}\psi^{(+)} + \bar{\psi}^{(+)}\not{A}^{(-)}\psi^{(+)} + \bar{\psi}^{(-)}\not{A}^{(-)}\psi^{(+)} +$
γ
e⁻
e⁺
e⁻
e⁻
γ
e⁺
γ
e⁻
e⁻
e⁻
γ
$\bar{\psi}^{(-)}\not{A}^{(+)}\psi^{(-)} + \bar{\psi}^{(+)}\not{A}^{(+)}\psi^{(-)} + \bar{\psi}^{(-)}\not{A}^{(-)}\psi^{(+)} + \bar{\psi}^{(-)}\not{A}^{(-)}\psi^{(-)}$
γ
e⁻
e⁺
e⁺
e⁺
γ
e⁻
e⁻
γ
e⁺
γ
e⁻

Figure 2.2: All possible diagrams derived from the first perturbative order of the S-matrix in QED.

Proof. Since $\mathcal{M}$ is a Lorentz scalar, it follows that it is invariant under translation transformations, which mean we have to impose:

$$\begin{gathered} [p^\mu, S^{(1)}] = 0 \\ \langle f | \, S^{(1)} p^\mu - p^\mu S^{(1)} \, | i \rangle = (p_i - p_f)^\mu \, \langle f | \, S^{(1)} \, | i \rangle = 0 \end{gathered} \tag{2.33}$$

We have to take into account two cases:

$$\begin{cases} p_i^\mu \neq p_f^\mu \Longrightarrow S_{fi} = 0 \\ p_i^\mu = p_f^\mu \Longleftrightarrow S_{fi} \neq 0 \end{cases} \tag{2.34}$$

but the last case implies that, given p_μ, p'_μ the momenta of the electron and positron and k_μ the momentum of the photon:

$$p_\mu + k_\mu = p'_\mu \longrightarrow (p_\mu + k_\mu)^2 = p'^2 \longrightarrow m^2 + 2p \cdot k = m^2 \longrightarrow p \cdot k = 0 \tag{2.35}$$

if we work in the centre of mass framework we have $p^\mu = (E, \vec{p})$ and $k^\mu = (|\vec{p}|, -\vec{p})$, so that the previous identity becomes:

$$p \cdot k = E|\vec{p}| + |\vec{p}|^2 = 0 \tag{2.36}$$

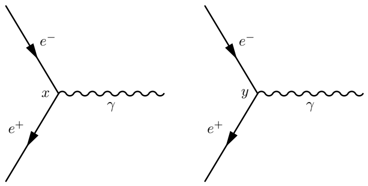
which unique solution is given by $\vec{p} = 0$, but this is not admitted since the centre of rest of the photon does not exist[3].
From this consideration it follows that in QED $S^{(1)} = 0$. □

Hence we can now focus on the second order of perturbation theory of the scattering amplitude

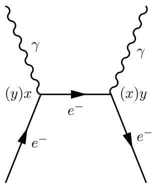
$$S^{(2)} = \frac{(ie)^2}{2!} \int d^4x d^4y T\Big\{ :\bar{\psi}\not{A}\psi :_x : \bar{\psi}\not{A}\psi :_y \Big\} \tag{2.37}$$

By using Wick's theorem we can write (2.37) as the sum of 6 contribution; some of them correspond to all possible QED 2-particles⟶2-particles scattering processes in the leading-order of perturbation theory, also called *tree level diagrams.* Any of these contributions are symmetric under exchange of the x, y coordinates.

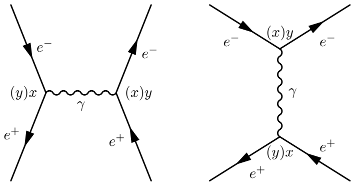
- $: \bar{\psi}\not{A}\psi(x)\bar{\psi}\not{A}\psi(y) :$, this term is null since it corresponds to two disconnected diagrams from the first order of perturbation theory



- $\overbracket{\bar{\psi}(x)\psi(y)} : \not{A}\psi(x)\bar{\psi}\not{A}(y) : +\overbracket{\psi(x)\bar{\psi}(y)} : \bar{\psi}\not{A}(x)\not{A}\psi(y) :$, this term corresponds to the *Compton scattering* $e^-\gamma \longrightarrow e^-\gamma$ (the second diagram is missing because the compiler was not able to draw it).

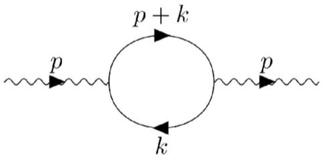


- $\overbracket{\not{A}(x)\not{A}(y)} : \bar{\psi}\psi(x)\bar{\psi}\psi(y) :$ corresponds to Bhabha scattering $e^+e^- \longrightarrow e^+e^-$, the scattering we will focus on during this thesis project. We have two different Feynman diagram for this process.

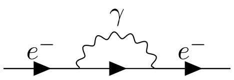


- $\overbracket{\bar{\psi}(x)\psi(y)}\overbracket{\psi(x)\bar{\psi}(y)} : \not{A}(x)\not{A}(y) :$ is the vacuum polarization contribution, a 1-loop Feynman diagram.

[3] this is due to the fact that the photon travels at the speed of light, hence it would be necessary an infinite amount of energy to accelerate to c, where the photon is at rest



- $\overbracket{\bar{\psi}(x)\psi(y)}\overbracket{\not{A}(x)\not{A}(y)} : \psi(x)\bar{\psi}(y) : + \overbracket{\psi(x)\bar{\psi}(y)}\overbracket{\not{A}(x)\not{A}(y)} : \bar{\psi}(x)\psi(y) :$ is called the electron self energy diagram, a 1-loop contribution too.



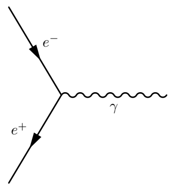
- $\overbracket{\bar{\psi}(x)\psi(y)}\overbracket{\psi(x)\bar{\psi}(x)}\overbracket{\not{A}(x)\not{A}(y)}$ is a null contribution since we have the maximum number of field contractions allowed, hence we do not have either initial or final particle, this is not a process. this contribution is called also vacuum diagram.

In this thesis we will focus on Bhabha scattering Feynman diagrams. Until now we have worked with only one fermionic field, conventionally representing the electron and the positron; however it is possible to generalize QED interactions to all the fermionic fields, namely leptons and quarks. The QED Lagrangian for the leptonic sector then becomes:

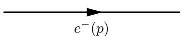
$$\mathcal{L} = \sum_{j=1}^{3} \left[\bar{\psi}_j (i\not{\partial} - m_j)\psi_j + e\bar{\psi}_j \not{A}\psi_j \right] \tag{2.38}$$

For the purposes of the thesis we will focus only on the computation of three different Feynman diagrams at tree level (second order in perturbation theory): *Bhabha scattering: t-channel and s-channel* and *muon pair production* $e^+e^- \longrightarrow \mu^-\mu^+$; before doing that we have to formulate the *Feynman rules* for QED:

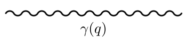
- *interaction vertex*: $ie\gamma_\mu$



- *fermionic propagator* (fermionic internal line): $\frac{i}{\not{p}-m}$



- *photon propagator* (photonic internal line): $\frac{-ig_{\mu\nu}}{q^2}$



- *external lines*:

e^- in	p —▶—●	$u_s(p)$
e^- in	p ●—▶—	$\bar{u}_s(p)$
e^+ in	p —◀—●	$\bar{v}_s(p)$
e^+ out	p ●—◀—	$v_s(p)$
γ in	k ∿∿∿●	$\varepsilon_s(k)$
γ out	k ●∿∿∿	$\varepsilon_s(k)$

where s indicates the spin for fermions and the polarization for the photon.

where $e = 4\pi\alpha\epsilon_0/(\hbar c) \sim 1.6 \times 10^{-19} C$[4] is the modulus of the charge of the electron, γ_μ is the four-vector of the Dirac matrices and $g_{\mu\nu}$ is the Minkowski metric with signature $(1, -1, -1, -1)$. We also chose the following normalization and completeness conditions for the spinor fields:

$$\begin{cases} \bar{u}_s(p)u_r(p) = 2m\delta_{rs} \\ \sum_s u_s(p)\bar{u}_s(p) = \gamma_\mu p^\mu + m \end{cases} \qquad \begin{cases} \bar{v}_s(p)v_r(p) = -2m\delta_{rs} \\ \sum_s v_s(p)\bar{v}_s(p) = \gamma_\mu p^\mu - m \end{cases} \tag{2.39}$$

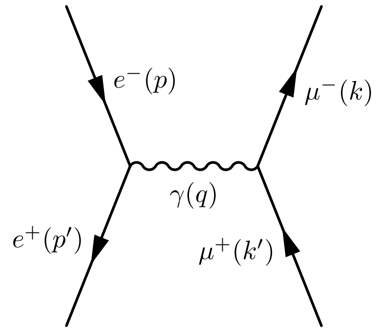


Figure 2.3

Figure 2.4: Tree level Feynman diagram related to muon pair production.

2.2.1 Muon pair production

Now it is possible to write down the matrix element referred to the scattering process $e^+e^- \longrightarrow \mu^+\mu^-$ at tree level.

[4] $\alpha \sim 1/137$ is the fine-structure constant, $\hbar = h/(2\pi) \sim 1.05 \times 10^{-34} J \cdot s$ is the reduced Planck constant, $c \sim 3 \times 10^8 m/s$ is the speed of light in the vacuum and $\epsilon_0 \sim 8.85 \times 10^{-12} \frac{C^2}{N \cdot m^2}$ is the vacuum electric permeability constant.

$$\mathcal{M} = -\bar{u}_\mu(k)(ie\gamma_\mu)v(k')_\mu \frac{-ig^{\mu\nu}}{(p+p')^2}\bar{v}_e(p')(ie\gamma_\nu)u_e(p) \tag{2.40}$$

We are interested in computing $|\bar{\mathcal{M}}|^2$ in the non-polarized case, which means average over initial spin numbers and sum the final ones, i.e.:

$$|\bar{\mathcal{M}}|^2 = \frac{1}{N_r}\sum_r\sum_s |\mathcal{M}|^2 \tag{2.41}$$

where r corresponds to initial spin numbers and s to the final ones. If we now insert (2.40) in (2.41), we are now able to write the expression of $|\bar{\mathcal{M}}|^2$ in terms of traces along spinor indices:

$$\frac{e^4}{4}\frac{1}{(p+p')^4}Tr\Big[(\not{k}+M_\mu)\gamma_\nu(\not{k}'-M_\mu)\gamma_\alpha\Big]Tr\Big[(\not{p}'-m_e)\gamma^\nu(\not{p}-m_e)\gamma^\alpha\Big] \tag{2.42}$$

where M_μ and m_e are the muon and electron masses respectively. Now we exploit the properties of Dirac matrices and we get to:

$$|\bar{\mathcal{M}}|^2 = \frac{4e^4}{(p+p')^4}\Big[2(p\cdot k')(p'\cdot k) + 2(p\cdot k)(p'\cdot k') + 2M_\mu^2(p\cdot p') + 2(k\cdot k')m_e^2 + 4m_e^2M_\mu^2\Big] \tag{2.43}$$

where $(a\cdot b) \equiv a^\nu b_\nu = a_\nu b^\nu$ is the relativistic scalar product. Remind that QFT is invariant under transformations of the *Lorentz* group $SO(1,3)$ (in Minkoswki spacetime with signature $(+,-,-,-)$), then we are free to set the system frame, that we choose to be the rest frame (or frame of the centre of mass), where the following relation holds:

$$\vec{p} + \vec{p'} = 0 \tag{2.44}$$

which means that both incoming and outcoming particles must travel in opposite directions. From the conservation of the total four-momentum vector we have that:

$$0 = p^\nu + p'^\nu = k^\nu + k'^\nu \tag{2.45}$$

Finally if we combine (2.44) and (2.45) we can define the following four-momentum vectors of the particles:

$$\begin{cases} p^\nu = (\sqrt{p^2+m_e^2}, 0, 0, p) \\ p'^\nu = (\sqrt{p^2+m_e^2}, 0, 0, -p) \\ k^\nu = (\sqrt{k^2+M_\mu^2}, 0, k\cdot sin(\theta), k\cdot cos(\theta)) \\ k'^\nu = (\sqrt{k^2+M_\mu^2}, 0, -k\cdot sin(\theta), -k\cdot cos(\theta)) \\ where \quad k^2 = p^2 + m_e^2 + M_\mu^2 \end{cases} \tag{2.46}$$

From total energy-conservation it follows that the process has a *energy threshold* to satisfy, which means the initial particles have to be enough energy to produce the muon pair at the final state; in the centre of mass frame the energy threshold is simply equal to the masses of the muons (production at rest). So we have the following constraint:

$$m_e^2 + p^2 = M_\mu^2 \Longrightarrow p = \sqrt{M_\mu^2 - m_e^2} \tag{2.47}$$

Even though we are solving the problem in $(3+1)$ dimensions, because of Lorentz-invariance we are free to set a planar reference system and work only in two dimensions (y, z axis for cartesian coordinates and r, θ for spherical coordinates). By inserting (2.46) in (2.43) and doing all the calculations we find the final result:

$$|\bar{\mathcal{M}}|^2 = e^4\Big[1 + p^2cos^2(\theta)\Big(\frac{1}{p^2+m_e^2} - \frac{M_\mu^2}{(p^2+m_e^2)^2}\Big) + \frac{m_e^2+M_\mu^2}{4(p^2+m_e^2)}\Big] \tag{2.48}$$

In high energy limit it is possible to consider both particles as massless, then (2.48) becomes:

$$|\mathcal{M}|^2_{m,M=0} = e^4\Big[1 + cos^2(\theta)\Big] \tag{2.49}$$

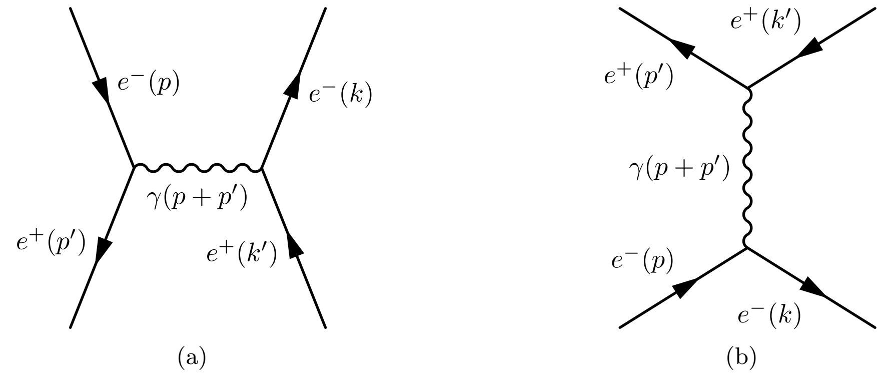


Figure 2.5: Tree level Feynman diagrams corresponding to Bhabha scattering.

2.2.2 Bhabha Scattering

At tree level Bhabha scattering has two different channels, related to two *topologically independent* Feynman diagrams. It is possible then to write the total amplitude $\mathcal{M}_{tot}$ for this process as the sum of the amplitudes of these two diagrams $\mathcal{M}_s$ and $\mathcal{M}_t$:

$$\mathcal{M}_{tot} = \mathcal{M}_s + \mathcal{M}_t \tag{2.50}$$

the modulus squared of the full unpolarized amplitude is then:

$$|\bar{\mathcal{M}_{tot}}|^2 = |\bar{\mathcal{M}_s}|^2 + |\bar{\mathcal{M}_t}|^2 + \mathcal{M}_s\bar{\mathcal{M}}_t^* + \mathcal{M}_t\bar{\mathcal{M}}_s^* = |\bar{\mathcal{M}_s}|^2 + |\bar{\mathcal{M}_t}|^2 + 2Re(\mathcal{M}_s\bar{\mathcal{M}}_t^*) \tag{2.51}$$

where we identify the term $2Re(\mathcal{M}_s\bar{\mathcal{M}}_t^*)$ as the interference term between the two channels, a key property that emerges in QFT.
From the QED Feynman rules it is possible to write down the amplitudes of the two channels in the space of momenta:

$$\begin{aligned}\mathcal{M}_t &= \bar{u}_e(k)(ie\gamma_\mu)u_e(p)\frac{-ig^{\mu\nu}}{(p-k)^2}\bar{v}_e(p')(ie\gamma_\nu)v_e(k')\\ \mathcal{M}_s &= -\bar{u}_e(k)(ie\gamma_\mu)v_e(k')\frac{-ig^{\mu\nu}}{(p+p')^2}\bar{v}_e(p')(ie\gamma_\nu)u_e(p)\end{aligned} \tag{2.52}$$

From (2.52) we can compute the amplitude squared for the t-channel:

$$\begin{aligned}|\mathcal{M}_t|^2 &= \frac{1}{4}\sum_{init}\sum_{fin}|\mathcal{M}_t|^2 = \frac{e^4}{4}\frac{1}{(p-k)^4}Tr\Big[(\not{k}+m_e)\gamma_\mu(\not{p}+m_e)\gamma_\alpha\Big]Tr\Big[(\not{p}'-m_e)\gamma^\mu(\not{k}'-m_e)\gamma^\alpha\Big]\\ &= \frac{e^4}{4(p-k)^4}Tr\Big[\not{k}\gamma_\mu\not{p}\gamma_\alpha + m_e^2\gamma_\mu\gamma_\alpha\Big]Tr\Big[\not{k}'\gamma^\mu\not{p}'\gamma^\alpha + m_e^2\gamma_\mu\gamma^\alpha\Big]\end{aligned} \tag{2.53}$$

By using the trace identites for the Dirac matrices we get to:

$$\frac{4e^4}{(p-k)^4}\Big[2(k\cdot k')(p\cdot p')+2(p\cdot k')(p'\cdot k)-2m_e^2(p\cdot k)+4m_e^4\Big] \tag{2.54}$$

As in the previous computation we are free to choose the reference system, so we choose to work in the centre of mass system, where by definition holds (2.44). We also choose to work on a planar system; then the four-momenta of the particles are:

$$\begin{cases} p^\nu=(\sqrt{p^2+m_e^2},0,0,p)\\ p'^\nu=(\sqrt{p^2+m_e^2},0,0,-p)\\ k^\nu=(\sqrt{p^2+m_e^2},0,p\cdot sin(\theta),p\cdot cos(\theta))\\ k'^\nu=(\sqrt{p^2+m_e^2},0,-p\cdot sin(\theta),-p\cdot cos(\theta)) \end{cases} \tag{2.55}$$

By inserting the four-momenta in (2.54) we get to the final result:

$$|\bar{\mathcal{M}}_t|^2=\frac{2e^4}{p^4(1-cos(\theta))^2}\Big[(p^2(1+cos(\theta))+2m^2)^2+2(2p^4-m^4)\Big] \tag{2.56}$$

that for high energies, where it is possible to consider the electrons massless, we get:

$$|\bar{\mathcal{M}}_t|^2{}_{m=0}=\frac{2e^4}{(1-cos(\theta))^2}[4+(1+cos(\theta))^2] \tag{2.57}$$

In both limits (2.56) and (2.57) there are divergences in $|\bar{\mathcal{M}}_t|^2$ at $\theta=0$ (called collinear divergence) and at $p=0$ in the low energy limit.

We note that the Bhabha s-channel Feynman diagram is the same as the muon pair production one, but a particle exchange $e\longleftrightarrow\mu$, which means that the corresponding amplitudes modulus squared will be the same but with an exchange if masses $M_\mu\longleftrightarrow m_e$. The only difference for this channel is that it can occur at any energy, the energy constraint is always satisfied.

$$\begin{aligned} |\bar{\mathcal{M}}_s|^2&=e^4\left[1+p^2cos^2(\theta)\left(\frac{1}{p^2+m_e^2}-\frac{m_e^2}{(p^2+m_e^2)^2}\right)+\frac{m_e^2}{2(p^2+m_e^2)}\right]\\ &=e^4\left[1+\frac{p^4cos^2(\theta)}{(p^2+m_e^2)^2}+\frac{m_e^2}{2(p^2+m_e^2)}\right] \end{aligned} \tag{2.58}$$

that, for higher energy, where I can consider the electron massless, we find:

$$|\bar{\mathcal{M}}_s|^2=e^4\big[1+p^2cos^2(\theta)\big] \tag{2.59}$$

which is the same result as in (2.49). This is due to an accidental $U(3)$ leptonic flavour symmetry that is softly broken because of the different masses of the leptons.

We now conclude the computation by evaluating the interference term between the two channels $2Re(M_s\bar{M}_t^*)$:

$$\begin{aligned} M_s\bar{M}_t^*&=\frac{1}{4}\sum_{in}\sum_{fin}M_sM_t^*\\ &=\frac{e^4}{4}\sum_{init}\sum_{fin}\bar{u}_e(k)\gamma_\mu v_e(k')\frac{-i}{(p+p')^2}\bar{v}_e(p')\gamma^\mu u_e(p)\bar{v}_e(k')\gamma_\alpha v_e(p')\frac{i}{(p-k)^2}\bar{u}_e(p)\gamma^\alpha u_e(k) \end{aligned} \tag{2.60}$$

In order to continue the computation it is necessary to rearrange spinors (we know that spinors are Grassmannian fields, hence they safisty anticommuting relations), so that it is possible to exploit the completeness relation in (2.39). So we get:

$$\begin{aligned} M_s\bar{M}_t^* &= \frac{-e^4}{4(p+p')^2(p-k)^2}\sum_{init}\sum_{fin}\bar{u}_e(k)\gamma_\mu v_e(k')\bar{v}_e(k')\gamma_\alpha v_e(p')\bar{v}_e(p')\gamma^\mu u_e(p)\bar{u}_e(p)\gamma^\alpha u_e(k) \\ &= \frac{-e^4}{4(p+p')^2(p-k)^2}Tr\Big[(\not{k}+m_e)\gamma_\mu(\not{k}'-m_e)\gamma_\alpha(\not{p}'-m_e)\gamma^\mu(\not{p}+m_e)\gamma^\alpha\Big] \end{aligned} \tag{2.61}$$

for this computation it is very useful to work with the Mandelstam variables *u,s,t*, which from identities (2.55) are defined as follow:

$$\begin{cases} s=(p+p')^2=(k+k')^2=4(p^2+m^2) \\ t=(p-k)^2=(p'-k')^2=2p^2(cos(\theta)-1) \\ u=(p-k')^2=(p'-k)^2=-2p^2(cos(\theta)+1) \end{cases} \tag{2.62}$$

with these new variables we can rewrite the interference term in a more compact way:

$$\begin{aligned} 2Re(M_s\bar{M}_t^*) &= \frac{-2e^4}{2st}[-8u^2+64m_e^2u-96m_e^4] \\ &= \frac{e^4}{(p^2+m_e^2)p^2(1-cos(\theta))}\big[-p^4(cos(\theta)+1)-4m_e^2p^2(cos(\theta)+1)-3m_e^4\big] \end{aligned} \tag{2.63}$$

For $p^2 >> m_e^2$ we can take $m_e=0$, so the interference term becomes:

$$2Re(M_s\bar{M}_t^*) = -\frac{2e^4(cos(\theta)+1)}{1-cos(\theta)} \tag{2.64}$$

In this chapter we have introduced the basic and fundamental concepts of scattering theory and QED, that will become the central topics of our thesis work, that tries to predict the scattering amplitude squared associated to a given Feynman diagram. This chapter was intended to introduce the necessary concepts of quantum field theory and scattering theory for the understanding of this thesis work; for a more extensive and in-depth discussion on QFT, reference can be made to manuals such as [40, 41, 37, 39].

Chapter 3

Feynman Diagram Learning: Methods and Results

In this last chapter we discuss the strategies we implemented and the achieved results in predicting scattering amplitudes modulus squared with QGNNs. We are focusing on tree level Feynman Diagrams concerning pure QED interaction, both in high energy limit (therefore considering massless particles) and massive-particles limit. In this dissertation all the quark sector of the *Standard Model* is ignored without loss of generality.

The model we are presenting is a *Quantum Graph Neural Network* as a *Parametrized Quantum Circuit* (PQC) that is going to be trained with a dataset of Feynman diagrams evaluated for different kinematic variables, namely the momenta of incoming and outcoming particles and the scattering angle between the two of them. Here we show that the procedure is completely general and can be used for studying a generic tree level Feynman diagram.
In sec.3.6 we will discuss all the problems and limitations about our work, especially the trainability issues when extending the model for predicting scattering amplitudes modulus squared for multi-diagram datasets.

Finally we tried to design a quantum circuit that was able to compute the interference term of two Feynman diagram; this could be interesting in order to compute the total amplitude modulus squared for scattering processes where more than one diagram is involved. The main idea was to mix two pre-trained QGNNs, each of them trained on a specific Feynman diagram, in the same circuit and by exploiting superposition of the states compute the interference term; however, no satisfactory results were found, which shows that this circuit still needs to be developed and improveds.
We developed our algorithms by leveraging Pennylane [42], an Python library developed by Xanadu, which provides an agile framewok for manipulating many-qubit states, both via quantun simulators and hardware interface; we preferred to work on state-vector qubit simulators because the architecture for both the *QGNN* and the interference-extracting protocol would be too complex to run on a NISQ quantum device without heavy error effects on the computation; another property of the state-vector simulator is that it simulates a noiseless circuit and it computes exact expectation values.

3.1 Reasons of QML for QFT problems

As showed in chapter 2, one of most important observables in QFT associated with a scattering process is the scattering amplitude modulus squared, which represents the probability of the process itself to occur. Conventionally, in order to make predictions of very complex processes (such as hadron collisions, non perturbative QCD computations, high order perturbative calculations) we have to rely on both analytical computations in the context of Quantum Field Theory (e.g Mathematica tools such as FeynCalc [43], FeynArts [44], FromCalc [45], LoopTools [46]) and computationally expensive simulations (e.g. MadGraph [47], Pythia [48] and CalcHEP [49]), that involves the generation of a large number of random events to approximate particle behavior and interactions.

In recent years there has been a growing interest in exploring alternative approaches using machine learning techniques based on deep learning [50, 51, 52] and graph neural networks [8], which results in a speed-up of the computational cost of the algorithm, to trade with a slight loss of the accuracy of the results. An interesting fact is that machine learning techniques are able to extrapolate by their own new properties of the system that were hidden in the data or even unknown, as we can see in [52], in which a deep neural network (DNN) was able to learn conservation laws of optics; this is due to their strong ability of generalization of the problem and identification of hidden correlations in data.

Another promising field of application of machine learning is anomaly detection in LHC hadronic processes [53, 54], which aims to spot anomaly events in the distribution of the data that could be due to new physics beyond the Standard Model; this field of study can be useful because it tries to give an hint on new physics by learning from actual data, so it can be done nowadays while waiting for higher performance technologies and colliders.

Quantum computing and quantum field theory are described by the same principles of quantum mechanics, so performing computations on quantum computers seems the most natural way to study quantum system and quantum processes; predicting quantum system with quantum computers might indeed not only ensure a speedup in the computation, but can result simpler and more efficient because we encode quantum information into quantum states, which are then processed via quantum operations, so that we do not incur into limitations in the computation due to the fact that we are representing a quantum state as classical pieces of information.

The potential benefits of employing quantum machine learning models for such tasks, driven by the natural connection between the quantum framework and the problem domain, are possible, but effective solutions remain an open challenge.

3.2 Choice of the Model

As we discussed in 2, the scattering amplitude modulus squared $|\mathcal{M}|^2$ is the term in (2.1) that represents the probability of the system to go from an initial state $|i\rangle$ to a specific final state $|f\rangle$, and in it all the dynamics of the process is encoded; We also discussed in 2 that scattering amplitudes are strictly related to Feynman diagrams.

The goal of our thesis is to compute with QML techniques the scattering amplitude modulus squared from the topology of its associated Feynman diagram and the other essential variables that characterizes the scattering process. Since $|\mathcal{M}|^2$ is a function of the kinematic variables, we restrict our dissertation to quantum supervised regression models, namely models that aim to learn a mapping between input data and a continuous output function. Quantum Neural Networks are excellent models for this type of problem, so we have decided to develop a problem-inspired circuit; in fact, from the definition of Feynman diagram given in chapter 2, it is straightforward to encode all the information in a labelled graph and process it with a *Quantum Graph Neural Network.*

3.3 Description of the Dataset

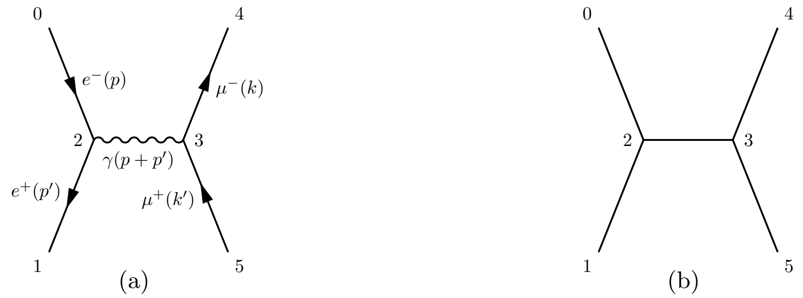
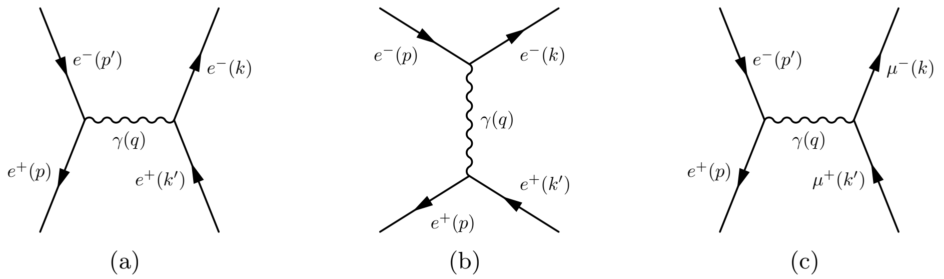


Figure 3.1: Examples of our framework, we encode a generic Feynman diagram (3.1a) as its corresponding graph (3.1b).

Since we are focusing on a quantum supervised regression model, the single element of the dataset is composed by a tuple of the input data $\vec{x_i}$ and by the true value of the target function $\hat{y}_i$, which is the theoretical value of $|\bar{\mathcal{M}}|^2$ (unpolarized scattering amplitude modulus squared); in this thesis, which represents a preliminary study of simple application of this model, we will deal with tree level Feynman diagrams, for which then is easy to do the full calculation and analytically determine $|\bar{\mathcal{M}}|^2$; on the other hand a supervised approach may not be optimal for more complex Feynman diagrams, for which it is not always easy to analytically compute the scattering amplitude modulus square.
Benchmark datasets of Feynman diagrams and their corresponding matrix elements do not exist, then the first step of this project was dedicated to construct synthetic datasets for a set of different tree level Feynman diagrams. So it was necessary to comprehend which pieces of information we had to encode and process in the quantum model.
First of all we need to efficiently encode information about the particles involved in the process; in QED the quantum numbers that fully describe a particle are: mass m, electric charge Q and spin (or polarization) S; when enlarging the study for different interactions we simply change the set of quantum numbers (es if dealing with EW interactions simply get the weak isospin and the hypercharge). In sec.2.2 we saw that for simple we computed the scattering amplitude modulus squared for the diagram in fig.3.1a, and in (2.48) we see that $\bar{\mathcal{M}}|^2$ depends only on three kinematic variables, namely the scattering angles between incoming and outcoming particles θ and the particles' modulus momenta k, p; to be more precise from (2.48) one gets that k and p variables are indeed correlated, hence one of the two can be ignored because it is redundant information.
These are, together with the vertices set V and edge list E of the Feynman diagram, that encodes the connectivity (i.e. the topology of the diagram), all the pieces of information we need to completely define the scattering process, and they are encoded into the dataset as a list of three elements: $\vec{x_i} = [G_i, \theta_i, p_i]$, where G_i is a labeled graph.
Let $\mathcal{G}$ and $\mathbb{R}$ be the set of graphs and real numbers respectively, the dataset then consists of objects $[G_i, \theta_i, p_i] \in \mathcal{G} \times \mathbb{R} \times \mathbb{R}$ as features, and labels $y_i \in \mathbb{R}$. The objects $G = (V, E)$ are labeled graphs associated with the experiment, together with the functions $n : V \longrightarrow \mathbb{R}^3, n(v) \in \{[1,0,0],[0,1,0],[0,0,1]\}$ represents the data associated with the nodes (initial state, propagator state, or final state) and that is a sort of time order, and $a : E \longrightarrow \mathbb{R}^3$, $a(e) = (m, s, Q)$ represents the data associated with the edges, that encodes particles by their quantum numbers: *mass, spin, and electric charge*. If we take the massless case in (2.48), we clearly see that we loose the dependence on the momentum p, hence when dealing with massless dataset the datapoint becomes $[(G_i, \theta), y_i]$.

In this project we will deal with three different Feynman diagrams, corresponding to Bhabha



(a) (b) (c)

Figure 3.2: Feynman diagrams studied during this project: Bhabha scattering s-channel (3.2a), Bhabha scattering t-channel (3.2b) and $e^+e^- \longrightarrow \mu^+\mu^-$ scattering (3.2c).

scattering t-channel and s-channel in both massive and massless regime and $e^+e^- \longrightarrow \mu^+\mu^-$ in pure QED for low energies.

We note that in the limit of high energies, where any leptonic family can be considered massless, we recover a $U(3)$ accidental leptonic flavour symmetry[1], hence fig.3.2a and fig.3.2c for high energies provides the same results.

3.4 Model Construction

We constructed a fully-quantum parametrized quantum circuit where the output is extrapolated as in (1.47):

$$f(G,\theta,p,\alpha,\omega) = \langle 0|U(G,\theta,p,\alpha)^\dagger W(\omega)^\dagger O(a,b) W(\omega) U(G,\theta,p,\alpha)|0\rangle \tag{3.1}$$

where in the massless case we remind that we loose the dependence on p.
The feature map $U(G,\theta,p,\alpha)$ takes as input a labeled graph $G = (V, E, \mathcal{X}_V, \mathcal{X}_E)$, the kinematic variables of the scattering θ, p and a set of trainable hyper-parameters $\alpha \in \mathbb{R}^3$; hence the feature map is trainable too in order to improve the performance by learning the importance of the particles quantum numbers (encoded as edge features) in the interaction process. In this thesis we are focusing on QED tree level Feynman diagrams in (3.2), this motivated us to set the circuit dimension to $|V(G)| = 6$ and keeping it constant throughout the work.
The massive feature map is defined as:

$$U(G,\theta,p,\alpha) = \prod_{v\in V(G)} R_2^{(v)}(\theta,p) \prod_{k\in V(G)} R_X^{(k)}\left(x_V^k \cdot \begin{bmatrix}\pi/2\\ \pi\\ 3\pi/2\end{bmatrix}\right) \prod_{e_{(i,j)}\in E(G)} R_{ZZ}^{(i,j)}(x_E^e \cdot \alpha) \prod_{a\in V(G)} H^{(a)} \tag{3.2}$$

where we associated each node of the graph to a single qubit; then $x_E^e \in \mathcal{X}_E$, space of edge features, $x_E^e \equiv a(e_{ij}) = (m, S, Q)$ and $x_V^k \in \mathcal{X}_V$, space of node features, with $x_V^k \equiv n(v^k) \in \{[1,0,0],[0,1,0],[0,0,1]\}$. Eq.(3.2) is then an Hamiltonian encoding of the data.

The operators we've used are the following:

- $R_{ZZ}(\phi) = e^{-i\frac{\phi}{2}\sigma_Z\otimes\sigma_Z} = \begin{bmatrix} e^{-i\frac{\phi}{2}} & 0 & 0 & 0\\ 0 & e^{i\frac{\phi}{2}} & 0 & 0\\ 0 & 0 & e^{i\frac{\phi}{2}} & 0\\ 0 & 0 & 0 & e^{-i\frac{\phi}{2}}\end{bmatrix}$ is the Ising ZZ coupling gate.
- $R_X(\phi) = e^{-i\frac{\theta}{2}\sigma_X} = \begin{bmatrix} cos(\theta/2) & -isin(\theta/2)\\ -isin(\theta/2) & cos(\theta/2)\end{bmatrix}$ is the single qubit X rotation.

[1] it softly breaks when we have to take into account the masses of leptons

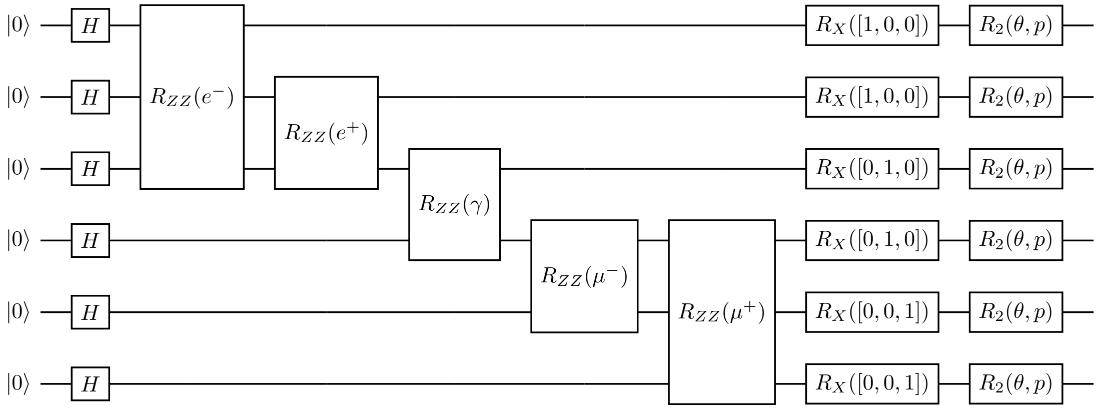


Figure 3.3: feature map that encode the $e^+e^- \longrightarrow \mu^+\mu^-$ in fig.3.2c.

- $R_2(\alpha,\beta) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & e^{-i\beta} \\ e^{i\alpha} & e^{i(\alpha+\beta)} \end{bmatrix}$.
- in the massless case $R_2(\alpha,\beta)$ operator is replaced by $R_1(\delta) = e^{i\delta/2}R_Z(\delta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{bmatrix}$, which is the phase-shift gate that introduces a relative phase between $|0\rangle$ and $|1\rangle$ states.

We then built an ansatz circuit which is pretty similar to the achitecture of the feature map, but it considers more vertices connections of the graph, hence more Ising-ZZ gates between the qubits. Namely we defined the ansatz as:

$$W(\omega) = \prod_{l=1}^{L}\left(\prod_{v\in V(\bar{G})} R_2^{(v)}(\omega_{|E(\bar{G})|+|V(\bar{G})|+v,l}) \prod_{k\in V(\bar{G})} R_X^{(k)}(\omega_{|E(\bar{G})|+k,l}) \prod_{e_{i,j}\in E(\bar{G})} R_{ZZ}^{(i,j)}(\omega_{ij,k})\right) \tag{3.3}$$

where L is the number of layers of the circuit and $\omega \in \mathbb{R}^{(|E(\bar{G})|+3|V(\bar{G})|)\times L}$ is the tensor of trainable parameters. The graph $\bar{G}$ has the same number of nodes as the encoded graph ($|V(G)| = |V(\bar{G})|$) and its topology is the union of all 2-particles $\longrightarrow$ 2-particles scattering at tree level; in particular it connects in any possible ways all the *initial nodes* (which have node feature $[1,0,0]$) to all the *interaction nodes* (which have node feature $[0,1,0]$) and in the same way it contains every possible connection between *interaction nodes* and *final nodes* (with node feature $[0,0,1]$).

We used this connectivity among qubits to make the ansatz more expressive. A standard framework for handling a graph-structured data is using graph convolutional neural network, which is based on node-permutation invariance of the graph. As is discussed in [35, 28], quantum convolutional layers are a special case of equivariant circuit, and they imply the constraint that the trainable parameters of the same layer must be equal. We do not want to put such a constraint in this model since graph nodes encode time-order information of the scattering process, hence permuting two nodes and their corresponding labels will lead to a new Feynman diagram with a different time-order and then a brand new process. In particular a richer explanation can be given in the case both initial nodes (in fig.3.2 nodes number 0 and 1) were exchanged with the final ones (in fig3.2 nodes number 4 and 5); this particular permutation will represent a *time reversal* transformation of the process, since the initial state becomes the final one and vice versa. From quantum field theory fundamental principles it is possible to recover the *CPT theorem*, which states that Nature is invariant under application of *Charge-Parity-Time* transformation,

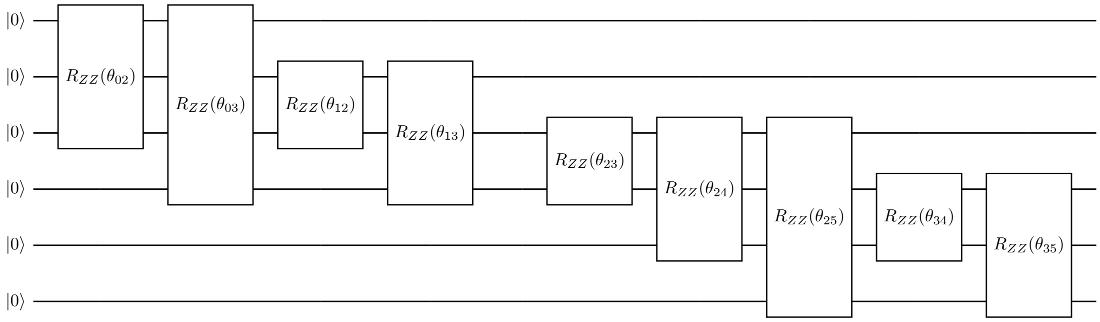


Figure 3.4: Qubit entanglement of the ansatz in eq.(3.3), it includes every possible topology of 2-particles⟶ 2-particles QED tree level Feynman diagrams.

but in general other transformations such as C, P, T, CP can be violated, such as in electroweak theory of the Standard Model. Hence for further developments that include different interactions than QED that violate T transformation we specifically have to build non-convolutional layers; but QED does preserve time reversal transformation, then for purely QED scattering process we are free to constrain the parameters and working on quantum convolutional layers. In this work we've chosen not to put such a constraint, to give to the model enough expressibility.

After the application of the feature map $U(G, \theta, p, \alpha)$ and the ansatz $W(\omega)$ the output is computed from (3.1), which is nothing but the expectation value of an observable O,

$$f(G, \theta, p, \alpha, \omega) = \langle\psi| O |\psi\rangle \tag{3.4}$$

where O is defined as:

$$O = \tilde{O}^{(1)} \otimes \mathbb{1}^{\otimes 5} \tag{3.5}$$

where $\tilde{O}$ is a one-qubit observable applied to the first qubit of the circuit. $\tilde{O}$ was defined as a diagonal and positive hermitian matrix, namely:

$$\tilde{O} = \begin{pmatrix} a^2 & 0 \\ 0 & b^2 \end{pmatrix} \tag{3.6}$$

where $a, b \in \mathbb{R}^+$ and $\tilde{O} \neq \mathbb{1}$ are two additional trainable parameters of the model optimized in the training process. This particularity of our model was inevitable since we weren't able to find a priori a good parametrization $\tilde{O}(\tilde{a}, \tilde{b})$ for the observable.

All the datasets are made of 1000 elements with random combinations of (p, θ) (in massless case we have only the variable θ); the initial datasets are then split into training and test sets with 0.8 and 0.2 proportion. The training process consists in a 5-fold cross-validation with a batched training set with batch-size= 20, with no hyper-parameter optimization, just to evaluate the mean performance of the model. The optimization steps are done after every batch and we use as loss function a cumulative-squared-error function

$$\mathcal{L}(\alpha, \omega) = \sum_{i=1}^{batch-size} \left[y_i - f(G_i, \theta_i, p_i, \alpha, \omega)\right]^2 \tag{3.7}$$

which is optimized by an Adam optimizer with learning rate $l_r = 1 \times 10^{-2}$. Since this is a regression task the only measure for the quality of the training is the behaviour of the loss

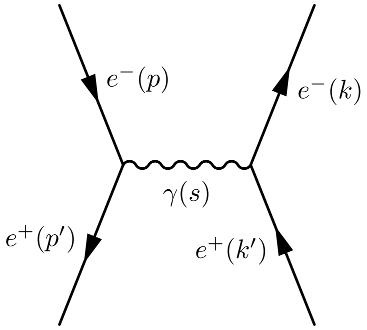


Figure 3.5: Feynman diagram corresponding to Bhabha scattering s-channel.

function with respect to the epochs of the training process.
After the 5-fold cross validation we take the best set of parameters, that corresponds to the one which minimizes the loss for the validation set, and use it to predict the instances of the test set. For every Feynman diagram the depth of the circuit is fixed, with a number of layers of the ansatz $L = 3$.

3.5 Single Diagram

First of all we focused on the simplest possible case, hence we tried to figure out if our QGNN was able to learn a single Feynman diagram and compute its squared matrix element $|\mathcal{M}|^2$; in this project were studied Feynman diagrams in (3.2) both in the massive and massless regime. Since we are focusing of one Feynman diagram at a time the input graph structure will be always the same, which means the QGNN is not properly learning the topology of the graph and distinguish it from different ones.

3.5.1 Bhabha Scattering

We now start focusing on Bhabha scattering process:

$$e^+e^- \longrightarrow e^+e^- \tag{3.8}$$

This is a simple scattering process which can be solved analitically at tree level; this process it is pretty important in our dissertation because it represents an example of scattering process with multiple channels, hence with multiple topologically indipendent Feynman diagrams (at tree level we have two of them), the *s-channel* and *t-channel* (this notation derives from the Mandelstam variables s, t, u in which can be written the transferred momentum of the interaction vertex). This will be used later as a benchmark example for our interference protocol.

S-channel

we recall the scattering amplitude associated to Feynman diagram in fig.3.5:

$$\mathcal{M}_s = i\frac{e^2}{s}\bar{u}(k)\gamma_\mu v(k')\bar{v}(p')\gamma^\mu u(p) \tag{3.9}$$

where we've defined the Mandelstam variable $s \equiv (p_\mu + p'_\mu)^2$.

for which in sec.2.2 we got the final results for the unpolarized scattering amplitude squared $|\bar{\mathcal{M}_s}|^2$ in both massive and massless regime:

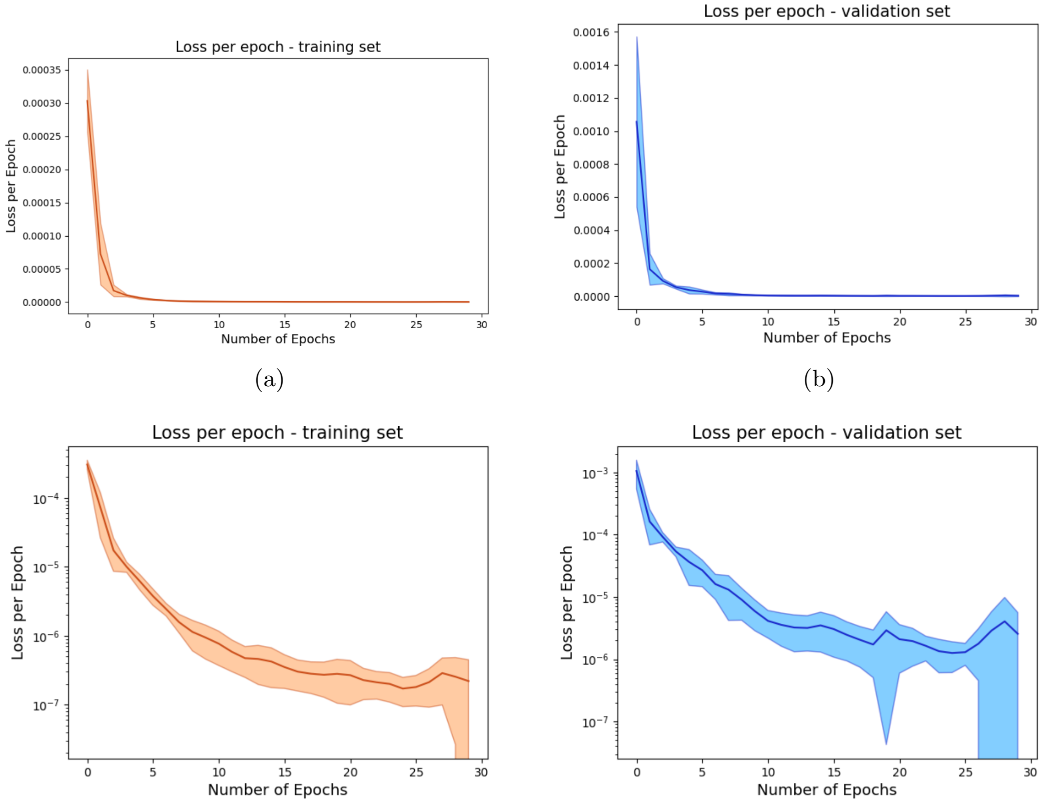
$$|\bar{\mathcal{M}}|^2{}_m = \frac{1}{4}\sum_{initspins}\sum_{finspins}|\mathcal{M}|^2 = e^4\left[1+\frac{p^4cos^2(\theta)}{(p^2+m^2)^2}+\frac{m^2}{2(p^2+m^2)}\right] \tag{3.10}$$

$$|\bar{\mathcal{M}}|^2{}_{m=0} = \frac{1}{4}\sum_{initialspins}\sum_{finalspins}|\mathcal{M}|^2 = e^4[1+cos^2(\theta)] \tag{3.11}$$

where $m \sim 0.511$ MeV is the mass of the electron and $p, \theta \in \mathbb{R}$.

Massless Regime

From (3.11) it is clear that the output is independent from the variable p, hence we build dataset of 1000 with different values of scattering angles $\theta \in [0.5, \pi]$; this choice of the interval was due to the fact that, as it will be shown in the following section, t-channel matrix element squared presents a vertical asymptote at $\theta = 0$, so we start at $\theta = 0.5$ in order to avoid exploding values of the output, which are in general difficult to learn.



(a)

(b)

(c)

(d)

Figure 3.6: Fig.3.6a-3.6c represent the training average loss for massless Bhabha s-channel at different training epochs for a 3-layers QGNN; on the other hand fig.3.6b-3.6d is the validation average loss for massless Bhabha s-channel at different training epochs for a 3-layers QGNN; the error bounds represents the standard deviation computed for the 5 folds of the cross validation.

Fig.3.6a represents the average loss function of the training after the 5-fold cross validation. It is important to note that the loss function assumes such small values because we are working

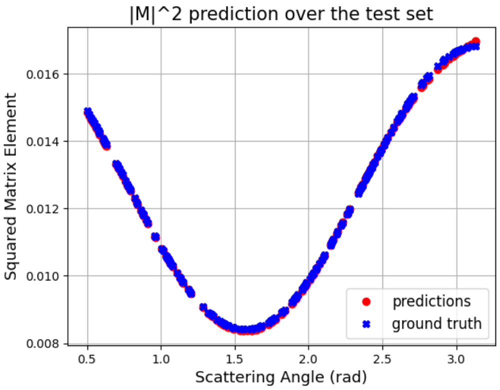


Figure 3.7: Prediction over the test set for Bhabha s-channel Feynman diagram after the 5-fold cross validation; this plot corresponds to the best set of circuit parameters.

on raw data and the theoretical values of $|\bar{\mathcal{M}}|^2$ have order of magnitude 10^{-2}. It is also easy to see that the model converges very quickly to the optimal configuration, since after more or less 5 epochs the loss value is $< 10^{-5}$, and it is also robust since there are only small fluctuations over the mean loss values.
If we take a look at fig.3.6b we see a similar behaviour for the average validation loss, with slightly bigger uncertainty bounds.
The mean squared error values after 30 training epochs are the following:

	average final loss function
training set	$(1.104 \pm 1.140) \times 10^{-8}$
validation set	$(1.597 \pm 1.907) \times 10^{-8}$

Table 3.1: average final value of the loss function per element for training and validation sets.

After 5-fold cross validation the optimal parameters are taken in order to evaluate the performance of the circuit over a fully-unknown dataset;

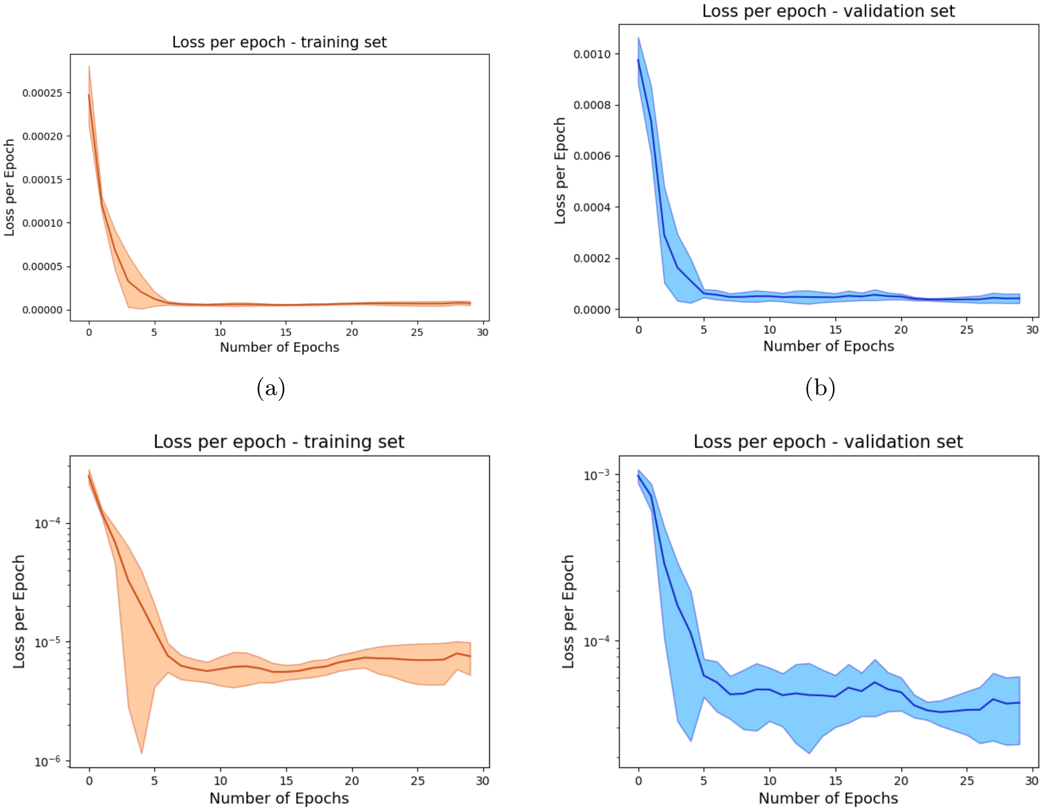
To evaluate the goodness of the model we choose as metric the MSE, also adopted during the optimization process.

$$MSE = \sum_{i=1}^{N} \frac{(y^{(i)}(p,\theta) - \hat{y}^{(i)})^2}{N} \tag{3.12}$$

where $\hat{y}^{(i)}$ is the theoretical value and $y^{(i)}(p,\theta)$ is the prediction of the trained network and N is the number of datapoints of the evaluated dataset. For high energy Bhabha s-channel the mean squared error is equal to 1.838×10^{-9} and the mean relative error was about $\sim 0.30\%$. Such a difference between the MSE from the cross-validation and from the test set is due to the fact that after cross-validation we computed the mean value of the loss, while in test evaluation we take the best model; however, we see that the results are consistent because are within the error bounds computed for the 5-fold cross validation.

Massive Regime

From (3.10) we find that the matrix element squared depends also on a second variable p, which represents the modulus of the momentum of the particles[2]. We do another 5-fold cross validation training process with features (p, θ) generated randomly from a uniform distribution in the interval $p \in [0, 5.0]$ MeV and $\theta \in [0.5, \pi]$. Also in this case we keep the number of ansatz layers fixed at 3.



(a)

(b)

(c)

(d)

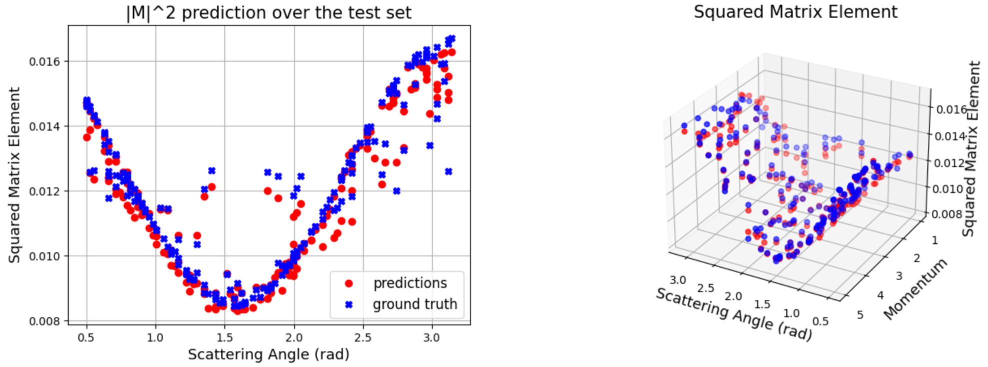
Figure 3.8: Fig.3.8a-3.8c represents the training average loss for Bhabha s-channel in the massive regime for a 3-layers QGNN; on the other hand fig.3.8b-3.8d is the validation average loss for massive Bhabha s-channel after a 5-fold cross validation.

The massive regime represents a more complex problem, so the uncertainty bounds in fig.3.8a and fig.3.8b are expected to be bigger than the ones in the massless case, but after 10 epochs we see that the uncertainty bound becomes very small and all the loss functions converges to the solution. At the end of the training process we find the following values for the training and validation average losses:

[2] remember that we are working in the centre of mass framework, hence we need only one momentum variable.

	average final loss function
training set	$(3.781 \pm 1.154) \times 10^{-7}$
validation set	$(2.646 \pm 1.154) \times 10^{-7}$

Table 3.2: average final value of the loss function per element for training and validation sets.

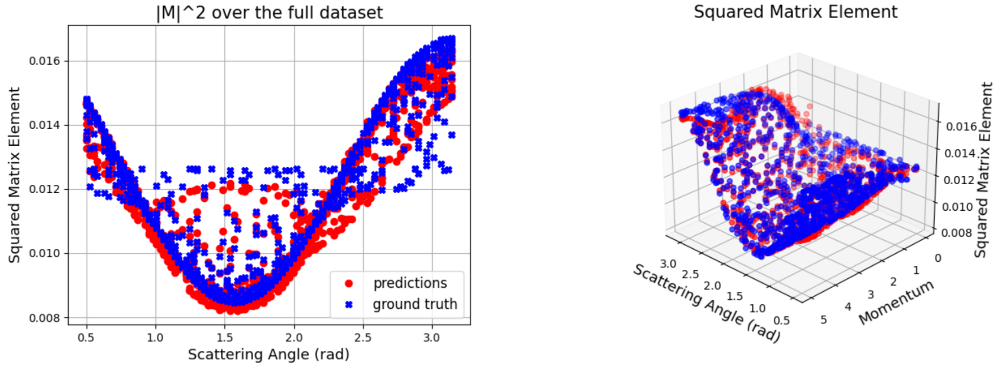


(a)

(b)

Figure 3.9: Test outcomes for the massive Bhabha s-channel.

The final model returns for the test set a mean squared error equal to 1.686×10^{-7} and a mean relative error equal to 2.35%. Because the test set is composed by 200 datapoints we are not able to comprehend well how the circuit has learnt this scattering process, hence we print the predictions of the trained model over the full dataset. From fig.3.10 we note that the circuit struggles to learn low-momentum scattering, where the matrix element becomes flatter and flatter, while converges to the ground truth values for higher momentum, where the massive solution tends to the massless regime.



(a)

(b)

Figure 3.10: Full-dataset outcomes for the massive Bhabha s-channel.

T-channel

In sec.2.2 we wrote the amplitude corresponding to the t-channel Feynman diagram in fig.3.11:

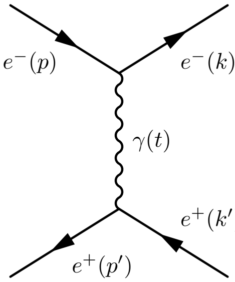


Figure 3.11: Feynman diagram corresponding t-channel, Bhabha scattering.

$$\mathcal{M}_t = -i\frac{e^2}{t}\bar{u}(k)\gamma_\mu u(p)\bar{v}(p')\gamma^\mu v(k') \tag{3.13}$$

where $t \equiv (k-p)^2$ is a Mandelstam variable. The unpolarized scattering amplitudes squared in both massive and massless regime are:

$$|\bar{\mathcal{M}}|^2{}_m = \frac{1}{4}\sum_{initspins}\sum_{finspins}|\mathcal{M}|^2 = \frac{2e^4}{p^4(cos(\theta)-1)^2}\left[(p^2(1+cos(\theta))+2m^2)^2+2(2p^4-m^4)\right] \tag{3.14}$$

$$|\bar{\mathcal{M}}|^2{}_{m=0} = \frac{1}{4}\sum_{initspins}\sum_{finspins}|\mathcal{M}|^2 = \frac{2e^4}{(1-cos(\theta))^2}\left[4+(1+cos(\theta))^2\right] \tag{3.15}$$

T-channel represents a totally different case than s-channel because, as it is possible to see in (3.14) and (3.15), the underlying functions present a vertical asymptote, at $p = 0$ or at $\theta = 0$, so both classical and quantum neural networks will struggle in learning those rapidly varying regions. For that reason we impose an arbitrary cutoff for the scattering angle at $\theta = 0.5$.

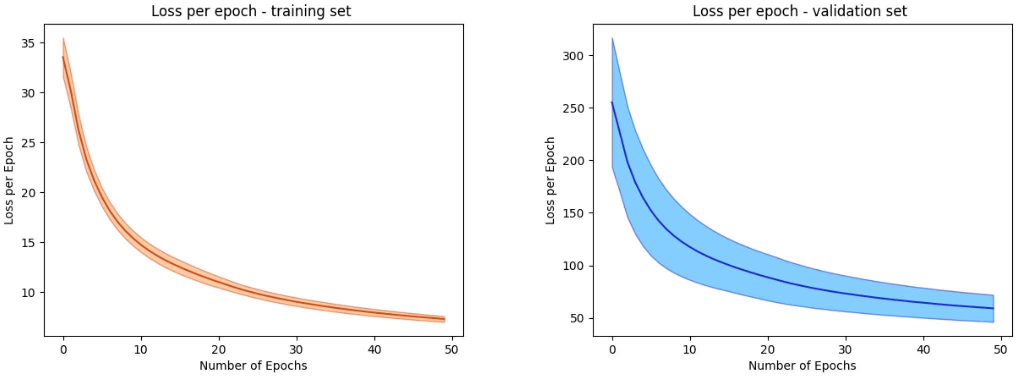
Massless Regime

We repeat the same procedure adopted for s-channel, but this time we perform a 5-fold cross validation training of 50 epochs; The dataset was composed by 1000 elements with different scattering angles, with $\theta \in [0.5, \pi]$. In fig.3.12 are represented the average training and validation loss of the circuit.

While fig.3.12a shows narrow uncertainty bounds and a convergence of the loss function at order < 10, we see that in fig.3.12 the uncertainty bounds are quite wide, which means a dependence of the performance on the initialization of the trainable parameters. After the 5-fold validation simulation we find a mean squared error equal to:

	average final loss function
training set	0.365 ± 0.015
validation set	0.369 ± 0.080

Table 3.3: average final value of the loss function per element for training and validation sets.



(a)

(b)

Figure 3.12: Average performance of our 3-layers deep QGNN for the massless Bhabha t-channel after a 5-fold cross validation: fig.3.12a shows the average training loss over the number of training epochs, while fig.3.12b is the validation average loss.

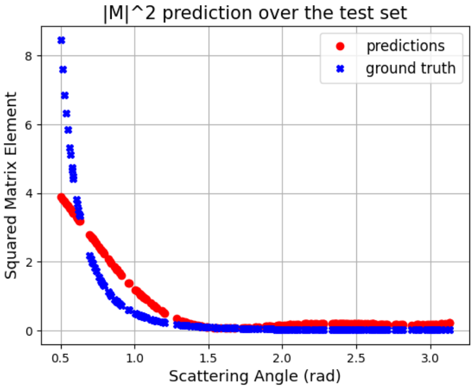


Figure 3.13: Predicted values for the massless Bhabha scattering t-channel of the pre-trained circuit.

As we can see there is a good agreement between theoretical values and predicted ones for scattering angles distant from the critical region, while for $\theta \longrightarrow 0$ the circuit misses the predictions. To evaluate the performance of the model we compute the mean squared error over the test set, which in this case was equal to 0.461, obviously due to the bad predictions of the model for smaller scattering angles θ. For Bhabha t-channel the mean relative error is not a good metric for evaluating the performance of the model; in fact its corresponding amplitude modulus squared lives in a very wide region of values, that goes to ∞ for $\theta \longrightarrow 0$ and that goes to 0 for $\theta \longrightarrow \pi$; this implies that the model will get to a trade-off in predicting all the values, but for small discrepancies of $|\bar{\mathcal{M}}|_t^{\,2}$ we will have an explosion of the relative error, since the true values goes to 0. We will then focus only on the MSE.

In order to build a better model we enlarge the architecture of the QGNN and set the number of layers (repetitions of the trainable ansatz) equal to 5, and repeated a 5-fold cross validation process with 50 training epochs per fold.

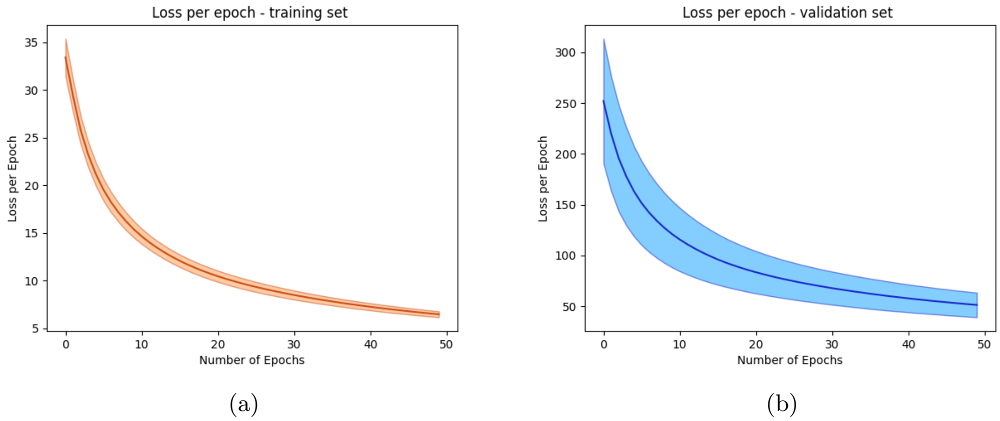


Figure 3.14: Loss evolution during the training of the 5-layers deep QGNN for the massless Bhabha t-channel.

After the training process we find the following average losses per element:

	average final loss function
training set	(0.322 ± 0.015)
validation set	(0.321 ± 0.075)

Table 3.4: average values of the loss function per element for training and validation sets.

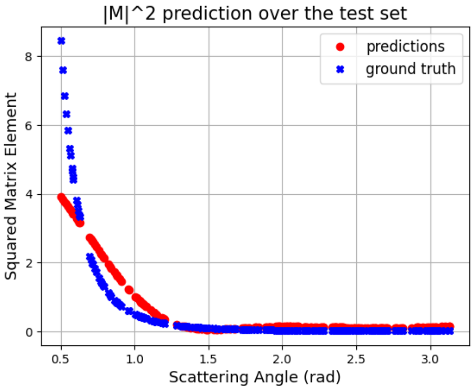
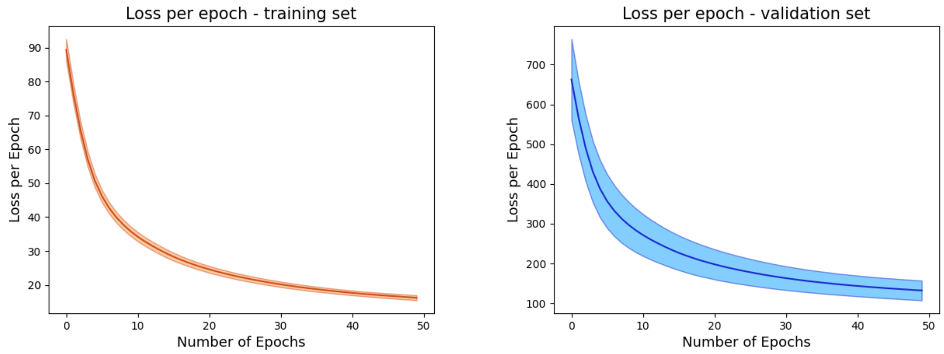


Figure 3.15: Prediction of $|\bar{\mathcal{M}}|^2$ for high energy t-channel computed by a 5 layers QGNN.

In this case we found that the MSE over the test set is around 0.417.
Then the increasing of the circuit depth from 3 layers to 5 does not produce a sensible improvement on the predictions of the circuit, but on the other hand we have a bigger computational cost since we deal with a more complex quantum algorithm.

Same as the high energy case we ask ourselves whether by enlarging the circuit by adding more ansatz gates (3.3) the regression performances improve.

We saw in this section that our model struggles to learn the true values of $|\bar{\mathcal{M}_t}|^2$ for Bhabha scattering t-channel both in high energy and low energy regime, probably due to the presence of a vertical asymptote for $\theta = 0$ (and $p = 0$ for the massive fermion case). We supposed that another problem was that a uniform splitting of the θ range, used to build the synthetic dataset, would under populate data from the asymptotical region, which is highly decreasing in a very small interval of values. We then build a new dataset with a non uniform splitting over the full interval $\theta \in [0.5, \pi]$ and we split them in two subsets $[0.5, 1.0], [1.0, \pi]$ (this can be interpreted as a similar approach of the adaptive quadrature) and take 500 elements for each subset; in this way we ensured that the asymptotical region was well populated. We then train a 3 layers QGNN over this denser value.



(a) (b)

Figure 3.16: Loss evolution for the Bhabha t-channel denser dataset.

	average final loss function
training set	0.811 ± 0.038
validation set	0.826 ± 0.155

Table 3.5: average values of the loss function per element after a 5-fold cross validation over a denser t-channel dataset.

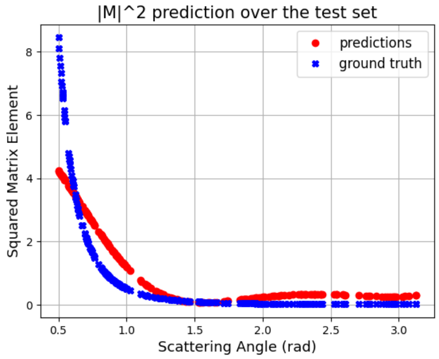
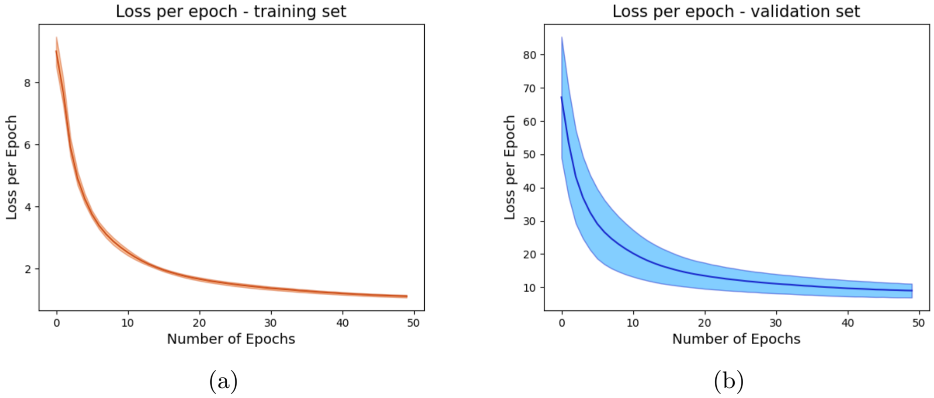


Figure 3.17: prediction for the denser t-channel dataset.

The MSE computed for the test set was around 0.947. From the MSE values it follows that the model looses regression capability and this is due to the fact that the asymptotical region seems not to improve, even with a major population; this means that is the architecture of the network itself that is not so indicated for this kind of functions. We also exploited other heuristic methods for regularize the divergence of $|\bar{\mathcal{M}_t}|^2$, such as inserting a small finite positive quantity in the denominator, or change the scattering angle cutoff. We then changed the asymptote and fix it to $\bar{\theta} = 0.6$.



(a) (b)

Figure 3.18: Loss evolution for Bhabha t-channel with a different angle cutoff.

	average final loss function
training set	$(5.579 \pm 0.202) \times 10^{-2}$
validation set	$(5.635 \pm 1.295) \times 10^{-2}$

Table 3.6: Average values of the loss function per element after a 5-fold cross validation over t-channel dataset with a angle cutoff $\bar{\theta} = 0.6$.

After a 5-fold cross validation process we take the best model and evaluate a test set composed by unseen data, for which we find an average MSE per element equal to 0.0684.

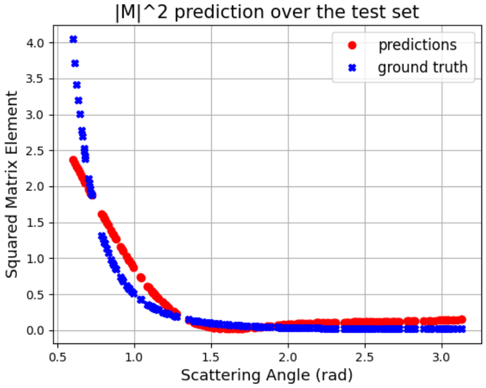


Figure 3.19: Prediction of the QGNN for Bhabha t-channel where we've truncated the divergence by shifting the cutoff for θ.

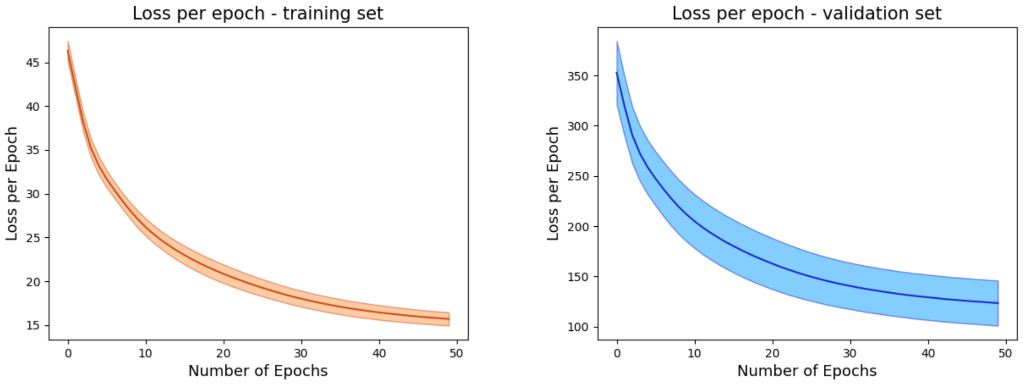
The performance improves of a factor ~ 6.75, but this is expected since we have cut a bigger region for the divergence, so the MSE must decrease. In fact we can still observe that the model suffers, even by shifting the cutoff angle, from a similar problem in learning $|\bar{\mathcal{M}_t}|^2$. So we can conclude that by further changing the cutoff angle $\bar{\theta}$ we can always improve the performance of the model, but on the other hand we loose essential information on the amplitude squared.

Massive Regime

The massive regime for Bhabha scattering t-channel is even more problematic since in this case there are present 2 divergences of $|\bar{\mathcal{M}}|^2$, the first one at $\theta \longrightarrow 0$ and the second one for $p \longrightarrow 0$. Differently from the massive Bhabha scattering s-channel, which is completely defined even at $p = 0$[3], we can easily see that for $p \longrightarrow 0$ (3.14) the leading terms are:

$$|\bar{\mathcal{M}}|^2{}_m \xrightarrow{p \longrightarrow 0} \frac{4e^4}{p^4(cos(\theta) - 1)^2} m^4 \longrightarrow \infty \tag{3.16}$$

Hence we study a dataset composed by 1000 elements with different kinematic parameters (p, θ), where $\theta \in [0.5, \pi]$ and $p \in [1.0, 5.0]$ MeV, in order to avoid the values to explodes to infinity.



(a) (b)

Figure 3.20: Average performance of the 3-layer QGNN for the massive Bhabha t-channel: in fig.3.20a is shown the average training loss of the circuit, while fig.3.20b represents the validation average loss.

After a 5-fold-cross validation process of 50 training epochs each the circuit computed the following values of MSE for both training and validation:

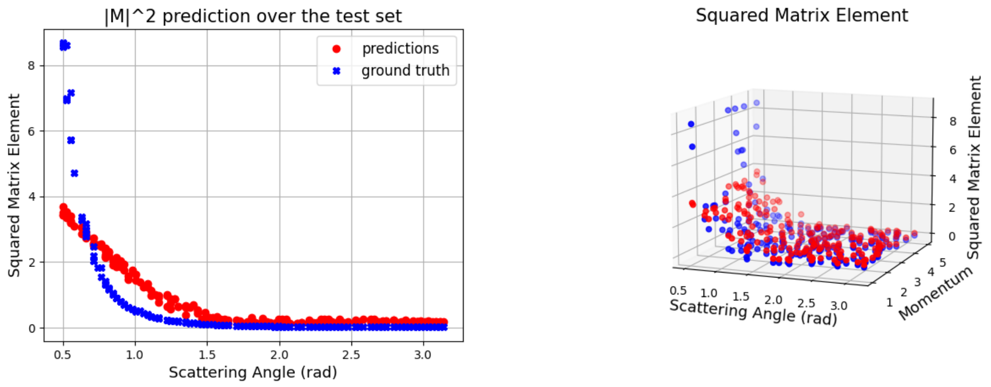
	average final loss function
training set	0.783 ± 0.037
validation set	0.771 ± 0.140

Table 3.7: average MSE computed by a 3-layer QGNN for the training and validation sets.

From fig.3.7 we can state that the performances of the network are twice bigger than the ones analyzed in the high energy limit, and that can be due to the presence of the divergence for $p \longrightarrow 0$.

[3] in fact if we evaluate (3.10) for $p = 0$ we get that:

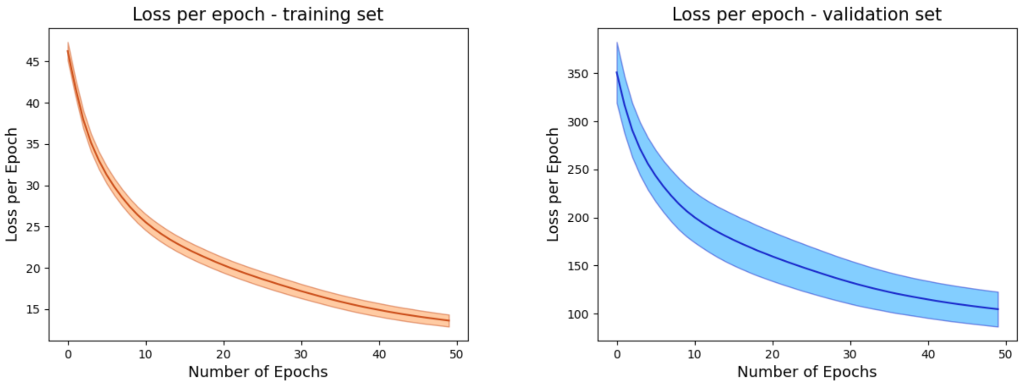
$$|\bar{\mathcal{M}}|^2{}_m(p = 0, \theta) = \frac{3e^4}{2}$$



(a) (b)

Figure 3.21: Predictions of the 3-layer QGNN for unseen data for Bhabha scattering t-channel with $m \neq 0$.

After a 5-fold cross validation we selected the best set of parameters and estimated $|\bar{\mathcal{M}}|^2$ for a test set composed by unseen datapoints (fig.3.21); our QGNN extimates the scattering amplitude for the t-channel in the low energy limit with a MSE about 1.167.
In the same fashion we did for the high energy limit, we add two more ansatz gates and train this 5-layers deep circuit over the same Bhabha t-channel $m_e \neq 0$ dataset.



(a) (b)

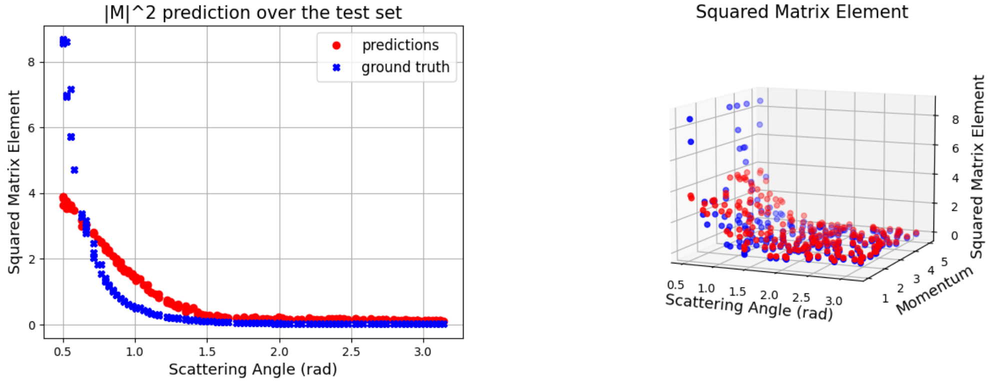
Figure 3.22: Average training and validation loss computed by out 5-layers deep QGNN after a 5-fold cross validation.

	average final loss function
training set	0.681 ± 0.036
validation set	0.654 ± 0.113

Table 3.8: average values of the loss function per element for the Bhabha scattering t-channel in the low energy limit.

The best set of circuit parameters then was evaluated for a test set composed by 200 unseen elements; the circuit computed the predicted values $|\bar{\mathcal{M}}|^2_{exp}$ within an average squared error of

1.016, slightly better than the shorter circuit.



(a)

(b)

Figure 3.23: Plots of the predictions of the pre-trained 5-layers QGNN for Bhabha t-channel in low energy limit.

Like in the high energy limit, as we expected, the performance of the QGNN increased: in fact on average both the training and validation losses get smaller and also the error bands decrease; however also a bigger circuit shows the same difficulties in training the true scattering amplitude due to the presence of two divergences at $\theta = 0$ and $p = 0$.

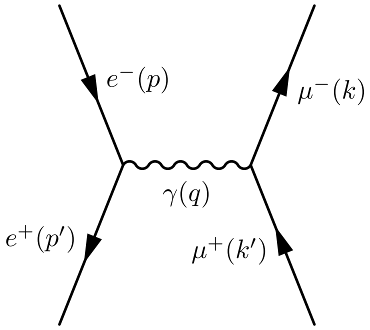


Figure 3.24

Figure 3.25: Feynman diagram related to $e^+e^- \longrightarrow \mu^+\mu^-$ tree level scattering process.

3.5.2 $e^-e^+ \longrightarrow \mu^+\mu^-$ Scattering

We now focus on a different QED scattering process, which at tree level correspond to the Feynman diagram in fig.3.25.
We recall from sec.2.2 the scattering amplitude corresponding to this Feynman diagram:

$$\mathcal{M} = -\bar{u}_\mu(k)(ie\gamma_\mu)v(k')_\mu \frac{-ig^{\mu\nu}}{(p+p')^2}\bar{v}(p')(ie\gamma_\nu)u(p) \tag{3.17}$$

for which we find the following scattering amplitude modulus square, which will be the target function for this dataset

$$|\bar{\mathcal{M}}|^2 = e^4\left[1 + p^2cos^2(\theta)\left(\frac{1}{p^2+m_e^2} - \frac{M_\mu^2}{(p^2+m_e^2)^2}\right) + \frac{m_e^2+M_\mu^2}{4(p^2+m_e^2)}\right] \tag{3.18}$$

For this scattering process we focus only on low energy process, where both leptons families are considered massive ($m_e, m_\mu \neq 0$); in fact in the Standard Model exists an accidental $U(3)$ leptonic flavour symmmetry that is softly broken, which implies that for high energy limits, where it is possible to take all leptons e, μ, τ massless and this symmetry is preserved, the matrix element and the cross section are both invariant under exchange of the leptonic flavours either in initial or final states; then it means that the s-channel of Bhabha scattering and $e^+e^- \longrightarrow \mu^+\mu^-$ scattering in high energy QED give the same result.
In fact it is possible to check that (3.18) for $M_\mu = m_e = 0$ is equal to (3.11)

Proof.

$$|\bar{\mathcal{M}}|^2{}_{e\mu} = e^4\left[1 + p^2cos^2(\theta)\left(\frac{1}{p^2+m_e^2} - \frac{M_\mu^2}{(p^2+m_e^2)^2}\right) + \frac{m_e^2+M_\mu^2}{4(p^2+m_e^2)}\right] \xrightarrow{m_e=M_\mu=0} e^4\left[1+cos^2(\theta)\right] \tag{3.19}$$

which is equal to $|\bar{\mathcal{M}_s}|^2$ in the massless case. □

However, in the low energy regime such a $U(3)$ accidental symmetry breaks and the two matrix elements diverges because electron and muon are no more indistinguishable particles since their differs from their mass values, formally their quantum numbers are now:

$$e^\pm = \left(m_e, \frac{1}{2}, \pm 1\right) \quad \mu^\pm = \left(M_\mu, \frac{1}{2}, \pm 1\right) \quad with \quad M_\mu \neq m_e \tag{3.20}$$

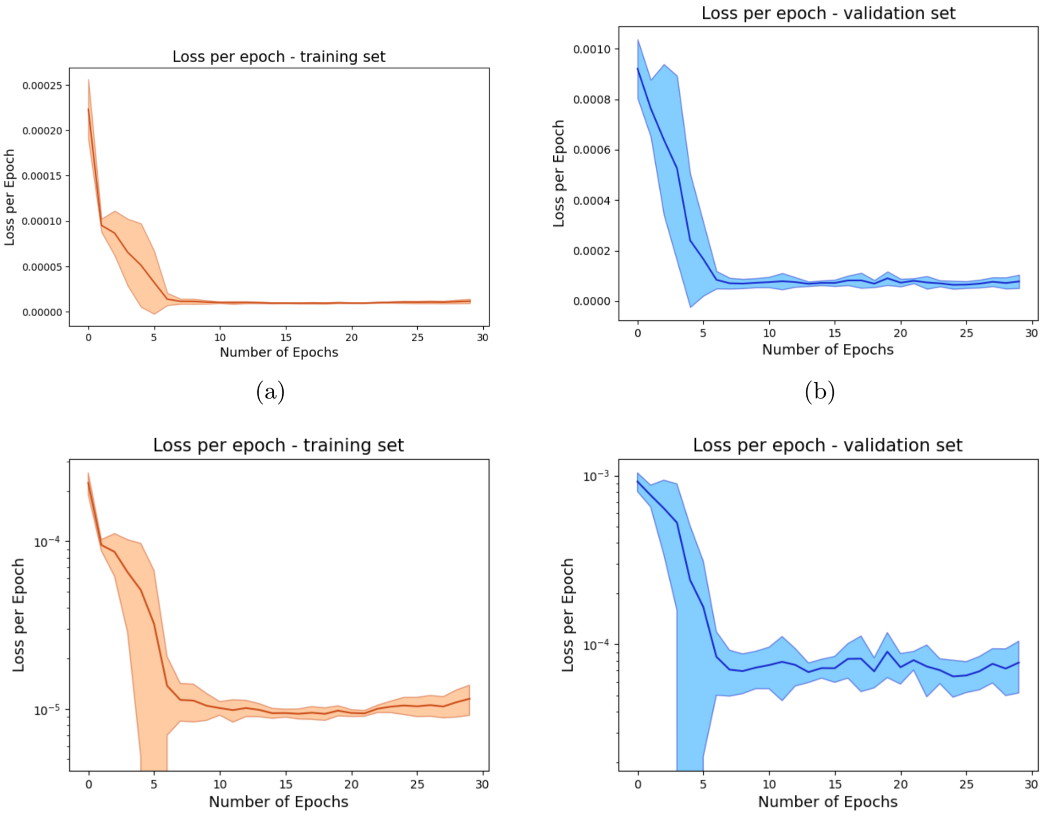
It is also important to note that the two processes happens at two different *threshold energy*, which we recall is the minimum kinetic energy that must be imparted to the initial particles in order for their collision to produce the desired result, which in our case is simply the rest energy of the final particles; for the $e^+e^- \longrightarrow \mu^+\mu^-$ process in fact we need the initial state to have enough energy to produce at rest a couple $\mu^+\mu^-$, which means that:

$$2(m_e^2 + p^2) \geq 2M_\mu^2 \Longrightarrow p \geq \sqrt{M_\mu^2 - m_e^2} \tag{3.21}$$

where m_e and M_μ are respectively the masses of the electon and the muon and $p \in \mathbb{R}^+$ is the modulus of the spatial momentum of the initial particles[4]; then the energy threshold is $\tilde{E}_{tot} = 2M_\mu \sim 211,32$ MeV, which corresponds to a lower bound of the momentum $\tilde{p} = \tilde{E}_{tot}/2$; so the dataset was generated in a range of momentum $p \in [120, 250]$ MeV (in fact for higher momentum both leptons can be considered ad massless) with scattering angles $\theta \in [0, \pi]$. Relation (3.21) is a general constraint that any scattering process must satisfy, and is easy to check that this is always true for Bhabha scattering s-channel, that is why such a channel can always happen, even with initial particles at rest ($p = 0$).

As it is possible to see in fig.3.26, the variance in the training average loss quite large for the first 10 training epochs, but after that the circuit converges to the optimal solution very quickly, and the validation average loss function assumes a very similar behaviour. Such a behaviour denotes a robustness of the circuit, because from different initializations of the circuit parameters we always fall into an good parameter configuration that can efficiently estimate the target outputs for both training and validation sets.

[4] we remember that we work in the centre of mass framework with initial four-momenta: $p^\nu_{e-} = (p, 0, 0, p)$ and $p^\nu_{e+} = (p, 0, 0, -p)$.



(a)

(b)

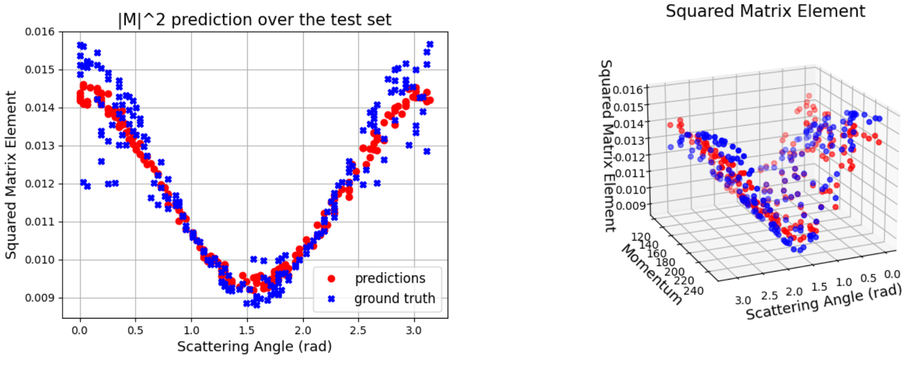
(c)

(d)

Figure 3.26: Average performance of our 3-layers deep QGNN for $e^+e^- \longrightarrow \mu^+\mu^-$ Feynman diagram; fig.3.26a represents the average loss evolution for the training steps, whereas fig.3.26b represents the average loss trend for each epoch for the validation set.

	average final loss function
training set	$(5.745 \pm 1.173) \times 10^{-7}$
validation set	$(4.871 \pm 1.654) \times 10^{-7}$

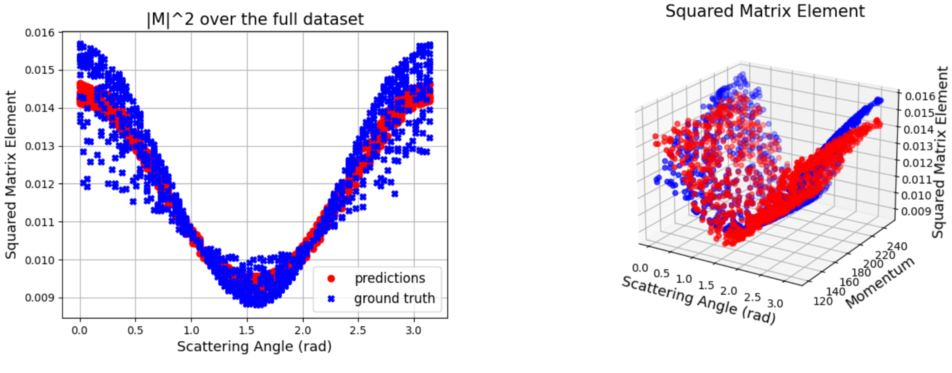
Table 3.9: average values of the loss function per element for the $e^+e^- \longrightarrow \mu^+\mu^-$ scattering (with massive fermions).



(a) (b)

Figure 3.27: Predictions over the test set of the trained QGNN for $e^+e^- \longrightarrow \mu^+\mu^-$ scattering.

The performance of the circuit over the test set is not as good as the one found for $e^+e^- \longrightarrow e^+e^-$ s-channel: in fact the network seems to properly predict the solutions for higher values of p, where $|\bar{\mathcal{M}}|^2$ tends to the massless regime, and it completely fails to compute lower momentum scattering processes, where the cosine wave function becomes flatter. The only differences in the two datasets are the masses of the two particles and the range of momentum we define study the scattering process (remember that for Bhabha s-channel we do not have energy bounds, whereas in $e^+e^- \longrightarrow \mu^+\mu^-$ we do), so it came up the idea that the circuit looses performances for higher values of momentum and masses, and it could be the case for real noisy hardware; for quantum noiseless simulator on the other hand we do not have this issue, then the loss of performance can be due to the fact that we are encoding p via a rotation of qubits, which are by definition periodic, but the target function is not periodic in p; since we are working in a quite large set of values ($p \in [120, 250]$ MeV) it will happen that for different p we will have the same rotation that encodes momentum, but different output function, so that the model sets in a trade-off solution, causing a loss of performance. This process occur also for Bhabha s-channel low energy dataset, but since we have $p \in [0, 5]$ MeV (for higher values we get to the high energy limit) the interval is much smaller, so we will have less ”antagonist” datapoints.



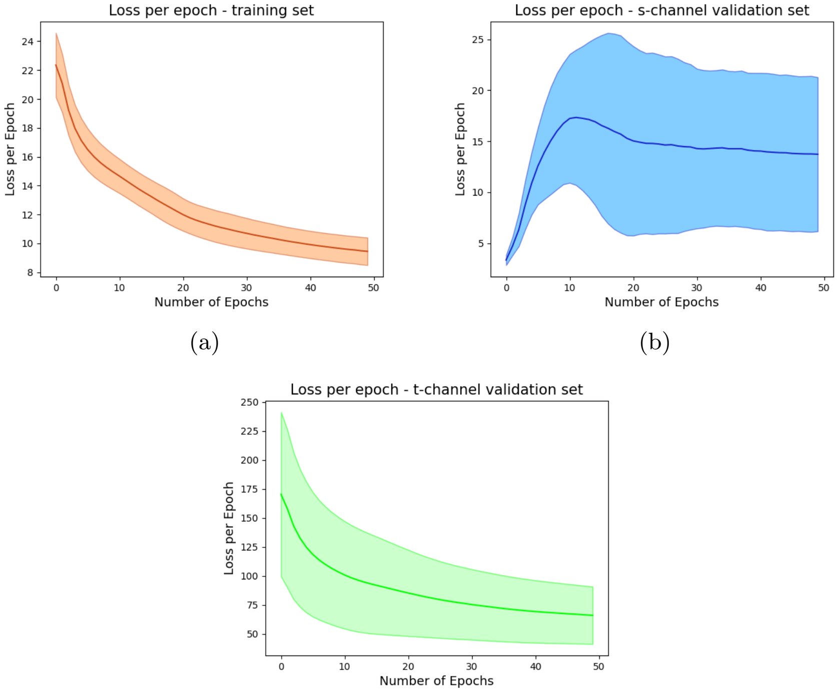
(a) (b)

Figure 3.28: $|\bar{\mathcal{M}}|^2$ values computed for all the elements of the dataset.

A better look on the behaviour of the QGNN is given by fig.3.27b, which shows in a 3-D image how $|\bar{\mathcal{M}}|^2$ becomes flatter for lower values of p.

At the end of the test set evaluation we find that the mean squared error for the test set is about 4.331×10^{-7}.
Fig.3.27 does not show how is the behaviour of the network for different momenta, and this is due to the fact that $|\bar{\mathcal{M}}|^2$ converges rapidly to the high energy limit; hence we plot the computed matrix element for all the 1000 points of the dataset.

3.6 Diagrams Combination



(a)

(b)

(c)

Figure 3.29: Fig.3.29a represents the evolution of the loss over the training set after a 5-fold cross validation; fig.3.29b-3.29c are the behaviours of the loss function evaluated for the validation sets: respectively the first figure is related to the points corresponding to Bhabha s-channel, whereas the latter one is computed for Bhabha t-channel datapoints.

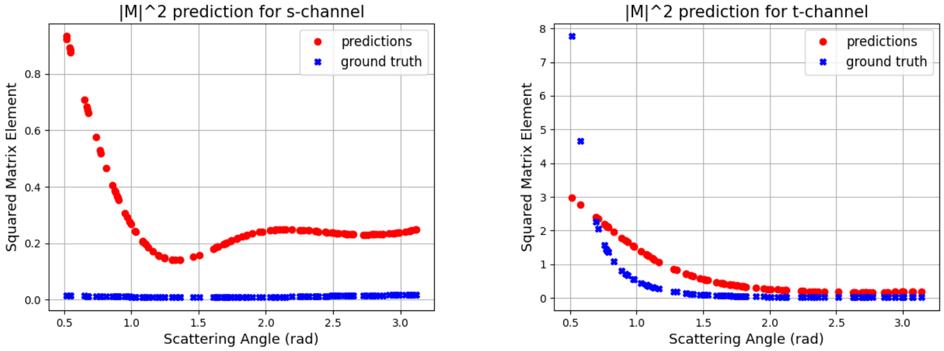
The next step of this project consisted in building a quantum circuit for a dataset of multiple Feynman diagrams, which was able to efficiently distinguish different classes of data (classification task) and compute the corresponding matrix element squared; for this porpouse we focus only on a dataset composed by both high energy Bhabha s-channel and t-channel. For this QGNN the feature map is kept the same as (3.2) and the only remarkable change we make is the ansatz architecture: we choose the fully-connected circuit in fig.1.9, which guarantees a more expressive ansatz, which outperformes the circuit in fig.3.4 because it allows all-to-all qubit connections and hence more graph topologies. In fact as we look at the data structures for both Bhabha s-channel and t-channel datasets we see that the only difference in the features is the graph topology, whereas all the node and edge features are the same; this is another reason why in sec.3.4 we did not define a node-permutation equivariant ansatz as in [35]: if we build such an equivariant ansatz the two different graph topologies will be seen the same for the QGNN and it will not be able to correctly distinguish them.

Since the only difference between the two Feynman diagrams is the graph topologies, with this combined dataset we are checking the pure graph classification capability of the circuit we've designed. For the three sets we find the following average values of the loss function after the 5-fold cross validation process:

	average loss function after 50 epochs
training set	0.472 ± 0.047
s-channel validation set	0.121 ± 0.067
t-channel validation set	0.758 ± 0.284

Table 3.10: Loss values for training set, s-channel datapoints and t-channel ones in the validation set.

We then take the best model and evaluate an completely unknown set of mixed datapoints and computed their associated scattering amplitude $|\bar{\mathcal{M}}|^2$.



(a)

(b)

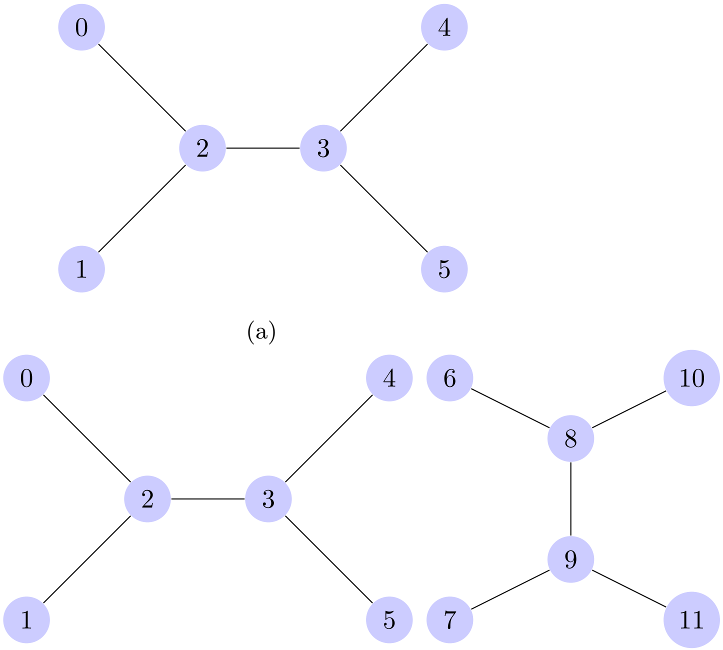
Figure 3.30: Predictions computed by our QGNN with the quantum graph transformer took from [29] for a test set composed by both Bhabha s-channel and t-channel Feynman diagrams.

As we expected the average losses for the two validation sets are worse than in the previous model, but for the s-channel datapoints the differense in huge as they differ by a factor 10^{-7}, so we suppose that also during the test set evaluation we will see very bad performances for computing $|\bar{\mathcal{M}}|^2$ for the s-channel datapoints; on the other hand the validation loss related to t-channel data is more or less twice the one computed in the previous section. The bad results shown in fig.3.30 prove that the network is not able to distinguish two different graph topologies, even with the fully-connected quantum graph transformer defined in [29].
The goodness of the model is evaluated by the computation of the mean squared error for t-channel and s-channel datapoints separately: in the Bhabha t-channel scattering we find a MSE equal to 0.527, whereas for the s-channel test set we find a MSE value equal to 0.104. As we said previously the circuit is not able to distinguish the two graph topologies and then correctly predict the corresponding scattering amplitude squared; especially the model completely fails for the s-channel, as we can see from fig.3.30a; moreover, except for the asympote region for t-channel scattering amplitude, $|\bar{\mathcal{M}_s}|^2$ and $|\bar{\mathcal{M}_t}|^2$ lives in a similar range of values of $\sim 10^{-2}$: in fact usually a networks struggles to span over a too wide range of output values.

It is worthy to discuss a bit on this part of the project; in general graph classification for highly similar graph-structured data is a complex task, both for classical and quantum neural networks, then probably the complexity of the model is not enough to deal with such a delicate task, so it might be useful to look for different strategies as in [55]; but, as we said in chapter 1, QML and especially QGML are very active research fields and not fully comprehended yet. However [8] seems to show very good results for an analogous classical network, which solve both classification and regression tasks for Bhabha scattering and $e^+e^- \longrightarrow \mu^+\mu^-$ scattering, but if we look at its data-structure closer we see that it adopts a big shortcut; in fact [8] consider both Bhabha channels as part of the same data, hence the graph structures of the two classes of graphs becomes very different in term of nodes connections and number of edges.

	Our data structure
s-channel	[(0,2),(1,2),(2,3),(3,4),(3,5)]
t-channel	[(0,2),(1,3),(2,3),(2,4),(3,5)]
	[8] data structure
$e^+e^- \to \mu^+\mu^-$	[(0,2),(1,2),(2,3),(3,4),(3,5)]
Bhabha scattering	[(0,2),(1,2),(2,3),(3,4),(3,5),(6,8),(7,9),(8,9),(8,10),(9,11)]

Table 3.11: Encoding strategies of the Feynman diagrams for the composed dataset.



(a)

(b)

Figure 3.31: Graphs structures analyzed in [8]: fig.3.31a corresponds to the edge list of $e^+e^- \to \mu^+\mu^-$ scattering, whereas fig.3.31b correspond to the adge list of the Bhabha scattering (they glue together the two possible channels).

Then, whether in our project the processed graph topologies are the ones corresponding to the single Feynman diagrams in 3.2, which edge lists are very similar and difficult to distinguish, in [8] the graph structures are totally different and easy to distinguish (in fact it is enough just to count the number of edges for each class), and so they are not testing the true classification capability of their graph neural network.

Moreover [8] omits all the information about single channels matrix elements, so that the GNN simply learns the total scattering process and does not have a full knowledge of the process; we do not want to loose these pieces of information, but we want our quantum model to apprehend the single Feynman diagrams (hence the Feynman rules for a given interaction, in this case QED) and exploit quantum properties in order to compute the interference term between two Feynman diagrams and hence recover the total the matrix element for simple scattering processes with two scattering channels (e.g. Bhabha scattering).

3.7 Interference Protocol

An interesting problem that emerges in quantum field theory is the computation of the scattering amplitudes when a scattering process concerns multiple scattering channels (i.e. multiple topologically independent Feynman diagrams), computation that in the most realistic cases (hadronic scattering or higher order perturbative corrections) can not be computed analytically and are classically expensive (review section 3.1); Formally we can write the total scattering amplitude modulus square $|\mathcal{M}_{tot}|^2$ of a scattering process composed by n topologically independent Feynman diagrams as:

$$|\mathcal{M}_{tot}|^2 = \sum_{i=1}^{n} |\mathcal{M}_i|^2 + \sum_{i=1}^{n} \sum_{j \neq i} \mathcal{M}_i \mathcal{M}_j^* = \sum_{i=1}^{n} |\mathcal{M}_i|^2 + \sum_{i=1}^{n} \sum_{j > i} 2Re(\mathcal{M}_i \mathcal{M}_j^*) \tag{3.22}$$

where the last term corresponds to the interference between i-th and j-th Feynman diagrams. Interference between different probability amplitudes in 3.22 is a fundamental concept in quantum mechanics, that enables different terms in the perturbation theory (then different Feynman diagrams) to interfere with each other in a way that affects the overall probability of a particular process. Interference terms are essential for obtaining consistent and accurate predictions in accordance with the experimental results.

We try to exploit quantum properties of our model in order to find a natural protocol that is able to compute the interference term between two Feynman diagrams. However we concluded that the QGNN model defined in sec.3.5 is not suitable to compute the interference term since its difficulty in extracting information on the scattering amplitude $\mathcal{M}$, which is essential in order to compute the interference term, as it is shown in (3.22).

In sec.3.5 we've defined a quantum graph neural network which was able to compute the matrix element of a single Feynman diagram; hence we have the quantum circuit $C(x, \theta)$ defined in sec.3.4 for which holds:

$$C(G, \theta, p, \omega) |0\rangle^{\otimes 6} \equiv |\psi\rangle \quad s.t. \quad \langle\psi| O(a, b) |\psi\rangle = |\mathcal{M}(G, \theta, p)|^2 \tag{3.23}$$

where (G, θ, p) is an element of the dataset and θ is the set of pre-trained parameters of the circuit C composed by the feature map (3.2) and the ansatz (3.3); moreover O is the trainable observable described in (3.5) and (3.6).

The state $|\psi\rangle$ after the application of the $QGNN$ can be factorized in the following way:

$$|\psi\rangle = \alpha |0\rangle_1 \otimes |\varphi\rangle + \beta |1\rangle_1 \otimes |\chi\rangle \tag{3.24}$$

where $|0\rangle_1, |1\rangle_1$ are 1-qubit states related to the first qubit of the QGNN, and $|\varphi\rangle, |\chi\rangle$ are 5-qubits normal states; then the expression of the amplitudes modulus squared computed by the QGNN becomes:

$$|\mathcal{M}|^2 = \langle O(a, b)\rangle = |\alpha|^2 a^2 \langle\varphi|\varphi\rangle + |\beta|^2 b^2 \langle\chi|\chi\rangle = |\alpha|^2 a^2 + |\beta|^2 b^2 \tag{3.25}$$

where we recall that $\alpha, \beta \in \mathbb{C}$ and $a, b \in \mathbb{R}^+$; since the QGNN depends on the input features, we expect the state coefficients $|\alpha|^2, |\beta|^2$ to be different for different data of the dataset. Scattering amplitude are defined as complex numbers, $\mathcal{M} \in \mathbb{C}$, so we can rewrite it as: $\mathcal{M} = Re(\mathcal{M}) + iIm(\mathcal{M})$, where $Re(\mathcal{M}), Im(\mathcal{M}) \in \mathbb{R}$. It is well known that in the complex vector space $\mathbb{C}$, (3.25) describes a circumference corresponding to every complex number with the same modulus $|\mathcal{M}|$, that means that a priori we are not able to uniquely define $Re(\mathcal{M})$ and $Im(\mathcal{M})$. However, since we are dealing with a mathematical locus which corresponds to a circumference, it is possible to write $Re(\mathcal{M})$ and $Im(\mathcal{M})$ in polar coordinate:

$$\begin{cases} Re(\mathcal{M}) = \sqrt{|\alpha|^2 a^2 + |\beta|^2 b^2} cos(\varphi) \\ Im(\mathcal{M}) = \sqrt{|\alpha|^2 a^2 + |\beta|^2 b^2} sin(\varphi) \end{cases} \tag{3.26}$$

where φ is an unknown angle that we have to determine in order to properly compute $\mathcal{M}$.

It is important to highlight once more that in our dissertation for any Feynman diagram we have different quantum circuits with different parameters. Then in the case of two topologically independent Feynman diagrams G_1 and G_2 (e.g. Bhabha scattering) we have two different QGNNs $C(G_1, \theta, p, \omega_1) \equiv C_1(\theta, p, \omega_1)$ and $C(G_2, \theta, p, \omega_2) \equiv C_2(\theta, p, \omega_2)$ such that:

$$\begin{aligned} C_1(\theta, p, \omega_1) |0\rangle^{\otimes 6} \equiv |\psi_1\rangle \quad , \quad \langle\psi_1| O_1(a, b) |\psi_1\rangle = |\mathcal{M}_1(\theta, p)|^2 \\ C_2(\theta, p, \omega_2) |0\rangle^{\otimes 6} \equiv |\psi_2\rangle \quad , \quad \langle\psi_2| O_2(c, d) |\psi_2\rangle = |\mathcal{M}_2(\theta, p)|^2 \end{aligned} \tag{3.27}$$

where we recall that $O_1(a, b)$ and $2^6 \times 2^6$ matrix defined as:

$$\begin{cases} O_1(a, b) = \tilde{O}_1^{(1)} \otimes \mathbb{1}^{\otimes 5}, \quad where \quad \tilde{O}_1(a, b) = \begin{pmatrix} a^2 & 0 \\ 0 & b^2 \end{pmatrix} \\ O_2(c, d) = \tilde{O}_2^{(1)} \otimes \mathbb{1}^{\otimes 5}, \quad where \quad \tilde{O}_2(c, d) = \begin{pmatrix} c^2 & 0 \\ 0 & d^2 \end{pmatrix} \end{cases} \tag{3.28}$$

In this formalism we can hence define both $\mathcal{M}_1$ and $\mathcal{M}_2$ as:

$$\begin{aligned} &\begin{cases} Re(\mathcal{M}_1) = \sqrt{|\alpha|^2 a^2 + |\beta|^2 b^2} cos(\varphi) = \sqrt{\langle O_1(a, b)\rangle} cos(\varphi) \\ Im(\mathcal{M}_1) = \sqrt{|\alpha|^2 a^2 + |\beta|^2 b^2} sin(\varphi) = \sqrt{\langle O_1(a, b)\rangle} sin(\varphi) \end{cases} \\ &\begin{cases} Re(\mathcal{M}_2) = \sqrt{|\gamma|^2 c^2 + |\delta|^2 d^2} cos(\eta) = \sqrt{\langle O_2(c, d)\rangle} cos(\eta) \\ Im(\mathcal{M}_2) = \sqrt{|\gamma|^2 c^2 + |\delta|^2 d^2} sin(\eta) = \sqrt{\langle O_2(c, d)\rangle} sin(\eta) \end{cases} \end{aligned} \tag{3.29}$$

We know that the interference term between two Feynman diagrams is given by $\mathcal{M}_1\mathcal{M}_2^* + \mathcal{M}_1^*\mathcal{M}_1 = 2Re(\mathcal{M}_1\mathcal{M}_2^*) = 2(Re(\mathcal{M}_1)Re(\mathcal{M}_2) + Im(\mathcal{M}_1)Im(\mathcal{M}_2))$; by inserting the identities in (3.29) we finally get the value for the interference term:

$$2Re(\mathcal{M}_1\mathcal{M}_2) = 2\sqrt{\langle O_1\rangle\langle O_2\rangle}(cos(\varphi)sin(\eta) - sin(\varphi)sin(\eta)) = 2\sqrt{\langle O_1\rangle\langle O_2\rangle} cos(\varphi - \eta) \tag{3.30}$$

As we previously said the two angles φ, η are unknown a priori. The way we extract $|\bar{\mathcal{M}}|^2$ from the QGNN (3.23) raises a phase ambiguity in the computation of $\mathcal{M}$ that we are not able to solve; in conclusion, we can state that the framework that we implemented in sec.3.5 works for computing simple scattering processes at tree level, but is not possible to build an efficient algorithm that can compute the interference term between two Feynman diagrams. In order to develop a protocol that computes without additional d.o.f the interference term we need a different QML approach, that is able to encode without ambiguity the scattering amplitude $\mathcal{M}$ of a process.

Chapter 4

Conclusions

Nowadays Quantum Computing is one of the most active and prolific research fields and it is experiencing a rapid expansion, both hardware development and in the study of better performing quantum algorithms designed for both NISQ devices (run either on actual quantum devices or on noisy simulators) and future noiseless ones (run in noiseless simulators). Moreover, Quantum Computing offers a very wide range of applications in several areas such as scientific ones or also financial or industrial ones, and it is believed that it will outperform the classical counterpart in many taks, in term of time-saving and performance. Quantum Computing is believed to be a very useful tool for the study of quantum systems such as lattices with interacting Hamiltonians, molecules and fundamental interactions.

In this thesis we focused on Quantum Machine Learning, which is the exploitation of quantum devices for machine learning purposes; machine learning became a very useful tool for physics applications and its usage has rapidly increased due to its efficiency in term of trade-off between computational cost and precision, and has become a solid alternative to deterministic but very expensive numerical methods. Many branches of physics benefited of both Quantum Computing and machine learning such as particle physics, astrophysics or physics of matter, but we are still in an early stage. This thesis aims to be an application of QML paradigms for the study of QFT and fundamental particle physics.

During this project we have defined a Quantum Graph Neural Network that was able to learn simple Feynman diagrams in order to compute, given a specific diagram topology, the corresponding matrix element squared $|\bar{\mathcal{M}}|^2$.

We tested our QGNN with tree-level pure QED Feynman diagrams in both high-energy (which implies massless fermions) and low-energy (that means take all fermions massive) regimes; in particular we built synthetic datasets corresponding to $e^+e^- \longrightarrow \mu^+\mu^-$ scattering and Bhabha scattering s-channel and t-channel.

In the first part of chapter 3 we tested the networks on different datasets composed by the same Feynman diagram with different values of momentum p of the initial particles and different scattering angles θ. We proved that our quantum circuit learns well sinusoidal functions such as Bhabha scattering s-channel or $e^+e^- \longrightarrow \mu^+\mu^-$ scattering with both massive and massless fermions and that the model is very robust: for Bhabha s-channel we found a MSE around 1.838×10^{-8} and 1.686×10^{-7} over the test set for respectively massless and massive cases, whereas for μ-pair production scattering we find an average relative error per element of for 4.331×10^{-7}.

On the other hand the quantum model struggles to learn the matrix element for the Bhabha t-channel, due both to the presence of an asymptote at $\theta = 0$ (and at $p = 0$ for the massive case) and to a strong sensibility to the initialization of the circuit parameters; after a 5-fold cross validation, the mean squared error computed by a 3 layers deep QGNN for the Bhabha t-channel was equal to 0.461 for massless electrons and 1.167 for massive ones. We also tried to increase the complexity of the model, hence we trained a 5 layers deep QGNN for both datasets, but there was not a relevant improvement in the performance of the newtork: a MSE equal to 0.417 for the high energy case and equal to 1.016 . In order to test the divergence at $\theta = 0$ we then tried to add more datapoints in the critical region, but the results did not improve significantly, probably due to a limitation of the quantum circuit for these kind of functions. We also tried to adopt different heuristic strategies to mitigate the divergence of the amplitude squared, but not significant improvements were noticed. In general we note that the circuit is less performing when considering massive fermions than for massless ones.

After evaluating the circuit for a single diagram we focused on a more complex problem and we tested whether our QGNN was able to perform both classification and regression tasks on a dataset composed by both Bhabha scattering channels: it turned out that such an architecture is not able to distinguish between two graph topologies and compute the corresponding matrix element squared, with an average relative error per element equal to 0.527 for t-channel data and equal to 0.104 for s-channel, in strong contradiction with the good-performing single-diagram QGNN. We stressed out the general problem on graph representation learning in Quantum Machine Learning, and we noted that for our use-case the problem was even more complex since the two diagrams topologies were very similar.

In the last part of this work was introduced the problem of computing interference terms between two Feynman diagrams by harnessing our QGNN. We proved that from our QGNN arises a phase ambiguity that does not allow us to uniquely identify the scattering amplitude $\mathcal{M}$, that is essential for computing the interference term. From this it follows that it is impossible to exploit the QGNN architecture to compute the interference term, and that we need to implement brand new QML frameworks. Lot of efforts are put into the development of new QML systems that can lead to our desired goal, the definition of a fully-quantum circuit that can extrapolate the interference term between two Feynman diagrams without any need to introduce degrees of freedom, by simply exploiting quantum properties such as entanglement and superposition between quantum states. This could be an outstanding result with many applications in real-world problems.

We are also confident that this preliminary project will be the starting point for new QML applications in the study of fundamental interactions.

Appendix A

Proofs and Examples

A.1 Approximating Quantum Circuits [7]

Proof. Suppose a quantum system starts in the state $|\psi\rangle$, and we perform either the unitary operation U, or the unitary operation V, and then a measurement is performed. Let M be the observable related to the measurement, and let P_U and P_V be the probabilities of obtaining the corresponding measurement outcomes for respectively gates U and V. Then

$$|P_U - P_V| = |\langle\psi|U^\dagger MU|\psi\rangle - \langle\psi|V^\dagger MV|\psi\rangle| \tag{A.1}$$

Let call $|\Delta\rangle \equiv (U - V)|\psi\rangle$, then for Cauchy-Schwarz inequality we can write

$$\begin{gathered}|P_U - P_V| = |\langle\psi|U^\dagger M|\Delta\rangle + \langle\Delta|MV|\psi\rangle| \leq \\ \leq |\langle\psi|U^\dagger M|\Delta\rangle| + |\langle\Delta|MV|\psi\rangle| \leq \||\Delta\rangle\| + \||\Delta\rangle\| \leq 2E(U,V)\end{gathered} \tag{A.2}$$

Suppose we perform a sequence $V_1, V_2, ..., V_m$ of gates intended to approximate some other sequence of gates, $U_1, U_2, ..., U_m$. Then the error caused by the entire sequence of imperfect gates is at most the sum of the errors in the individual gates, namely $E(U_m U_{m-1}...U_1, V_m V_{m-1}...V_1) \leq \sum_{j=0}^{m} E(U_j, V_j)$ This statement will be proved by induction; let we fist start with the case m = 2: some states $|\psi\rangle$ we have:

$$E(U_2U_1, V_2V_1) = \|(U_2U_1 - V_2V_1)|\psi\rangle\| = \|(U_2U_1 - V_2U_1)|\psi\rangle + (V_2U_1 - V_2V_1)|\psi\rangle\| \tag{A.3}$$

Using the triangle inequality we finally obtain:

$$E(U_2U_1, V_2V_1) \leq \|(U_2 - V_2)U_1|\psi\rangle\| + \|V_2(U_1 - V_1)|\psi\rangle\| \leq E(U_2, V_2) + E(U_1, V_1) \tag{A.4}$$

Then by induction this statement can be proven for generic m. □

A.2 Equivariant Quantum Neural Network

In this section we will show with a simple example how to build an equivariant embedding and symmetrize the qubit gates in order to build an equivariant ansatz. Assume that we have data points $x = (x1, x2) \in \mathbb{R}^2$ which predictions respects coordinate-exchanging and simultaneous-sign-flip symmetries, formally:

$$y(x_1, x_2) = y(x_2, x_1) = y(-x_1, -x_2) \tag{A.5}$$

which correspond to a $\mathcal{G} \equiv \mathbb{Z}_2 \otimes \mathbb{Z}_2 = \{(0,0), (0,1), (1,0), (1,1)\}$ symmetry.

The group representation acting on the feature space $V_g : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ has four elements:

$$\begin{aligned} V_{(0,0)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad V_{(1,0)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\ V_{(0,1)} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad V_{(1,1)} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \end{aligned} \tag{A.6}$$

Assume the feature map of the circuit is of the form: $U(x_1, x_2) = R_Z(x_1) \otimes R_Z(x_2)$. It is known that in order to be equivariant a feature map must transform as:

$$U(V_g[x]) = U_g U(x) U_g^\dagger \quad , \tag{A.7}$$

where U_g unitary representation U_g that acts on the Hilbert space $U_g : \mathcal{H} \longrightarrow \mathcal{H}$ of the quantum system. In this case the feature map $U(x)$ transforms as:

$$\begin{aligned} U(x_2, x_1) = R_Z(x_2) \otimes R_Z(x_1) &= (\text{SWAP})U(x1, x2)(\text{SWAP}) \\ U(-x_1, -x_2) = R_Z(-x_1) \otimes R_Z(-x_2) &= (X \otimes X)U(x_1, x_2)(X \otimes X)` \\ U(-x_2, -x_1) = R_Z(-x_2) \otimes R_Z(-x_1) &= \text{SWAP}(X \otimes X)U(x_1, x_2)(X \otimes X)\text{SWAP} \end{aligned} \tag{A.8}$$

in this way it is possible to recover the values of $U_g \quad \forall g \in \mathcal{G}$:

$$\begin{aligned} U_{(0,0)} = \mathbb{1} \otimes \mathbb{1}, \quad U_{1,0} = \text{SWAP}, \\ U_{(0,1)} = X \otimes X, \quad U_{(1,1)} = \text{SWAP}(X \otimes X) \end{aligned} \tag{A.9}$$

Now by using gate symmetrization technique it is possible to convert an ansatz to an equivariant one by replacing the ansatz's gateset with its equivariant counterpart. In general the gateset is taken as the set of the Pauli matrices; since we are dealing with two features, that can be encoded into two qubits, we have that gateset $\Sigma = \{X_1, Y_1, Z_1, X_2, Y_2, Z_2, Z_1 Z_2\}$, which is not an equivariant gateset with respect to $\mathcal{G}$ because in general $\exists \sigma_i \in \Sigma, g \in \mathcal{G}$ such that $[\sigma_i, U_g] \neq 0$.
Hence we need to use twirling formula eq.(1.61) in order to project the gateset into an equivariant one. It is possible to check that $[Z_1 Z_2, U_g] = 0 \quad \forall g \in \mathcal{G}$, so $Z_1 Z_2$ operator in already an equivariant operator.

$$\begin{aligned} \mathcal{T}_U g[X \otimes \mathbb{1}] =& \frac{1}{4}\big[X \otimes \mathbb{1} + \text{SWAP}(X \otimes \mathbb{1})\text{SWAP} + (X \otimes X)(X \otimes \mathbb{1})(X \otimes X) \\ & + \text{SWAP}(X \otimes X)(X \otimes \mathbb{1})(X \otimes X)\text{SWAP}\big] \\ =& \frac{1}{4}\big[2(X \otimes \mathbb{1}) + 2(\mathbb{1} \otimes X)\big] \\ =& \frac{1}{2}\big[X_1 + X_2\big] \end{aligned} \tag{A.10}$$

it is easy to see that X_1 and X_2 give the same results. Now we symmetrize operator $Z \otimes \mathbb{1}$:

$$\begin{aligned} \mathcal{T}_U g[Z \otimes \mathbb{1}] =& \frac{1}{4}\big[Z \otimes \mathbb{1} + \text{SWAP}(Z \otimes \mathbb{1})\text{SWAP} + (X \otimes X)(Z \otimes \mathbb{1})(X \otimes X) \\ & + \text{SWAP}(X \otimes X)(Z \otimes \mathbb{1})(X \otimes X)\text{SWAP}\big] \\ =& \frac{1}{4}\big[Z \otimes \mathbb{1} + \mathbb{1} \otimes Z - Z \otimes \mathbb{1} - \mathbb{1} \otimes Z\big] = 0 \end{aligned} \tag{A.11}$$

Analogously for Z_2, Y_1 and Y_2. All these terms gets to 0 because $XYX = -Y$ and $XZX = -Z$. Then the equivariant gateset is the following:

$$\Gamma = \left\{ \frac{X_1 + X_2}{2}, Z_1 Z_2 \right\} \tag{A.12}$$

which has a much lower cardinality than the initial gateset Σ; This interesting property leads to a lower number of trainable parameters of the circuit and then to a lower complexity of the architecture.

Bibliography

[1] Albert Einstein, Boris Podolsky, and Nathan Rosen. "Can quantum-mechanical description of physical reality be considered complete?" In: *Physical review* 47.10 (1935), p. 777.

[2] David Bohm and Yakir Aharonov. "Discussion of experimental proof for the paradox of Einstein, Rosen, and Podolsky". In: *Physical Review* 108.4 (1957), p. 1070.

[3] John S Bell. "On the einstein podolsky rosen paradox". In: *Physics Physique Fizika* 1.3 (1964), p. 195.

[4] Alain Aspect, Philippe Grangier, and Gérard Roger. "Experimental tests of realistic local theories via Bell's theorem". In: *Physical review letters* 47.7 (1981), p. 460.

[5] Alain Aspect, Jean Dalibard, and Gérard Roger. "Experimental test of Bell's inequalities using time-varying analyzers". In: *Physical review letters* 49.25 (1982), p. 1804.

[6] Stuart J Freedman and John F Clauser. "Experimental test of local hidden-variable theories". In: *Physical Review Letters* 28.14 (1972), p. 938.

[7] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[8] Harrison Mitchell, Alexander Norcliffe, and Pietro Liò. "Learning Feynman Diagrams using Graph Neural Networks". In: *arXiv preprint arXiv:2211.15348* (2022).

[9] Richard J Trudeau. *Introduction to graph theory*. Courier Corporation, 2013.

[10] Martin Loebl. "Introduction to graph theory". In: *Discrete Mathematics in Statistical Physics: Introductory Lectures*. Springer, 2010, pp. 13–49.

[11] Shuicheng Yan et al. "Graph embedding and extensions: A general framework for dimensionality reduction". In: *IEEE transactions on pattern analysis and machine intelligence* 29.1 (2006), pp. 40–51.

[12] Amir Globerson et al. "Euclidean embedding of co-occurrence data". In: *Advances in neural information processing systems* 17 (2004).

[13] Fenxiao Chen et al. "Graph representation learning: a survey". In: *APSIPA Transactions on Signal and Information Processing* 9 (2020), e15.

[14] Blake Shaw and Tony Jebara. "Structure preserving embedding". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. 2009, pp. 937–944.

[15] Seongjun Yun et al. "Graph transformer networks". In: *Advances in neural information processing systems* 32 (2019).

[16] Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[17] Peter Wittek. *Quantum machine learning: what quantum computing means to data mining*. Academic Press, 2014.

[18] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. “Effect of data encoding on the expressive power of variational quantum-machine-learning models”. In: *Physical Review A* 103.3 (2021), p. 032430.

[19] Jacob Biamonte et al. “Quantum machine learning”. In: *Nature* 549.7671 (2017), pp. 195–202.

[20] M Cerezo et al. “Challenges and opportunities in quantum machine learning”. In: *Nature Computational Science* 2.9 (2022), pp. 567–576.

[21] Hsin-Yuan Huang et al. “Quantum advantage in learning from experiments”. In: *Science* 376.6598 (2022), pp. 1182–1186.

[22] Ryan LaRose and Brian Coyle. “Robust data encodings for quantum classifiers”. In: *Physical Review A* 102.3 (2020), p. 032420.

[23] Marco Cerezo et al. “Variational quantum algorithms”. In: *Nature Reviews Physics* 3.9 (2021), pp. 625–644.

[24] Adrián Pérez-Salinas et al. “Data re-uploading for a universal quantum classifier”. In: *Quantum* 4 (2020), p. 226.

[25] Zoë Holmes et al. “Connecting ansatz expressibility to gradient magnitudes and barren plateaus”. In: *PRX Quantum* 3.1 (2022), p. 010313.

[26] Kouhei Nakaji and Naoki Yamamoto. “Expressibility of the alternating layered ansatz for quantum computation”. In: *Quantum* 5 (2021), p. 434.

[27] Guillaume Verdon et al. “Quantum graph neural networks”. In: *arXiv preprint arXiv:1909.12264* (2019).

[28] Jin Zheng, Qing Gao, and Yanxuan Lü. “Quantum graph convolutional neural networks”. In: *2021 40th Chinese Control Conference (CCC)*. IEEE. 2021, pp. 6335–6340.

[29] Georgios Kollias et al. “Quantum Graph Transformers”. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2023, pp. 1–5.

[30] Slimane Thabet, Romain Fouilland, and Loic Henriet. “Extending Graph Transformers with Quantum Computed Aggregation”. In: *arXiv preprint arXiv:2210.10610* (2022).

[31] Michael Ragone et al. “Representation theory for geometric quantum machine learning”. In: *arXiv preprint arXiv:2210.07980* (2022).

[32] Martin Larocca et al. “Group-invariant quantum machine learning”. In: *PRX Quantum* 3.3 (2022), p. 030341.

[33] Johannes Jakob Meyer et al. “Exploiting symmetry in variational quantum machine learning”. In: *PRX Quantum* 4.1 (2023), p. 010328.

[34] Quynh T Nguyen et al. “Theory for equivariant quantum neural networks”. In: *arXiv preprint arXiv:2210.08566* (2022).

[35] Andrea Skolik et al. “Equivariant quantum circuits for learning on weighted graphs”. In: *npj Quantum Information* 9.1 (2023), p. 47.

[36] Steven Weinberg. *The quantum theory of fields: Volume 1, foundations.* Cambridge university press, 2005.

[37] Matthew D Schwartz. *Quantum field theory and the standard model.* Cambridge university press, 2014.

[38] Freeman J Dyson. “The S matrix in quantum electrodynamics”. In: *Physical Review* 75.11 (1949), p. 1736.

[39] Franz Mandl and Graham Shaw. *Quantum field theory.* John Wiley & Sons, 2010.

[40] Ta-Pei Cheng and Ling-Fong Li. *Gauge theory of elementary particle physics.* Oxford university press, 1994.

[41] Pierre Ramond. *Field theory: a modern primer.* Routledge, 2020.

[42] Ville Bergholm et al. "Pennylane: Automatic differentiation of hybrid quantum-classical computations". In: *arXiv preprint arXiv:1811.04968* (2018).

[43] Rolf Mertig. "Guide to FeynCalc 1.0". In: *University of Würzburg* (1992).

[44] Thomas Hahn. "Generating Feynman diagrams and amplitudes with FeynArts 3". In: *Computer Physics Communications* 140.3 (2001), pp. 418–431.

[45] Thomas Hahn and M Perez-Victoria. "Automated one-loop calculations in four and D dimensions". In: *Computer Physics Communications* 118.2-3 (1999), pp. 153–165.

[46] T Hahn. "Generating and calculating one-loop Feynman diagrams with FeynArts, FormCalc, and LoopTools". In: *arXiv preprint hep-ph/9905354* (1999).

[47] Johan Alwall et al. "MadGraph 5: going beyond". In: *Journal of High Energy Physics* 2011.6 (2011), pp. 1–40.

[48] Torbjörn Sjöstrand et al. "An introduction to PYTHIA 8.2". In: *Computer physics communications* 191 (2015), pp. 159–177.

[49] Alexander Belyaev, Neil D Christensen, and Alexander Pukhov. "CalcHEP 3.4 for collider physics within and beyond the Standard Model". In: *Computer Physics Communications* 184.7 (2013), pp. 1729–1769.

[50] Florian Bury and Christophe Delaere. "Matrix element regression with deep neural networks—Breaking the CPU barrier". In: *Journal of High Energy Physics* 2021.4 (2021), pp. 1–26.

[51] Matthew Feickert et al. "Deep Learning for the Matrix Element Method". In: *arXiv preprint arXiv:2211.11910* (2022).

[52] Yongxin Jing et al. "A deep neural network for general scattering matrix". In: *Nanophotonics* (2023). DOI: `doi:10.1515/nanoph-2022-0770`. URL: `https://doi.org/10.1515/nanoph-2022-0770`.

[53] M Crispim Romão, Nuno Filipe Castro, and R Pedro. "Finding new physics without learning about it: anomaly detection as a tool for searches at colliders". In: *The European Physical Journal C* 81.1 (2021), p. 27.

[54] Jack Collins, Kiel Howe, and Benjamin Nachman. "Anomaly detection for resonant new physics with machine learning". In: *Physical review letters* 121.24 (2018), p. 241803.

[55] Yanhu Chen, Cen Wang, Hongxiang Guo, et al. "Novel architecture of parameterized quantum circuit for graph convolutional network". In: *arXiv preprint arXiv:2203.03251* (2022).

Ringraziamenti - Acknowledgements

Ai team di QISMIB e CERN QTI che mi hanno seguito durante la tesi:

Ringrazio vivamente il mio relatore interno Dr. Andrea Giachero per aver accettato e seguito questo progetto di tesi in fisica teorica, ma soprattutto lo ringrazio per avermi permesso di passare tre mesi con il Dr. Michele Grossi, la Dr. Sofia Vallecorsa e tutto il gruppo di CERN QTI, un'esperienza indimenticabile. Un ringraziamento anche a Roberto Moretti del team QISMIB di Milano Bicocca, il quale mi ha seguito e aiutato per tutto questo percorso con enorme disponibilità, le piacevoli discussioni con lui son sempre state fonte di ottimi spunti di riflessioni. Un ringraziamento speciale al mio relatore esterno Dr. Michele Grossi, il quale ha seguito incessantemente i miei progressi con passione ed interesse; lo ringrazio per i momenti di confronto e di lavoro, durante i quali ho provato ad assimilare quanto più potevo da lui, e per ogni singolo momento di allegria e spensieratezza condiviso. Ringrazio inoltre la Dr.Sofia Vallecorsa, la quale mi ha permesso di fare questa esperienza al CERN e mi ha dato interessanti spunti di riflessione. Ringrazio anche Massimiliano Incudini, che e mi è sempre stato d'aiuto durante tutta la tesi, mettendoci passione, simpatia e dedizione, ma soprattutto mi ha formato e mi ha aiutato ad ambientarmi durante le prime settimane di soggiorno al CERN. An equally warm thank you to Vasilis Belis, who joined the working group showing dedication and great interest in the project; I also thank him for all the shared moments of leisure.
I would like to thank all the people I have met while working at CERN Openlab and with whom I have shared very stimulating and enjoyable moments of leisure and exchange of ideas; in particular, I thank Matteo Bunino, Roman, Anastasiya, Carla Rieger, Su Yeoun Chang, Tobias Duswald, Francesco Barone, Lukas Breitwieser and Marina Banjac.

Ai miei tre nonni Mene, Gigi e Achille: vi sento sempre presenti, ad illunimare il cammino con i vostri insegnamenti. Mi avete insegnato a sorridere alla vita, l'importanza dell'amicizia e di credere sempre nella bontà delle persone. Il vostro ricordo inoltre mi insegna ogni giorno l'importanza del tempo.
Alle mie nonne Marisa, Maria Ida e Bruna: mi fate sempre sentire speciale; negli anni mi avete cresciuto insegnandomi la pazienza e l'affetto incondizionato. Vedete sempre la bellezza e la bontà in me, spronandomi a migliorare come persona, e per tutto questo vi ringrazio.

Ai miei fratelli Tommaso, Rebecca, Davide e Chiara: vi vedo crescere e penso sempre a quando giocavamo insieme in giardino, o quando vi tenevo in braccio da piccoli, e vi ringrazio per essere i miei fratelli. Mi avete insegnato la bellezza della condivisione, perciò voglio condividere questo traguardo con voi, che con il vostro sorriso, le nostre litigate, i nostri bellissimi ricordi, le vostre persone, non avrei raggiunto. Vi voglio bene.

Ai miei genitori: siete la mia perpetua fonte d'ispirazione, dalla vostra dedizione e senso del dovere alla vostra saggezza, mi avete modellato nella persona che sono ora, e per questo vi ringrazio. Siete sempre stati i primi a credere in me, anche quando io stesso ero il primo dei miei detrattori, tanto da permettermi di inseguire i miei sogni e le mie ambizioni. Mi avete donato una vita meravigliosa, sempre traboccante di allegria e spensieratezza e per cui son pronto a lottare per dimostrare di meritarla.