



DOCTORAT EN FÍSICA

Quantum computing applications in High Energy Physics: clustering, integration and generative models

Author:

Jorge Juan Martínez de Lejarza Samper

Supervisors:

Dr. Germán Vicente Rodrigo García

Dr. Leandro Javier Cieri

Departament de Física Teòrica
IFIC, Universitat de València - CSIC

Programa de Doctorat en Física - 3126

València, August 2025

Title:

Quantum computing applications in High Energy Physics: clustering, integration and generative models

Author:

Jorge Juan Martínez de Lejarza Samper

Departament de Física Teòrica

Instituto de Física Corpuscular (IFIC)

Universitat de València, Spain

Consejo Superior de Investigaciones Científicas (CSIC), Spain

Supervisors:

Dr. Germán Vicente Rodrigo García

Instituto de Física Corpuscular (IFIC)

Universitat de València, Spain

Consejo Superior de Investigaciones Científicas (CSIC), Spain

and

Dr. Leandro Javier Cieri

Departament de Física Teòrica

Instituto de Física Corpuscular (IFIC)

Universitat de València, Spain

Consejo Superior de Investigaciones Científicas (CSIC), Spain

Copyright © 2024 Jorge Juan Martínez de Lejarza Samper

This research has been carried out at Instituto de Física Corpuscular, IFIC, Valencia, which is a joint centre of research between the Spanish Research Council (Consejo Superior de Investigaciones Científicas, CSIC) and the University of Valencia.

Dr. Germán Vicente Rodrigo García, Investigador Científico del Consejo Superior de Investigaciones Científicas y **Dr. Leandro Javier Cieri**, Investigador CIDEAGENT por la Universidad de Valencia

CERTIFICAN,

Que la presente memoria, “*Quantum computing applications in High Energy Physics: clustering, integration and generative models*”, ha sido realizada bajo su dirección en el Instituto de Física Corpuscular (Centro Mixto Universidad de Valencia - CSIC) por **Jorge Juan Martínez de Lejarza Samper** y constituye su Tesis Doctoral para optar al título de Doctor por la Universitat de València una vez cursados los estudios en el Doctorado en Física.

Y para que conste, en cumplimiento de la legislación vigente, presenta en la Universitat de València la referida Tesis Doctoral, y firma el presente certificado, en

Valencia, Agosto 2025

Dr. Germán Vicente Rodrigo García

Dr. Leandro Javier Cieri

We live on an island surrounded by a sea of ignorance. As our island of knowledge grows, so does the shore of our ignorance. - John Archibald Wheeler

Life before death. Strength before weakness. Journey before destination.
- *Brandon Sanderson*

*A ti, por ser
hogar en
cualquier lugar*

Acknowledgements

En estos cuatro años de doctorado he vivido una etapa que me ha enriquecido profundamente, tanto a nivel académico y profesional como a nivel personal. Este camino me ha permitido enfrentarme a retos intelectuales que me han estimulado y formado académicamente, pero también me ha brindado la oportunidad de asistir a conferencias y realizar estancias por todo el mundo, descubriendo otras culturas, otras formas de hacer ciencia, y conociendo a personas increíbles con las que compartir ideas, debatir y de las que aprender. Cada conversación, cada viaje, cada reto superado ha contribuido a darme más perspectiva y a dar forma a la persona que soy hoy. Siento que acabo este doctorado como una persona más madura, con ideas y objetivos vitales más claros, y convencido de que todo este proceso me ha transformado de una forma que difícilmente habría alcanzado por otros caminos. Por todo ello, me siento profundamente agradecido.

En primer lugar, me gustaría agradecer a Germán por ser un excelente supervisor y por encontrar siempre tiempo para mí y mis proyectos. Su infinita sabiduría y su capacidad para transformar proyectos en publicaciones han sido una fuente constante de inspiración. Desde el primer momento me he sentido muy arropado trabajando con él, siempre abierto a que comenzara nuevos proyectos y apoyándome para viajar a escuelas y conferencias para complementar mi aprendizaje en esta etapa. Ahora me toca comenzar una nueva etapa como investigador más independiente pero de verdad espero que nuestros caminos científicos vuelvan a cruzarse en un futuro no muy lejano.

En la misma línea, también quiero agradecer a Leandro por su apoyo como co-director, por sus consejos y por saber transmitir calma y esa característica “buena onda” en los momentos más necesarios. Asimismo, deseo agradecer a la Generalitat Valenciana por su apoyo a través del proyecto No. ACIF/2021/219.

I would also like to thank Michele Grossi, for his constant help and interest in keeping a very fruitful collaboration since we started working together at the QT4HEP conference at CERN in 2022. You have always been there for me as an unofficial PhD co-supervisor and I am very grateful for that.

Still from a scientific perspective, although not related to the content of the thesis, I would like to thank Stefano Carrazza from hosting me during my time at University of Milan working on `Qibo`. I am also grateful to Fabio Scafrimuto for giving me the opportunity to join IBM Quantum for an internship in Zurich during the final months of my PhD, where I got to experience first-hand how is to work at one of the world leaders in quantum computing. Also to Alvaro Ballon for giving me the amazing opportunity of doing an internship at Xanadu to work with him in Toronto for what it has been one of the best summers of my life so far!

The next lines are for those who might not have affected directly to the content of this thesis but have definitely contributed to my happiness and well-being throughout the journey.

First, I would like to thank the huge squad of “El Rincón del IATA” at IFIC, from those who were there when I started and have already left (Gui, Selo, Fabián, Ele, Drona and Chitra, Ema and Pier, Andrés, Masha) to those who started after and will still be there when I am gone (Baibhab, Androniki, Juliana, Dom, Kostas, David, Muñoz, Andrea, Aure, Rafa, Arnau, Miguel, Sergio, Juanma). For sharing time over seminars, lunches, picaetas, conferences, trips, geotastic, petanca, M&M’s and more! Then, I also want to thank all

the people I have met during my research stays, internships, conferences that have made a significant impact in my PhD journey. That includes the people from Santander (Celia, Pablo, Carmen) and Oviedo (Andrea, Carlos, Alejandro, Clara), the QTI people at CERN (Matteo, Oriel, Vasilis, Alice, Ema), my fellow ex-residents at Xanadu (Emiliano, Ahmed, Siddhu, Praveen, Kasper, Serene, Juan), and many more names I cannot fit into these lines.

La siguiente línea va para Guillermo Alonso. Recuerdo que empecé el doctorado viendo y aprendiendo de tus vídeos en YouTube sobre computación cuántica, ya que al principio no tenía ni idea. Por eso, poder acabar yendo a Xanadu, conocerte en persona, poder trabajar contigo y aprender de ti y, además, poder considerarte ahora un amigo, ha sido una experiencia muy bonita que me llevo de estos años.

También me gustaría agradecer a mis amigos de la carrera (Pastor, Alicia, Rebeca, Pablet, Zanón, Miró, Ana) por seguir siendo parte de mi vida durante estos años de doctorado. Entre cenas de sushi, pizzas o hamburguesas me transmitíais como era la vida del físico más allá del doctorado y la academia.

Los siguientes en la línea de agradecimientos no pueden ser otros que los integrantes de BOM: Raúl, Jorge, Dani, Adolfo, Iván, Seguí, Papeco y Jaime. Habéis sido, sois y seréis siempre un pilar fundamental en mi vida y lo primero que me viene a la cabeza cuando pienso en el término “mejores amigos”. Gracias por las risas, nuestras coñas y vocabulario que nadie más entiende, los viajes BOM, y por hacer esta etapa un poco más fácil cada día.

Para acabar, me gustaría dar las gracias a mi familia por esta tesis. A mi padre, por haberme inculcado desde pequeño esa curiosidad y pasión por el conocimiento y la ciencia, y por haberme servido de modelo académico a seguir. A mi hermano, a quien siempre he admirado, por su ayuda a nivel académico desde el colegio hasta la universidad y cuyos pasos he ido siguiendo de una manera casi inconsciente. A mi madre, por vivir con orgullo cada pequeño logro que iba consiguiendo y por creer siempre en mí incluso en los momentos en los que yo mismo dudaba. También a mi familia extendida Salva y Sandra por acogerme como uno más y por su constante implicación y apoyo en este camino.

Finalmente, solo puedo acabar estos agradecimientos nombrando a una persona. Gracias Paula por ser mi apoyo incondicional, mi fan número uno, y la mejor compañera de vida que uno podría pedir. Gracias por ser capaz siempre de sacarme una sonrisa, incluso en los días más complicados. Tu forma de ser y tu energía han sido siempre el mejor antídoto contra el estrés y lo que siempre ha sido capaz de alegrar mis días. Gran parte de lo que he conseguido ha sido gracias a ti, y me siento tremendamente afortunado de tenerte a mi lado y de poder seguir compartiendo contigo cada nueva etapa de mi vida.

Siempre serás mi hogar.

Preface

Abstract

This thesis investigates the potential of quantum computing to address key computational challenges in high-energy physics (HEP), particularly in light of the increasing complexity of modern particle physics experiments and the continued absence of experimental discoveries beyond the Standard Model (SM). Despite its remarkable success, the SM leaves several fundamental questions unresolved, such as the origin of neutrino masses, the nature of dark matter, and the matter-antimatter asymmetry. In this regard, the role of advanced experimental computational tools becomes more crucial than ever, both for analyzing the vast volumes of data produced in collider experiments and for improving the accuracy and reach of theoretical predictions.

Quantum computing, inspired by Richard Feynman’s vision of simulating quantum systems with quantum machines, offers a paradigm that could outperform classical methods in specific domains. This thesis explores how quantum algorithms can be applied to problems in HEP, which is fundamentally governed by quantum field theory. The contributions span three main research areas: quantum clustering, quantum Monte Carlo integration, and quantum generative modeling for multivariate distributions.

Chapter 1 provides a brief historical introduction of quantum mechanics, quantum field theory and quantum computing and outlines the motivation for this work.

Chapter 2 presents a concise overview of the SM. It introduces the elementary particles and their interactions, and then describes the quantum field theories that form the SM by examining the different terms in its Lagrangian: Quantum Chromodynamics (QCD) for the strong interaction, the Electroweak (EW) theory unifying electromagnetic and weak interactions, the Higgs sector responsible for EW symmetry breaking, and the Yukawa terms generating fermion masses.

Chapter 3 introduces the basics of QC from a software point of view. First, we highlight the differences between quantum bits (qubits) and classical bits, as well as between quantum gates and their classical counterparts. Then, we explain the structure of quantum algorithms and how they are represented through quantum circuits, illustrating key examples relevant to this thesis, such as the *SwapTest*, Grover’s algorithm, and variational algorithms. Finally, it provides an introduction to QML, detailing some of the most widely used methods in the field.

The core results of this work are presented in Chapters 4, 5, and 6, which focus on applications of quantum algorithms to relevant problems in particle physics. First, in Chapter 4, we introduce two quantum subroutines, one for computing Minkowski-based distances and another for finding the maximum in an unsorted list, which are then applied to a jet clustering problem using simulated LHC data. Chapter 5 introduces Quantum

Fourier Iterative Amplitude Estimation (QFIAE), a quantum Monte Carlo integrator that combines a quantum neural network for Fourier decomposition with Quantum Amplitude Estimation, enabling an end-to-end quantum pipeline without classical preprocessing. This method is validated by computing scattering amplitudes, Feynman loop integrals via the Loop–Tree Duality formalism, and NLO (Next-to Leading Order) decay rates for physical processes on both simulators and real quantum devices. Finally, in Chapter 6 we propose a quantum Chebyshev probabilistic model designed to learn probability distributions, offering exponential resolution improvements by adding extra qubits. Then, the model is applied to learn and generate fragmentation functions of partons fragmenting into kaons and pions.

The thesis concludes in Chapter 7 with a discussion of the implications of these results, potential pathways toward finding useful quantum applications in HEP, and future research directions as quantum hardware progresses toward fault-tolerant architectures. Chapter 8 presents a summary of the thesis in Spanish, and the appendices provide further details on selected topics from Chapters 4 to 6.

Overall, these results show how quantum algorithms can already tackle relevant HEP problems on current hardware, while paving the way for future fault-tolerant applications that fully exploit quantum computational advantages.

Resumen

Esta tesis estudia el potencial de la computación cuántica para abordar desafíos computacionales clave en física de altas energías, en un contexto marcado por la creciente complejidad de los experimentos modernos en física de partículas y la ausencia continuada de descubrimientos experimentales más allá del Modelo Estándar. A pesar de su notable éxito, el Modelo Estándar deja sin resolver cuestiones fundamentales como el origen de las masas de los neutrinos, la naturaleza de la materia oscura o la asimetría entre materia y antimateria. En este sentido, el papel de las herramientas computacionales avanzadas resulta hoy más esencial que nunca, tanto para analizar los enormes volúmenes de datos generados en experimentos de colisionadores como para mejorar la precisión y el alcance de las predicciones teóricas.

La computación cuántica, inspirada en la visión de Richard Feynman de simular sistemas cuánticos mediante ordenadores cuánticos, ofrece un paradigma con potencial para superar a los métodos clásicos en casos concretos. En esta tesis se explora cómo los algoritmos cuánticos pueden aplicarse a problemas en física de altas energías, gobernados de manera fundamental por la teoría cuántica de campos. Las contribuciones de esta tesis se organizan en tres áreas principales: técnicas de *clustering* cuántico, integración cuántica de Monte Carlo y modelado generativo cuántico de distribuciones multidimensionales.

El Capítulo 1 presenta una breve introducción histórica a la mecánica cuántica, la teoría cuántica de campos y la computación cuántica, y expone la motivación de este trabajo.

En el Capítulo 2 se ofrece una visión global del Modelo Estándar. Se introducen las partículas elementales y sus interacciones, para después describir las teorías cuánticas de campos que lo conforman, a través de los distintos términos de su lagrangiano: la Cromodinámica Cuántica, que describe la interacción fuerte; la teoría Electrodébil, que unifica las interacciones electromagnética y débil; el sector de Higgs, responsable de la ruptura espontánea de simetría electrodébil; y los términos de Yukawa, que generan las masas de los fermiones mediante su interacción con el campo de Higgs.

El Capítulo 3 introduce los fundamentos de la computación cuántica desde un punto de vista de software. Se comparan los bits cuánticos (*qubits*) con los bits clásicos, así como las puertas cuánticas con sus equivalentes clásicas. A continuación, se describe la estructura de los algoritmos cuánticos y su representación mediante circuitos cuánticos, ilustrando ejemplos relevantes para esta tesis como el *SwapTest*, el algoritmo de Grover y los algoritmos variacionales. El capítulo concluye con una introducción al aprendizaje automático cuántico (QML), repasando algunos de los métodos más empleados en este ámbito.

Los resultados principales se presentan en los Capítulos 4, 5 y 6, centrados en la aplicación de algoritmos cuánticos a problemas de interés en física de partículas. En el Capítulo 4 se introducen dos subrutinas cuánticas: una para calcular distancias basadas en la métrica de Minkowski y otra para localizar el valor máximo en una lista no ordenada; ambas se aplican al problema de *jet clustering* empleando datos simulados del LHC. El Capítulo 5 presenta el integrador cuántico de Monte Carlo denominado *Quantum Fourier Iterative Amplitude Estimation* (QFIAE), que combina una red neuronal cuántica para descomposición en series de Fourier con la técnica de *Quantum Amplitude Estimation*, lo que permite un flujo de trabajo íntegramente cuántico sin preprocesamiento clásico. El método se valida mediante el cálculo de amplitudes de dispersión, integrales de bucle de Feynman con el formalismo *Loop-Tree Duality* (LTD) y tasas de decaimiento a NLO (*Next-to Leading Order*) en teoría cuántica de campos perturbativa, empleando tanto simuladores como dispositivos cuánticos

reales. Por último, en el Capítulo 6 se propone un modelo probabilístico cuántico basado en polinomios de Chebyshev, diseñado para aprender distribuciones de probabilidad y ofrecer mejoras exponenciales de resolución mediante la adición de qubits extra. El modelo se aplica al aprendizaje y generación de funciones de fragmentación de partones en kaones y piones.

La tesis concluye en el Capítulo 7 con una discusión sobre el alcance de los resultados, posibles vías para identificar aplicaciones cuánticas útiles en HEP y las líneas de investigación futuras a medida que el hardware cuántico avanza hacia arquitecturas tolerantes a fallos. El Capítulo 8 presenta un resumen en castellano, mientras que los apéndices recogen detalles adicionales de los temas tratados en los Capítulos 4 a 6.

En conjunto, los resultados muestran que, incluso con el hardware actual, los algoritmos cuánticos pueden abordar ya problemas relevantes en HEP, sentando a la vez las bases para aplicaciones futuras tolerantes a fallos que exploten plenamente las ventajas de la computación cuántica.

Resum

Aquesta tesi estudia el potencial de la computació quàntica per a abordar desafiaments computacionals clau en física d'altres energies, en un context marcat per la creixent complexitat dels experiments moderns en física de partícules i l'absència continuada de descobriments experimentals més enllà del Model Estàndard. Malgrat el seu notable èxit, el Model Estàndard deixa sense resoldre qüestions fonamentals com l'origen de les masses dels neutrins, la naturalesa de la matèria fosca o l'asimetria entre matèria i antimatèria. En aquest sentit, el paper de les eines computacionals avançades resulta hui més essencial que mai, tant per a analitzar els enormes volums de dades generats en experiments de col·lisionadors com per a millorar la precisió i l'abast de les prediccions teòriques.

La computació quàntica, inspirada en la visió de Richard Feynman de simular sistemes quàntics mitjançant ordinadors quàntics, ofereix un paradigma amb potencial per a superar els mètodes clàssics en casos concrets. En aquesta tesi s'explora com els algorismes quàntics poden aplicar-se a problemes en física d'altres energies, governats de manera fonamental per la teoria quàntica de camps. Les contribucions d'aquesta tesi s'organitzen en tres àrees principals: tècniques de *clustering* quàntic, integració quàntica de Monte Carlo i modelatge generatiu quàntic de distribucions multidimensionals.

El Capítol 1 presenta una breu introducció històrica a la mecànica quàntica, la teoria quàntica de camps i la computació quàntica, i exposa la motivació d'aquest treball.

En el Capítol 2 s'ofereix una visió global del Model Estàndard. S'introdueixen les partícules elementals i les seues interaccions, per a després descriure les teories quàntiques de camps que el conformen, a través dels diferents termes del seu lagrangià: la Cromodinàmica Quàntica, que descriu la interacció forta; la teoria Electrodèbil, que unifica les interaccions electromagnètica i dèbil; el sector de Higgs, responsable de la ruptura espontània de simetria electrodèbil; i els termes de Yukawa, que generen les masses dels fermions mitjançant la seua interacció amb el camp de Higgs.

El Capítol 3 introdueix els fonaments de la computació quàntica des d'un punt de vista de *software*. Es comparen els bits quàntics (*qubits*) amb els bits clàssics, així com les portes quàntiques amb les seues equivalents clàssiques. A continuació, es descriu l'estructura dels algorismes quàntics i la seua representació mitjançant circuits quàntics, il·lustrant exemples rellevants per a aquesta tesi com el *SwapTest*, l'algoritme de Grover i els algorismes variacionals. El capítol conclou amb una introducció a l'aprenentatge automàtic quàntic (QML), revisant alguns dels mètodes més emprats en aquest àmbit.

Els resultats principals es presenten en els Capítols 4, 5 i 6, centrats en l'aplicació d'algorismes quàntics a problemes d'interés en física de partícules. En el Capítol 4 s'introdueixen dues subrutines quàntiques: una per a calcular distàncies basades en la mètrica de Minkowski i una altra per a localitzar el valor màxim en una llista no ordenada; ambdues s'apliquen al problema de *jet clustering* emprant dades simulades del LHC. El Capítol 5 presenta l'integrador quàntic de Monte Carlo denominat *Quantum Fourier Iterative Amplitude Estimation* (QFIAE), que combina una xarxa neuronal quàntica per a descomposició en sèries de Fourier amb la tècnica de *Quantum Amplitude Estimation*, cosa que permet un flux de treball íntegrament quàntic sense preprocessament clàssic. El mètode es valida mitjançant el càlcul d'amplituds de dispersió, integrals de bucle de Feynman amb el formalisme *Loop-Tree Duality* (LTD) i taxes de desintegració a NLO (*Next-to Leading Order*) en teoria quàntica de camps pertorbativa, emprant tant simuladors com dispositius quàntics reals. Finalment, en el Capítol 6 es proposa un model probabilístic quàntic basat en polinomis de Chebyshev, dissenyat per a aprendre distribucions de probabilitat i oferir

milliores exponencials de resolució mitjançant l'addició de *qubits* extra. El model s'aplica a l'aprenentatge i generació de funcions de fragmentació de partons en kaons i pions.

La tesi conclou en el Capítol 7 amb una discussió sobre l'abast dels resultats, possibles vies per a identificar aplicacions quàntiques útils en HEP i les línies d'investigació futures a mesura que el maquinari quàntic avança cap a architectures tolerants a fallades. El Capítol 8 presenta un resum en castellà, mentre que els apèndixs recullen detalls addicionals dels temes tractats en els Capítols 4 a 6.

En conjunt, els resultats mostren que, fins i tot amb el maquinari actual, els algorismes quàntics poden abordar ja problemes rellevants en HEP, assentant alhora les bases per a aplicacions futures tolerants a fallades que aprofiten plenament els avantatges de la computació quàntica.

List of publications

This PhD thesis is based on the following publications:

Articles

- [1] **Jorge J. Martínez de Lejarza**, Hsin-Yu Wu, Oleksandr Kyriienko, Germán Rodrigo, Michele Grossi, *Quantum Chebyshev Probabilistic Models for Fragmentation Functions* [arXiv:2503.16073](#) (2025).
- [2] **Jorge J. Martínez de Lejarza**, David F. Rentería-Estrada, Michele Grossi, Germán Rodrigo, *Quantum integration of decay rates at second order in perturbation theory*, *Quantum Sci. Technol.* **10** 025026, [arXiv:2409.12236](#) (2024).
- [3] **Jorge J. Martínez de Lejarza**, Leandro Cieri, Michele Grossi, Sofia Vallecorsa, Germán Rodrigo, *Loop Feynman integration on a quantum computer*, *Phys. Rev. D* **110** 074031, [arXiv:2401.03023](#) (2024).
- [4] Selomit Ramírez-Urbe, Andrés E. Rentería-Olivo, David F. Rentería-Estrada, **Jorge J. Martínez de Lejarza**, Prasanna K. Dhani, Leandro Cieri, Roger J. Hernández-Pinto, German F.R. Sborlini, William J. Torres Bobadilla, Germán Rodrigo, *Vacuum amplitudes and time-like causal unitary in the loop-tree duality*, *JHEP* **01** 103, [arXiv:2404.05492](#) (2024).
- [5] **Jorge J. Martínez de Lejarza**, Michele Grossi, Leandro Cieri, Germán Rodrigo, *Quantum Fourier Iterative Amplitude Estimation*, [2023 IEEE International Conference on Quantum Computing and Engineering \(QCE\)](#), Bellevue, WA, USA, pp. 571-579, [arXiv:2305.01686](#) (2023) .
- [6] **Jorge J. Martínez de Lejarza**, Leandro Cieri, Germán Rodrigo, *Quantum clustering and jet reconstruction at the LHC*, *Phys. Rev. D* **106** 036021, [arXiv:2204.06496](#) (2022).
- [7] Andrea Delgado, Kathleen E. Hamilton, Prasanna Date, Jean-Roch Vlimant, Duarte Magano, Yasser Omar, Pedrame Bargassa, Anthony Francis, Alessio Gianelle, Lorenzo Sestini, Donatella Lucchesi, Davide Zuliani, Davide Nicotra, Jacco de Vries, Dominica Dibenedetto, Miriam Lucio Martinez, Eduardo Rodrigues, Carlos Vazquez Sierra, Sofia Vallecorsa, Jesse Thaler, Carlos Bravo-Prieto, Su Yeon Chang, Jeffrey Lazar, Carlos A. Argüelles, **Jorge J. Martinez de Lejarza**, *Quantum Computing for Data Analysis in High-Energy Physics*, [arXiv:2203.08805](#) (2022).

Conference proceedings

- [8] **Jorge J. Martínez de Lejarza**, Leandro Cieri, Germán Rodrigo, *Quantum jet clustering with LHC simulated data*, *PoS ICHEP2022* **241**, [arXiv:2209.08914](#) (2022).

Other research works carried out during the course of PhD

- [9] Konstantinos Pyretzidis, **Jorge J. Martínez de Lejarza**, Germán Rodrigo, *Unlocking Multi-Dimensional Integration with Quantum Adaptive Importance Sampling* [arXiv:2506.19965](#) (2025).

Different talks about the topic of this thesis were also given at the following conferences and events:

- Oral presentation of a technical paper in “Quantum Technologies for High-Energy Physics (QT4HEP25)”: [Quantum Chebyshev Generative modeling for fragmentation functions](#) (2025).
- Oral presentation during poster session on *Quantum Techniques for Machine Learning* (QTML24): [Quantum integration of decay processes at high-energy colliders](#) (2024).
- Oral presentation during poster session on *Quantum Techniques for Machine Learning* (QTML23): [Quantum integration of Feynman loop integrals](#) (2023).
- Invited speaker to give a seminar at CERN QTI-TH Forum: [Quantum Fourier Iterative Amplitude Estimation](#) (2023).
- Oral presentation of a technical paper in *IEEE Quantum Week 2023*: [Quantum Fourier Iterative Amplitude Estimation](#) (2023).
- Oral presentation during poster session on Spring School on *Quantum Information Processing – Applications on Gate-based and Annealing Systems*: [Quantum jet clustering with LHC simulated data](#) (2023).
- Invited speaker to give a seminar on *2nd workshop of AI Initiative for Science: Quantum Machine Learning for Particle Physics* (2022).
- Oral presentation during poster session on *International Conference on Quantum Technologies for High-Energy Physics* (QT4HEP22): [Quantum jet clustering with LHC simulated data](#) (2022).
- Oral presentation on *International Conference on High Energy Physics* (ICHEP): [Quantum clustering and jet reconstruction at the LHC](#) (2022).

Science communication

Different events or material I have created during my thesis about science communication in quantum computing:

- **Jorge J. Martínez de Lejarza** *Lab 0: Hello Quantum World! - Qiskit Global Summer School 2025* (2025).
- **Jorge J. Martínez de Lejarza**, Alberto Maldonado *Lab 2: Cutting through the noise - Qiskit Global Summer School 2025* (2025).
- **Jorge J. Martínez de Lejarza** *PennyLane Codebook: Variational Quantum Algorithms* (2024).
- **Jorge J. Martínez de Lejarza**, Guillermo Alonso-Linaje *PennyLane Demo: How to use quantum arithmetic operators* (2024).
- **Jorge J. Martínez de Lejarza**, Serene Shum *PennyLane Demo: Multidimensional regression with a variational quantum circuit* (2024).
- Assistant in a hands-on tutorial in “Quantum Machine Learning” in: [Navigating Quantum and AI Career Trajectories: A Beginner’s Mini-Course on Computational Methods and their Applications](#) at Perimeter Institute (2024).
- **Jorge J. Martínez de Lejarza**, *Tutorial: Quantum Fourier Iterative Amplitude Estimation* (2023).
- Outreach presentation with a live demo to a non-technical audience within the Quantum Spain project about [Quantum Computing and the protocol BB84](#) (2023).

Contents

Acknowledgements	v
Preface	xiii
Publications	xvi
1 Introduction	1
1.1 Historical introduction	1
1.2 Motivation	3
2 Standard Model	5
2.1 The Standard Model of particle physics in a nutshell	5
2.2 A Lagrangian to rule them all	7
2.2.1 Quantum Chromodynamics	7
2.2.2 Electroweak interactions	8
2.2.3 Electroweak Spontaneous Symmetry Breaking	9
2.2.4 Yukawa sector	11
2.3 Precision Physics in the Standard Model	11
3 Quantum Computing	13
3.1 Classical and quantum bits	13
3.2 Classical and quantum gates	15
3.3 Quantum algorithms	17
3.3.1 Swap Test	18
3.3.2 Grover's Algorithm	20
3.3.3 Variational Quantum Algorithms	24
3.3.3.1 Ansatz	24
3.3.3.2 Cost functions	25
3.3.3.3 Optimizer and Gradient Descent	25
3.4 Quantum Machine Learning	29
3.4.1 Encoding	30
3.4.1.1 Basis encoding	30
3.4.1.2 Amplitude encoding	30
3.4.1.3 Angle encoding	31
3.4.1.4 Dynamic encoding	31
3.4.2 Quantum Kernels	32
3.4.3 Quantum Neural Networks	34
3.4.4 Quantum Generative Models	38

4	Quantum clustering and jet reconstruction at the LHC	41
4.1	Introduction	41
4.2	Jets	42
4.3	Clustering algorithms	44
4.3.1	K-means algorithm	44
4.3.2	Affinity Propagation algorithm	45
4.3.3	Generalised k_T -jet algorithm	46
4.4	Quantum algorithms	47
4.4.1	Quantum distance	47
4.4.1.1	Euclidean quantum distance	47
4.4.1.2	Quantum invariant sum squared in Minkowski space	48
4.4.2	Quantum maximum searching algorithm	49
4.5	Quantum clustering algorithms	51
4.5.1	Quantum K-means with Minkowski-type distance	51
4.5.2	Quantum Affinity Propagation algorithm	54
4.5.3	Quantum k_T jet algorithm	55
4.6	Conclusions	57
5	Quantum integration	59
5.1	Introduction	59
5.2	Loop Feynman integrals in LTD	61
5.2.1	Loop-Tree Duality	62
5.3	Quantum Fourier Iterative Amplitude Estimation	63
5.3.1	Iterative Quantum Amplitude Estimation	64
5.3.2	Quantum Monte Carlo Integration	65
5.3.3	Fourier Quantum Monte Carlo Integration	66
5.3.4	Quantum Fourier Series	67
5.3.5	Quantum Fourier Iterative Amplitude Estimation	69
5.3.6	Benchmark example: integral of $1 + x^2$	70
5.4	Loop Feynman integration on a quantum computer	73
5.4.1	QFIAE implementation	73
5.4.1.1	Simulators implementation	73
5.4.1.2	Hardware implementation	74
5.4.2	Tadpole loop integrals	75
5.4.2.1	Tadpole integration on simulator	76
5.4.2.2	Tadpole integration on hardware	77
5.4.3	Bubble loop integral	78
5.4.3.1	Bubble tensor loop integral	80
5.4.4	Triangle loop integral	80
5.4.4.1	Triangle tensor loop integral	81
5.4.5	Pentagon loop integral	82
5.5	Quantum decay rates at second order in perturbation theory	84
5.5.1	Causal unitary and decay rates at NLO	85
5.5.2	Quantum integration of NLO decay rates	87
5.6	Conclusions	90
6	Quantum generative models for Fragmentation Functions	92
6.1	Introduction	92
6.2	Fragmentation Functions	93
6.3	Quantum Chebyshev Probabilistic Model	95
6.4	Quantum Chebyshev Probabilistic Models for Fragmentation Functions	99

6.4.1	Model	99
6.4.2	Results	102
6.5	Conclusions	105
7	Outlook and final remarks	107
8	Resumen de la tesis	110
A	Controlled <i>SwapTest</i>	127
B	Training and sampling of fragmentation functions of kaon and pion	129
C	Correlations in Quantum Chebyshev Probabilistic Models	132
C.1	Von Neumann entropy in quantum registers	132
C.2	Purity in quantum registers	133
C.3	Mutual information in quantum registers	134
	List of Figures	136
	List of Tables	142
	Bibliography	143

1 Introduction

1.1 Historical introduction

At the beginning of the 20th century, a series of groundbreaking discoveries made evident something that until that moment was not questionable. Classical theories could not explain certain phenomena and a better theory was needed.

The first problem that intrigued the physics community was the blackbody radiation problem. Blackbody is a term used to describe objects that absorb all incident electromagnetic radiation and emit radiation across a continuous spectrum that depends only on its temperature. Classical physics failed to explain the radiation that blackbody would emit, predicting infinite energy at high frequencies, a contradiction known as the ultraviolet catastrophe. In 1900, Max Planck proposed that energy is emitted in discrete packets (quanta), leading to a formula that accurately described the observed radiation spectrum and resolved the inconsistency [10].

Inspired by Planck's work, in 1905 Albert Einstein proposed his solution to explain the photoelectric effect. This effect consists of the emission of electrons from a material caused by electromagnetic radiation such as ultraviolet light. Classical electromagnetism predicted that light waves transfer energy to electrons continuously, causing emission once enough energy accumulates. However, experiments showed that electrons are emitted only when light exceeds a specific frequency, independent of intensity or exposure time. To explain this, Einstein proposed that light consists of discrete energy packets, or photons, rather than continuous waves, which awarded him the Nobel Prize in 1911.

Years later, in the decade of 1920, a generation of very talented physicists laid the foundation of Quantum Mechanics (QM). In 1924, Louis de Broglie proposed that matter, like light, could present wave-like behavior, introducing the concept of wave-particle duality that would be of vital importance for later discoveries. Then, in 1925, Werner Heisenberg introduced the first mathematical formalism of QM, the Matrix Mechanics, describing particles through probabilities rather than the deterministic laws of classical mechanics. Shortly afterward, Erwin Schrödinger developed Wave Mechanics, describing the quantum behavior in his well-known equation, which describes how particles evolve as waves, a cornerstone of quantum theory.

In 1927, Heisenberg transformed the field again with his Uncertainty Principle [11], which states that one cannot simultaneously determine a particle's position and momentum with complete accuracy. This principle showed a fundamental limit in measurement at the quantum level, revealing a probabilistic nature to reality itself. Around the same time, Niels Bohr and his colleagues developed the Copenhagen interpretation, which suggested that the act of measurement affects quantum systems. Meaning that a quantum system can be in a superposition of different quantum states and the measurement itself forces

nature to choose one of the multiple states. This interpretation highlighted the inherent uncertainty in QM, questioning the classical concept of an objective reality.

As QM continued to develop, it had to be reconciled with the theory of Special Relativity, by Albert Einstein [12]. Eventually, this would lead to the formulation of Quantum Field Theory (QFT) in the 1930s. In particular, Paul Dirac played a critical role by developing the relativistic theory of the electron, predicting antimatter and laying the fundamentals for QFT. By the mid-20th century, quantum electrodynamics (QED) emerged as the first complete QFT, describing interactions between electrons and photons with extraordinary precision. Over the next few decades, QFT evolved into the Standard Model (SM) of particle physics, describing all known fundamental particles and their interactions through the strong, weak, and electromagnetic forces. This framework allowed physicists to make precise predictions for high-energy experiments, establishing QFT as the foundation of modern particle physics and one of the theories capable of making the most accurate results ever seen.

In the meantime, while quantum theory was revolutionizing the physics world, information science was reshaping our world with the advent of the invention of the century, *the machine that changed the world* [13], the computer. In 1936, Alan Turing and Alonzo Church working independently and together, described what an algorithm is and defined the limits of what can be computed. This led to the Church–Turing thesis [14], which states that any calculation that can be performed can be solved by an algorithm on a computer, as long as there is enough time and memory. That same year, Turing also introduced the concept of Turing Machines [15], which are theoretical devices that describe how digital computers work. These machines laid the groundwork for modern computers and introduced the idea of storing programs, which most computers use today. Turing Machines were designed to explore what can and cannot be computed, taking into account the limitations of computational power.

Since the invention of such machines, scientists of across different fields started using them to simulate the natural world. In light of the new discoveries, Richard Feynman in the early 1980s was among the first ones to ponder the idea of simulating physical systems with a system that also follows the same rules, a quantum system. In fact, their words in “Simulating Physics with Computers” [16] have now become a must-have for papers and seminar talks in the field of quantum computation and quantum information, they read “nature isn’t classical, dammit, and if you want to make a simulation of nature, you’d better make it quantum mechanical, and by golly it’s a wonderful problem, because it doesn’t look so easy”.

This idea of simulating nature with quantum systems led to the development of Quantum Computing (QC). Unlike classical computers, which process information in bits representing either 0 or 1, quantum computers use quantum bits, or qubits, represented by quantum states that can exist in a superposition of both $|0\rangle$ and $|1\rangle$ states. Quantum operations are carried out with quantum gates, fundamental units of quantum circuits that manipulate qubits through carefully controlled operations. These quantum gates enable the use of three main properties of QM that are essential for quantum computation: entanglement, where the state of one qubit directly influences the state of another; superposition, where a qubit can be in a combination of different states simultaneously; and interference, the process by which probabilities of qubit states amplify or cancel each other out.

From a theoretical point of view, the field of QC has made significant progress in understanding quantum algorithms and their potential applications. However, designing

these algorithms poses unique challenges. In the notoriously famous book “Quantum Computing and Quantum Information” [17], Nielsen and Chuang highlight two key reasons for this. First, humans naturally have a better intuition for classical computing, making it hard to come up with algorithms well-suited for quantum computers. Second, for a quantum algorithm to be truly valuable, it must outperform classical alternatives by performing the same computations with fewer resources. Some classical problems are theoretically solvable but practically unattainable due to their exponential complexity. The challenge lies in whether efficient quantum algorithms can perform these intractable computations in significantly less time, making them practically solvable. Despite the difficulties, in 1994 Peter Shor introduced a groundbreaking algorithm that efficiently performs prime factorization on quantum computers, with an exponential advantage in time. Soon after in 1996, Lov Grover developed an algorithm that provides a quadratic speedup for searching in unstructured databases. Recently, Variational Quantum Circuits (VQC) have emerged as a promising method for developing new quantum algorithms. These quantum circuits utilize parameterized quantum gates optimized by classical techniques, enabling researchers to tackle complex problems across various fields, including quantum chemistry and machine learning. Quantum Machine Learning (QML), in particular, aims to harness quantum computing’s unique properties to enhance data processing and pattern recognition. While these algorithms showcase the capabilities of quantum computing, the range of efficient quantum solutions remains limited. Researchers are actively working to discover more efficient quantum algorithms, requiring creativity due to the absence of a clear roadmap in this emerging and exciting field.

On the other hand, from an experimental point of view, different architectures for quantum computers have been proposed, each leveraging unique technologies to manipulate qubits. Superconducting qubits, used by companies like IBM and Google, rely on circuits cooled to near absolute zero to maintain the coherence in the quantum states. Ion-trap quantum computers, developed by companies such as IonQ and Quantinuum, use electromagnetic fields to trap individual ions and perform quantum operations. Other architectures include photonic quantum computers, pursued by Xanadu and PsiQuantum, which use photons as qubits, and neutral-atom quantum computers, developed by companies like Pasqal and Atom Computing, which trap neutral atoms using lasers. Each approach has its own advantages and challenges, with ongoing development aiming to find the most scalable and stable design for practical QC. It has become evident that a race is taking place in the industry to determine which company can first develop a reliable quantum computer that achieves the long-awaited quantum advantage, potentially transforming the landscape of computational technology as we know it today.

1.2 Motivation

Despite the proliferation of algorithms and new methods in the field of QC, many of these endeavors have fallen short of addressing real-world challenges. Bridging this gap is essential, as tackling real-world problems offers invaluable insights into the limitations of current algorithms and methods. Engaging with such challenges provides a deeper understanding of existing knowledge gaps, facilitating the development of innovative algorithms and solutions. A clear illustration of this phenomenon can be found in the history of the European Council for Nuclear Research (CERN), where the invention of the World Wide Web (WWW) emerged as a solution to the practical challenge of sharing information among scientists across the globe. This historical precedent underscores the importance of addressing problems that pose technical challenges to state-of-the-art classical computers in driving innovation and advancing the field of QC.

Going back to Richard Feynman’s quote, the main niche for QC lies in simulating quantum systems. In this direction, several areas have found important applications. For instance, quantum chemistry has seen success with algorithms like the Variational Quantum Eigensolver (VQE) [18] which computes the ground state of molecules, a daunting task for classical computers when dealing with large molecules. Another area of great interest is QFT. One of its greatest achievements is the SM of particle physics, one of the most conceptually rich frameworks in physics that provides the most accurate predictions of any scientific theory to date. However, it still presents numerous challenges. High-energy physics collisions, such as those at the Large Hadron Collider (LHC) at CERN, are described by QFT, and represent some of the most complex problems to analyze. Furthermore, inconsistencies observed in their analysis suggest that there may be new physics yet to be discovered or that the SM may need to incorporate new elements to address these new phenomena.

With these considerations in mind, this thesis identifies a promising area of application that combines both concepts. On the one hand, QFT is inherently quantum, making it well-suited for study through quantum systems. On the other hand, it encapsulates some of the most challenging problems in theoretical and computational science. Addressing these challenges with QC tools could yield to new insights into the fundamental particles that constitute the universe and also to advancements for QC itself. This thesis, therefore, aims to deepen our understanding of the universe by inspiring advancements in QC that could reshape modern computational paradigms.

2 Standard Model

2.1 The Standard Model of particle physics in a nutshell

The Standard Model (SM) stands as the most widely accepted theory in particle physics. It consists of a Quantum Field Theory (QFT) that merges Quantum Mechanics and Special Relativity into a consistent field theory framework, enabling the classification of all known elementary particles and describing the fundamental interactions among themselves. Specifically, the SM arises from a combination of gauge theories that satisfy Noether's theorem¹, such as Quantum Electrodynamics (QED), the Weak theory by Glashow, Weinberg, and Salam, and Quantum Chromodynamics (QCD).

As shown in Fig. 2.1, the SM particles can be divided into those constituting matter (fermions) and those mediating interactions (bosons).

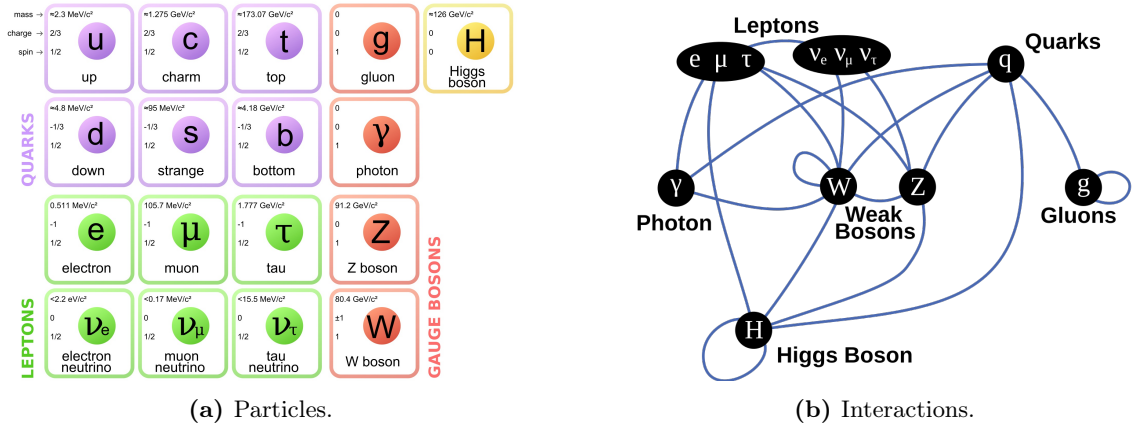


Figure 2.1: (a) Classification of the particles in the SM. Picture taken from [19]. (b) Interactions among particles in the SM. Picture taken from [20].

The matter particles, or fermions, have half-integer spin and follow the Pauli Exclusion Principle, which states that two identical fermions cannot simultaneously occupy the same quantum state. These particles are named after the Fermi-Dirac statistics that govern their behavior. Each fermion has an associated antiparticle with the same mass but opposite electric charge. Fermions are further classified into quarks and leptons:

- **Leptons:** Elementary particles that can exist freely or bound, do not interact via the strong force, and have nonzero mass. There are three lepton families (or flavors): the electron e , muon μ , and tau τ . Each family consists of a pair of particles with opposite lepton numbers. One is the charged, massive particle that gives the family

¹Noether's Theorem: If a Lagrangian $\mathcal{L}$ is invariant under a continuous symmetry, then a corresponding conservation law exists.

its name, such as the electron, and the other is a neutral, nearly massless particle known as the neutrino.

- **Quarks:** Elementary particles that interact via the strong force and are never found free in nature, instead existing only within hadrons, composite particles made up of quarks, such as protons and neutrons. There are six types of quarks, arranged with leptons into three generations, each ordered by increasing discovery date and mass. The first generation includes the up (u) and down (d) quarks, which constitute ordinary matter (protons and neutrons). The second generation contains the charm (c) and strange (s) quarks, which appear in cosmic radiation or decay processes. Finally, the third generation consists of the top (t) and bottom (b) quarks, observed only under extreme conditions, such as those after the Big Bang or in particle accelerators such as the Large Hadron Collider (LHC) at CERN.

The SM describes the fundamental forces as a result of the exchange (emission and absorption) of mediating particles known as bosons. These particles are named after the Bose-Einstein statistics that describe their behavior. In contrast to fermions, bosons have integer spin and are not restricted by the Pauli Exclusion Principle. The different bosons include:

- **Photons (γ):** Massless particles mediating the electromagnetic interaction between electrically charged particles.
- **Bosons W^+ , W^- , Z :** Massive particles mediating the weak interaction. Together with photons, they are the mediators of the electroweak interaction.
- **Gluons (g):** Massless particles that mediate the strong interaction among particles with color charge, such as quarks and other gluons.

In addition to these bosons, the Higgs boson, a massive, spin-zero particle, plays a unique role by providing particles with mass through the Higgs mechanism.

The SM is an extraordinarily complex and elegant theory, where concepts like symmetry play a fundamental role in governing particle dynamics. Its predictive power is remarkable, successfully anticipating major discoveries [21], such as: the second generation of quarks in the 1970s [22], the third generation of leptons in the 1980s [23], the $W^\pm$ and Z bosons [24, 25], the third generation of quarks in the 1990s [26], and ultimately the Higgs boson in 2012 [27, 28]. Not only has the SM predicted previously unknown particles, but it has also achieved unprecedented agreement between theoretical predictions and experimental data, such as the electron’s anomalous magnetic moment and the masses of the $W^\pm$ and Z bosons, establishing a benchmark for predictive accuracy in physics. For a more detailed review of the SM, see [29].

Yet, despite these considerable successes, the SM cannot explain a range of phenomena, preventing it from being a “theory of everything.” Some of its limitations include [30]:

- **Neutrinos:** The SM is not able to explain why neutrinos are massive. However, experimental evidence confirms that at least two neutrinos possess nonzero mass. Additionally, the SM does not account for the experimentally observed neutrino oscillations.
- **Matter-antimatter asymmetry:** Following the Big Bang, a slight asymmetry between matter and antimatter led to a universe composed predominantly of matter. However, the SM cannot fully explain the magnitude of this asymmetry via baryogenesis.

- **Dark Matter:** Observational cosmology provides indirect evidence of a type of matter that neither emits nor interacts with light, known as dark matter, which the SM does not predict.
- **Gravity:** While the SM unifies three of the four fundamental forces (electromagnetism, weak, and strong forces), it does not incorporate general relativity, leaving the development of a quantum theory of gravity an open challenge.

In addition to these questions, the SM raises philosophical ones as well, as certain aspects seem arbitrary: Why are there only three fermion generations? Why is electric charge quantized in multiples of $1/3$? Why does the SM contain 19 free parameters?

Analyzing both the strengths and limitations of the SM reveals it to be a highly successful and detailed theory with incredible predictive power. However, as it still cannot account for all observed phenomena, the SM motivates continued research in particle physics, encouraging us to seek answers to the many questions that remain open.

2.2 A Lagrangian to rule them all

The Lagrangian is a mathematical function that is used to describe the dynamics of a physical system. It contains the kinetic and potential energy of a system, allowing us to derive the equations of motion through the Principle of Least Action. In QFT, the Lagrangian provides a compact framework to express the fundamental interactions and symmetries of elementary particles. For the SM, the Lagrangian encodes the behavior of fermions, gauge bosons, and the Higgs field, as well as the interactions that define the electroweak and strong forces.

The Lagrangian of the SM is built based on fundamental symmetries. These symmetries arise from one of the most fundamental principles of physics: the laws of physics must be independent of the position and speed of a given observer. This translates to the laws of physics must be invariant under Lorentz transformations. As a QFT, the SM uses fields to describe particles, but these fields are not physical observables that can be directly measured. Thus, different field configurations with no physical relevance must produce identical measurable outcomes. The transformations that map one of these configurations to another are known as gauge transformations.

The symmetry group of the SM is defined by the gauge symmetries of the electroweak and strong interactions:

$$G_{\text{SM}} = SU(3)_C \otimes SU(2)_L \otimes U(1)_Y, \quad (2.1)$$

where the subgroup $SU(3)_C$ corresponds to the strong interactions, whereas the $SU(2)_L \otimes U(1)_Y$ relates to the electroweak interactions. This connection arises because the mediators (bosons) of the electroweak and strong interactions are generated by the groups $SU(2)_L \otimes U(1)_Y$ and $SU(3)_C$ respectively.

2.2.1 Quantum Chromodynamics

Quantum Chromodynamics (QCD) is a QFT built to describe the strong interaction between quarks, the fundamental components of hadrons. The discovery of these elementary particles that conform matter marked a significant shift in our understanding of particle physics. Their presence inside the hadrons required the introduction of a new quantum number to satisfy the Fermi-Dirac statistics. This quantum number, the color charge, requires that each quark can have $N_C = 3$ different colors. Although this charge

needs to be introduced to ensure that Fermi-Dirac statistics is not violated, colored states are not observed in nature. This led to the formulation of the confinement hypothesis, by which all asymptotic states must be singlets in color space. The main consequence of confinement is that quarks cannot be observed as free particles except at high energies. Instead, they can only be found in color-neutral bound states.

The force that binds the colored quarks within hadronic states is the strong interaction, which is much stronger at long distances than at short distances, unlike other forces such as the electromagnetic force or gravity. This explains how quarks with the same electric charge remain bound together inside a hadron, rather than repelling each other. The QCD theory is built by applying the $SU(3)_C$ gauge symmetry. This requires the introduction of $N_C^2 - 1 = 8$ new massless boson fields G_a^μ called gluons. With these considerations, the Lagrangian of the QCD theory reads:

$$\begin{aligned} \mathcal{L}_{\text{QCD}} = & -\frac{1}{4} \left(\partial^\mu G_\nu^a - \partial^\nu G_\mu^a \right) \left(\partial_\mu G_\nu^a - \partial_\nu G_\mu^a \right) + \sum_f \bar{q}_f \left(i \gamma^\mu \partial_\mu - m_f \right) q_f \\ & - g_s G_a^\mu \sum_f \bar{q}_f \gamma_\mu \left(\frac{\lambda^a}{2} \right) q_f + \frac{g_s}{2} f^{abc} \left(\partial^\mu G_\nu^a - \partial^\nu G_\mu^a \right) G_\mu^b G_\nu^c \\ & - \frac{g_s^2}{4} f^{abc} f^{ade} G_b^\mu G_c^\nu G_\mu^d G_\nu^e, \end{aligned} \quad (2.2)$$

where ∂^μ represents a covariant derivative, $q_f, \bar{q}_f$ represent the fields of a quark and an anti-quark, respectively, γ_μ are the Dirac matrices, m_f is the mass of the quark, g_s is the strong coupling, λ^a are the generators of the $SU(3)$ group and f^{abc} are the structure constants of $SU(3)$.

The self-interactions, encoded in the cubic and quartic terms of the gluon fields, reveal two key features of QCD, asymptotic freedom and confinement, as they determine the sign of the β function that controls the energy evolution of the strong coupling. To understand this concept it is convenient to take a look at the behavior of the strong coupling

$$g_s(k^2) \sim \frac{1}{\log(k^2/\Lambda^2)}, \quad (2.3)$$

where k is the energy scale of the interaction and Λ is a fundamental parameter in QCD that defines the energy scale at which the strong force becomes non-perturbative. According to Eq. (2.3), the strong coupling decreases as the energy k increases. On the contrary, for k close to Λ the strong coupling raises leading to the confinement of quarks and gluons.

2.2.2 Electroweak interactions

The Electroweak (EW) sector of the SM unifies the electromagnetic and weak interactions under a single theoretical framework. This unification is achieved by introducing a gauge theory based on the symmetry group $SU(2)_L \otimes U(1)_Y$. Unlike QCD, which acts only on quarks, the EW interaction involves both quarks and leptons.

The $SU(2)_L$ component corresponds to the weak interactions mediated by the $W^\pm$ and Z bosons, while $U(1)_Y$ describes hypercharge, which is related to the electromagnetic interaction, mediated by photons.

However, in order to define a unified EW theory it makes more sense to talk about four different mediator bosons. Two of these, W_μ^1 and W_μ^2 , arise from the $SU(2)_L$ symmetry and are responsible for the weak charged current interactions. These bosons correspond

to the $W^\pm$ particles. The other two bosons, W_μ^3 and B_μ , are associated with the neutral current interactions and are related to the Z boson and the photon through the mixing mechanism and the mixing angle θ_W :

$$\begin{pmatrix} W_\mu^3 \\ B_\mu \end{pmatrix} = \begin{pmatrix} \cos \theta_W & \sin \theta_W \\ -\sin \theta_W & \cos \theta_W \end{pmatrix} \begin{pmatrix} Z_\mu \\ A_\mu \end{pmatrix}, \quad (2.4)$$

where A_μ and Z_μ represent the fields of the photon and Z boson, respectively.

The Lagrangian for the EW interaction is expressed as:

$$\mathcal{L}_{\text{EW}} = -\frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{4}W_{\mu\nu}^i W^{i\mu\nu} + \mathcal{L}_{\text{CC}} + \mathcal{L}_{\text{NC}}. \quad (2.5)$$

where the first two terms correspond to the kinetic terms for the new gauge fields:

$$B_{\mu\nu} \equiv \partial_\mu B_\nu - \partial_\nu B_\mu, \quad W_{\mu\nu}^i \equiv \partial_\mu W_\nu^i - \partial_\nu W_\mu^i - g\epsilon^{ijk}W_\mu^j W_\nu^k.$$

The $W_{\mu\nu}^i$ field introduces self-interactions among the gauge fields, whose intensity is quantified by the g coupling.

Regarding the other two terms, $\mathcal{L}_{\text{CC}}$ and $\mathcal{L}_{\text{NC}}$ are the charged and neutral current Lagrangians

$$\mathcal{L}_{\text{CC}} = -\frac{g}{2\sqrt{2}} \left(W_\mu^\dagger \left[\sum_{i,j} \bar{u}_i \gamma^\mu (1 - \gamma_5) V_{ij} d_j + \sum_l \bar{\nu}_l \gamma^\mu (1 - \gamma_5) l \right] + \text{h.c.} \right), \quad (2.6)$$

$$\mathcal{L}_{\text{NC}} = \mathcal{L}_{\text{QED}} + \mathcal{L}_{\text{NC}}^Z = -eA_\mu \sum_f \bar{f} \gamma^\mu Q_f f - \frac{e}{2 \sin \theta_W \cos \theta_W} Z_\mu \sum_f \bar{f} \gamma^\mu (v_f - a_f \gamma^5) f. \quad (2.7)$$

In the charged-current Lagrangian, g is the $SU(2)$ gauge coupling, $\bar{u}_i$ and d_j denote the different anti-up and down quarks for the different families, $\bar{\nu}_l$ and l represent the antineutrino and lepton for the different families, $\gamma_5 = i\gamma^0\gamma^1\gamma^2\gamma^3$ is the product of four Dirac matrices and V_{ij} is the Cabibbo-Kobayashi-Maskawa matrix that describes the flavor-changing charged currents. In the neutral-current Lagrangian we have separated the Quantum Electrodynamics (QED) Lagrangian $\mathcal{L}_{\text{QED}}$ from the weak interactions in $\mathcal{L}_{\text{NC}}^Z$. In Eq. (2.7), e represents the electric charge of the electron, $\bar{f}$ and f are the fermionic fields, Q is the fermion electromagnetic charge, and $a_f = T_3^f$ and $v_f = T_3^f (1 - 4|Q_f| \sin^2 \theta_W)$, with T_3^f being the weak isospin.

At this point, using gauge symmetry principles we have built a Lagrangian that encodes the dynamics to describe the electromagnetic and weak interactions. Nevertheless, the gauge symmetry employed to construct this theory forbids any mass terms in the EW Lagrangian. Yet, we have experimental proof that both the fermions and the Z and $W^\pm$ bosons are massive, so there has to be a missing piece in our theory. In other words, we need to find a mechanism that gives masses to the particles in the SM.

2.2.3 Electroweak Spontaneous Symmetry Breaking

In order to generate masses in the SM the EW gauge symmetry has to be broken. However, we also need a fully symmetric Lagrangian to ensure renormalizability. This raises an interesting question: how can we reconcile these seemingly contradictory requirements?

To answer this question the Brout-Englert-Higgs mechanism was proposed [31, 32]. The main idea is to introduce a new particle into the SM which is invariant under transformations of the symmetry group G_{SM} , but whose vacuum state is not. Then, we will have a Lagrangian with a degenerate set of states at the minimum energy level. However, only one of those states can be the real vacuum state, and when it is chosen the symmetry is said to be broken. This process, known as Spontaneous Symmetry Breaking (SSB), leads to the emergence of new degrees of freedom in the form of massless bosons, referred to as Goldstone bosons. An interpretation of these bosons is presented as the Goldstone theorem [33], in which they are excitations of the field around the vacuum that do not increase the energy of the system.

This SSB mechanism serves the purpose of giving mass to the $W^\pm$ and Z bosons while leaving the photon massless. Mathematically, let us start by introducing two additional complex scalar fields which transform under $SU(2)_L$ as a doublet:

$$\phi(x) \equiv \begin{pmatrix} \phi^{(+)}(x) \\ \phi^{(0)}(x) \end{pmatrix}, \quad (2.8)$$

and a Lagrangian that describes these fields

$$\mathcal{L}_S = (D_\mu \phi)^\dagger D^\mu \phi - \mu^2 \phi^\dagger \phi - \frac{\lambda}{2} (\phi^\dagger \phi)^2, \quad (2.9)$$

where $\lambda > 0$ and $\mu^2 < 0$ to ensure that there exists an infinite set of degenerate states that minimize the potential. The covariant derivative is given by

$$D^\mu \phi = \left[\partial^\mu + igW^\mu(x) + ig \frac{\tan(\theta_W)}{2} B^\mu(x) \right] \phi. \quad (2.10)$$

Now, the vacuum state has to be chosen among all the infinite degenerate states with minimal energy. This choice of the vacuum state will provoke the SSB of the EW gauge group into the electromagnetic one

$$SU(2)_L \otimes U(1)_Y \xrightarrow{\text{SSB}} U(1)_{\text{QED}}. \quad (2.11)$$

When SSB takes place, three of the four fields in the scalar doublet can be reinterpreted as Goldstone bosons, which are absorbed by the $W^\pm$ and Z bosons to provide their longitudinal components. This mechanism explains how the weak mediators acquire mass. The remaining scalar field becomes a fluctuation H around the vacuum value v , identified as the EW vacuum expectation value (vev). This introduces a new scalar particle in the SM, known as the Higgs boson.

With these considerations, the scalar Lagrangian becomes:

$$\mathcal{L}_S = \frac{1}{2} \partial_\mu H \partial^\mu H + \left[\frac{g^2}{4} W_\mu^\dagger W^\mu + \frac{g^2}{8 \cos^2 \theta_W} Z_\mu Z^\mu \right] (v + H)^2 - \frac{1}{2} M_H^2 H^2 + \frac{M_H^2}{2v} H^3 - \frac{M_H^2}{8v^2} H^4, \quad (2.12)$$

where $M_H = \sqrt{\lambda v}$. From Eq. (2.12) one can see that the weak bosons acquire masses as they present quadratic terms. In particular, the coefficients associated with these terms represent the boson masses, which are defined as:

$$M_W = M_Z \cos \theta_W = \frac{1}{2} vg. \quad (2.13)$$

Having successfully generated masses for the weak bosons, the masses of the fermions are still missing. How can this be addressed?

2.2.4 Yukawa sector

The procedure to include the fermion masses into the SM Lagrangian consists of reutilizing the same scalar doublet of Eq. (2.8). Now, taking into account the symmetry requirements of the SM after SSB, we can write the following gauge-invariant fermion-scalar couplings, known as the Yukawa sector:

$$\mathcal{L}_Y = - \left(1 + \frac{H}{v} \right) \sum_f (m_u \bar{u}_f u_f + m_d \bar{d}_f d_f + m_l \bar{l}_f l_f) \quad (2.14)$$

and thus provides the SM with mass terms for all fermions besides the neutrinos. Now, we can gather all the pieces from Eq. (2.2), Eq. (2.5), Eq. (2.12) and Eq. (2.14) to build the full Lagrangian of the SM:

$$\mathcal{L}_{\text{SM}} = \mathcal{L}_{\text{QCD}} + \mathcal{L}_{\text{EW}} + \mathcal{L}_S + \mathcal{L}_Y. \quad (2.15)$$

With this, we have successfully derived the expression of the Lagrangian of the SM. A Lagrangian that unifies the strong, weak, and electromagnetic forces within a single elegant framework, making it one of the most remarkable achievements in modern physics. A Lagrangian that contains the dynamics of all the elementary particles known up to date. *A Lagrangian to rule them all.*

2.3 Precision Physics in the Standard Model

The SM has been so far very successful in providing very accurate predictions of the behavior of the particles that conform the universe. A new generation of particle physics colliders experiments [34, 35, 36, 37, 38, 39, 40, 41] is designed to unravel in greater detail the mysteries of the universe. Current and upcoming experiments, such as those conducted at the Large Hadron Collider (LHC) at CERN, are generating unprecedented volumes of data. Just to give a rough idea, when the LHC is active it produces around 1 billion collisions per second. The CERN Data Center² is storing more than 30 petabytes of data per year from the LHC experiments, enough to fill about 1.2 million Blu-ray discs, i.e. 250 years of HD video [42]. This has increased the pressure on the theoretical physics community to produce more precise predictions, enabling accurate comparisons with experimental results to ensure no subtle details are missed. If new physics exists beyond the SM, these efforts aim to discover it.

Given the enormous computational challenges of analyzing such vast datasets, advanced and sophisticated methods are essential. Over the past decades, machine learning (ML) techniques have been extensively applied to address these challenges [43], demonstrating their potential in tasks such as data classification, event reconstruction, and particle identification. ML has been adopted in many analyses of large experimental collaborations such as LHCb, CMS, and ATLAS at the LHC, where it helps in improving the efficiency and precision of the mentioned tasks.

While ML has proven useful in handling a wide range of problems, it also presents its own drawbacks. First, its success depends heavily on the availability of high-quality training data. Second, although their efficiencies often outperform those of the classical methods, they work as black boxes providing little information about their functioning and when

²As part of my PhD journey I carried out a research stay at the CERN Quantum Technology Initiative (CERN-QTI) group and I was sitting in my office every day in the Data Center, just above all this vast amount of data.

doing fundamental science, this might not be what one wants. Third, their accurate predictions rely on the necessity of significant computational resources, which might limit their scalability. On top of the mentioned limitations, despite its widespread use, ML remains a classical computational approach that may struggle with capturing the intrinsic “quantumness” of quantum mechanical processes.

In this context, the advent of Quantum Computing (QC) in the past years has opened new possibilities by directly leveraging the principles of quantum mechanics. Quantum algorithms, executed on quantum computers, have the potential to provide computational advantages over problems that are intractable for classical methods. Equally important, they may provide unique insights into the behavior of elementary particles by analyzing quantum phenomena through the lens of QC. Since the SM is a QFT, a theory intrinsically quantum in nature, QC arises as the natural choice of tool to analyze their properties and intricacies. This synergy offers a promising avenue of research for exploring the fundamental building blocks of nature at the microscopic scale.

3 Quantum Computing

3.1 Classical and quantum bits

The *bit* is the fundamental unit of classical computation and information. It represents the smallest possible piece of information, taking on one of two binary values: 0 or 1. In physical implementations, these binary states are typically represented by the presence (1) or absence (0) of electrical current flowing through a circuit.

The importance of classical bits lies in their simplicity and reliability. They enable the storage and processing of vast amounts of information in a deterministic manner, making them the cornerstone of modern digital technology. Classical bits are used in all digital devices, from the smartphones in our pockets to the supercomputers in data centers, and have been crucial in making technological breakthroughs across different fields.

However, classical bits face several limitations. One significant constraint is their binary nature, which restricts them to representing only one state at a time. This limitation appears when dealing with complex computational problems that require exponential resources as the problem size increases. Another limitation is related to Moore's law, which states that the number of transistors on an integrated circuit doubles approximately every two years while their cost decreases. Although this has enabled technological advancement for decades, we are inevitably approaching a physical limit where further miniaturization of transistors approaches atomic scales, leading to the emergence of quantum mechanical effects. At these dimensions, phenomena such as electron tunneling and the Heisenberg Uncertainty Principle introduce unpredictability in transistor behavior. Additionally, classical bits are unable to accurately simulate certain natural phenomena, particularly those governed by Quantum Mechanics (QM). This limitation restricts our ability to model and understand complex systems in fields such as chemistry, materials science, and particle physics, where quantum effects play a crucial role.

On the other hand, quantum bits, or *qubits*, offer a new approach to addressing some of the inherent limitations of classical bits. They differ to classical bits in the following aspects [44]:

- **Superposition:** Unlike classical bits, qubits can exist in a superposition of states. This means that a qubit can be in a state $|0\rangle$ or $|1\rangle$, or any linear combination of these states:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (3.1)$$

with $\alpha^2 + \beta^2 = 1$. The state $|\psi\rangle$ of Eq. (3.1) can also be expressed as a vector in a sphere of radius 1 whose position is defined by two angles θ and ϕ :

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle. \quad (3.2)$$

This way of defining the superposition state of $|\psi\rangle$ is called the Bloch sphere and can be visualized as in Fig. 3.1

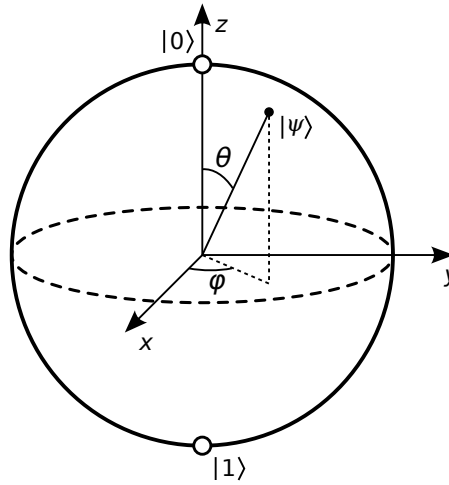


Figure 3.1: Bloch sphere representation of a qubit, showing quantum states as points on a unit sphere. Picture taken from [45].

Although a single qubit in superposition has limited utility on its own, groups of qubits in superposition can generate complex, multidimensional computational spaces. These spaces enable new representations and solutions for complex problems.

- **Entanglement:** Qubits can be entangled, a phenomenon where the state of one qubit is completely dependent on the state of another. This property is independent of the distance between them and was difficult to understand by scientists in the early days of QM. Albert Einstein referred to this term as “spooky action at distance”. From a utility point of view, entanglement can create correlations between qubits that can be exploited to perform complex computations more efficiently than classical systems when combined with other properties such as superposition. For example, in a system of two entangled qubits, the measurement of one qubit immediately determines the state of the other.
- **Decoherence:** Decoherence is the process in which quantum systems lose their “quantumness” by collapsing into single states measurable by classical physics. This phenomenon is provoked by interactions with the environment and is one of the key characteristics of qubits. The decoherence time of qubits marks how long they can preserve their quantum properties, hence determining the duration of the quantum algorithms they can execute before decaying into classical states.
- **Interference:** Interference is the phenomenon in which entangled quantum states interact and amplifies certain probabilities while diminishing others. This feature can be leveraged to build algorithms to amplify correct solutions and cancel out incorrect ones. Shor’s and Grover’s algorithms are examples that leverage quantum interference to achieve a quantum speedup with respect to their classical counterparts.
- **Measurement:** The measurement process in quantum computing is inherently probabilistic. When a qubit is measured, it collapses from its superposition state into one of its basis states, $|0\rangle$ or $|1\rangle$, with probabilities defined by the squared magnitudes of the coefficients α and β of Eq. (3.1). This contrasts with the deterministic nature

of classical bit measurements, presenting unique challenges while also offering new opportunities in the design of quantum algorithms.

From an architectural point of view, bits and qubits also present substantial differences. Historically, bits have been represented through various physical mediums, evolving with technology. Early methods included mechanical systems like punched cards, where bits were represented as holes (1) or no holes (0) in a physical card. Later, magnetic storage used the orientation of magnetized regions and transistors, which form today's computers, enabled bits to be encoded as electrical signals. On the other hand, the architectures for qubits are much more diverse and there is no clear hegemony from one over the others. All of them present their own advantages and challenges. Some of the most used technologies to build qubits include:

- **Superconducting qubits:** This technology leverages superconducting circuits operating at very low temperatures. These qubits present high-speed computation and relatively straightforward fine-tuned control. Major tech companies such as Google, IBM¹, and Intel are building quantum processors based on this architecture.
- **Trapped ion qubits:** They utilize charged atomic particles confined by electromagnetic fields which have long coherence times and high-fidelity measurements. Several companies are developing quantum computers based on this technology, including Quantinuum and IonQ.
- **Neutral atoms:** Laser-cooled neutral atoms are used as qubits, providing scalability and long coherence times. Companies like PASQAL and QuEra are pioneering this technology.
- **Photons:** Photonic quantum computing uses individual photons as qubits, offering advantages like room-temperature operation and compatibility with existing optical infrastructure. Companies developing photonic quantum computers include Xanadu¹ and PsiQuantum.

3.2 Classical and quantum gates

Another major difference between bits and qubits is the kind of operations we use to construct algorithms, or in other words, the logic gates we apply. In classical computing, every algorithm is decomposed into its fundamental building blocks, the logical gates. From a simple arithmetic operation $0 + 1$ to the modern and complex transformers that form the Large Language Models (LLMs), such as ChatGPT, all are essentially based on logic gates.

Logic gates are designed using electronic components such as transistors, diodes, and resistors. These gates perform logical operations based on their inputs, producing outputs that are either 0 or 1. Their functionality is based in Boolean algebra, a branch of mathematics that deals with logical operations on binary variables [46].

We can classify logic gates into three main categories:

- **AND:** An AND gate is used to perform logical multiplication of binary input. The output of the AND gate will be 1 if both inputs are 1, and 0 otherwise.

¹During my PhD, I had the opportunity to do two amazing internships at IBM Research in Zurich and Xanadu in Toronto, where I gained firsthand experience of what it is like to work at and contribute to companies that are leading the development of useful quantum computers.

- **NOT:** The NOT gate is also known as inverter. It outputs a 1 when the input is 0 and vice versa.
- **OR:** The output of OR gate will be 1 if any of the input states is 1, and 0 otherwise.

Then, a combination or negation of these three can build the other 4 logic gates: NOR, NAND, XOR, Buffer and XNOR. The Fig. 3.2 show the different logic gates and their truth tables.

Buffer



Input	Output
0	0
1	1

Inverter



Input	Output
0	1
1	0

AND



A	B	Output
0	0	0
1	0	0
0	1	0
1	1	1

NAND



A	B	Output
0	0	1
1	0	1
0	1	1
1	1	0

OR



A	B	Output
0	0	0
1	0	1
0	1	1
1	1	1

NOR



A	B	Output
0	0	1
1	0	0
0	1	0
1	1	0

XOR



A	B	Output
0	0	0
1	0	1
0	1	1
1	1	0

XNOR



A	B	Output
0	0	1
1	0	0
0	1	0
1	1	1

Figure 3.2: Truth tables for fundamental logic gates: AND, OR, NOT, NAND, NOR, XOR, and XNOR, and Buffer illustrating input-output relationships for binary operations. Picture taken from [47].

Now that we have refreshed how logic gates work on classical bits, let us move on to quantum gates. Unlike many classical logic gates, such as NOR or AND, quantum logic gates are reversible. That means that all quantum operations can be undone. Reversibility is a fundamental property of QM and has important implications for QC. Some of them include that it allows changing from one representation basis to another, such as from the computational basis to the Fourier basis in Shor's algorithm [48], and that it enables error correction techniques [49].

Mathematically, quantum gates are represented via unitary matrices. A gate acting on n qubits (a quantum register) is described by a $2^n \times 2^n$ unitary matrix. The set of all such gates, combined with matrix multiplication as the group operation, forms the unitary group $U(2^n)$. The quantum states these gates manipulate are unit vectors in a 2^n -dimensional complex space, utilizing the complex Euclidean norm. The basis vectors of this space, often called eigenstates, represent the possible outcomes when measuring the qubits' states. Any quantum state can be expressed as a linear combination of these basis vectors. Most common quantum gates operate on one or two qubits, analogous to classical logic gates acting on one or two bits. This allows for the construction of more complex quantum circuits and algorithms through the combination of these fundamental operations.

Common quantum gates include the Hadamard gate (H), which creates superposition, the controlled-NOT gate (CNOT) for entanglement operations, and rotation gates like

R_X , R_Y , and R_Z that manipulate the qubit's state on the Bloch sphere. In Fig. 3.3 is depicted a summary of the most common quantum gates, including their mathematical representation.

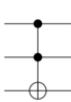
Operator	Gate(s)	Matrix
Pauli-X (X)	 	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)	 	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP	 	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Figure 3.3: Common quantum gates, including their names (with abbreviations), circuit representations, and corresponding unitary matrices. Picture taken from [50].

To illustrate how quantum gates act on a quantum state, let us consider a simple example. We consider the case where the Hadamard gate H , which operates on a single qubit, acts on the computational basis state $|0\rangle$:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}. \quad (3.3)$$

This shows that the Hadamard gate transforms $|0\rangle$ into an equal superposition of $|0\rangle$ and $|1\rangle$.

3.3 Quantum algorithms

Not every problem in classical computing can be solved more efficiently using a quantum algorithm. In fact, only a few selected problems can benefit by quantum algorithms, typically those that can exploit quantum phenomena like superposition and entanglement. Problems that require large-scale parallelism or exhibit specific types of structure, such as factoring large numbers (Shor's algorithm [48]) or solving unstructured search problems (Grover's algorithm [51]), are where quantum algorithms show the most potential. These problems may lead to a quantum speedup, or *quantum advantage*, where quantum

algorithms provide faster solutions than their classical counterparts. For many other computational tasks, classical algorithms remain more efficient and easier to implement. Therefore, the scope of quantum algorithms is currently limited, and their real-world advantages are most evident in specialized areas rather than general-purpose computing.

A quantum algorithm can be understood as a sequence of quantum gates applied to quantum states, followed by measurements, and designed to achieve a specific computational goal. This process is typically represented as a quantum circuit, which serves as a visual and mathematical framework for describing the operations of the algorithm.

To illustrate how designing a quantum algorithm works, we can show a very simple example that given a two-qubit state in the zero state $|00\rangle$ aims to build a Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. In other words we want to build a quantum circuit that can be represented by the unitary U_{Bell} such that:

$$U_{\text{Bell}}|00\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \quad (3.4)$$

Analyzing the form of the Bell state one can realize that we need to perform an operation that incorporates superposition and another one that adds the entanglement to the system. This can be done by the Hadamard gate and the CNOT gate respectively. Therefore the quantum circuit that performs that operation is presented in Fig. 3.4:

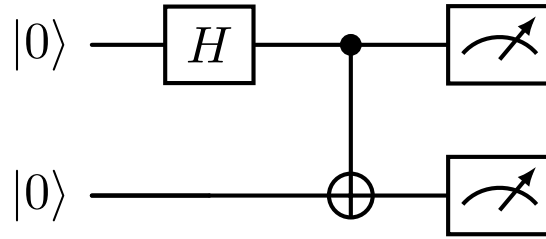


Figure 3.4: Quantum circuit that represents the maximally entangled state, the Bell state.

Note that the circuit in Fig. 3.4 also includes the measurements of both qubits. The measurements are usually the final part of our quantum algorithm, but sometimes they can be skipped if we plan to insert our quantum algorithm as a subroutine in a more complex quantum algorithm.

In the next subsections we will explore some quantum algorithms that are relevant for the topics discussed in this thesis.

3.3.1 Swap Test

Measuring the similarity between quantum states is a crucial component in the development of quantum algorithms, as it is a fundamental part of tasks like state verification, clustering, and pattern recognition. The naive solution one can think of to compare the similarity between a state $|\psi\rangle$ and a state $|\phi\rangle$ involves measuring the qubits of each state and reconstructing them from their measurement probabilities. However, this method is not only suboptimal but also “destructive”, as the act of measurement collapses the quantum states, leaving them unusable for subsequent operations. This limitation highlights the

necessity for a non-destructive method to quantify the similarity between quantum states, preserving their coherence and enabling further processing.

One elegant solution to this problem was proposed in [52] and receives the name of the *Swap Test*. A quantum algorithm that measures the overlap between two quantum states $|\psi\rangle$ and $|\phi\rangle$ in a non-destructive way. The main idea of the *Swap Test* is using an ancillary qubit and a controlled-swap (CSWAP) operation, which exchanges two states if the control qubit is 1, to indirectly infer the overlap between the two states, leaving them unchanged and in their respective quantum registers.

Let us consider $|\psi\rangle$ and $|\phi\rangle$, which are two input real quantum states of n and m qubits, respectively, such that $n \geq m$ (otherwise, we exchange the labels ψ and ϕ), and an ancillary qubit. The controlled *Swap Test* proceeds in three steps:

- **Initialization:**

- Start with the initial state:

$$|\Psi_0\rangle = |0, \psi, \phi\rangle, \quad (3.5)$$

where the ancillary qubit is initialized to $|0\rangle$.

- **Step 1: Apply a Hadamard gate:**

- Apply a Hadamard (H) gate to the ancillary qubit, leaving the states $|\psi\rangle$ and $|\phi\rangle$ unchanged:

$$|\Psi_1\rangle = (H \otimes I^{\otimes n+m}) |\Psi_0\rangle = \frac{1}{\sqrt{2}} (|0, \psi, \phi\rangle + |1, \psi, \phi\rangle). \quad (3.6)$$

- Here, $I^{\otimes n+m}$ is the identity acting on the states $|\psi\rangle$ and $|\phi\rangle$, and the tensor product $\otimes$ is omitted in the composed states (e.g $|0\rangle \otimes |\psi\rangle \otimes |\phi\rangle = |0, \psi, \phi\rangle$).

- **Step 2: Apply the controlled swap gate (CSWAP):**

- Apply the CSWAP gate, which swaps the m qubits of $|\phi\rangle$ with the first m qubits of $|\psi\rangle$:

$$|\Psi_2\rangle = \text{CSWAP} |\Psi_1\rangle = \frac{1}{\sqrt{2}} (|0, \psi, \phi\rangle + |1, \phi, \psi'\rangle). \quad (3.7)$$

- Here, $|\psi'\rangle$ is the swapped version of $|\psi\rangle$, where the m first qubits of ψ are swapped with the rest $n - m$ qubits.

- **Step 3: Apply another Hadamard gate:**

- Apply a Hadamard gate to the ancillary qubit again:

$$|\Psi_3\rangle = (H \otimes I^{\otimes n+m}) |\Psi_2\rangle = \frac{1}{2} (|0\rangle \otimes (|\psi, \phi\rangle + |\phi, \psi'\rangle) + |1\rangle \otimes (|\psi, \phi\rangle - |\phi, \psi'\rangle)). \quad (3.8)$$

The resulting probability of measuring the ancillary qubit in the state $|0\rangle$ is given by:

$$\begin{aligned} P_{\Psi_3}(|0\rangle) &= |\langle 0 | \Psi_3 \rangle|^2 = \frac{1}{4} ||\psi, \phi\rangle + |\phi, \psi'\rangle|^2 = \frac{1}{2} + \frac{1}{2} \text{Re} [\langle \phi, \psi' | \psi, \phi \rangle] \\ &= \frac{1}{2} + \frac{1}{2} \langle \psi' | \phi \rangle \langle \phi | \psi \rangle, \end{aligned} \quad (3.9)$$

If $m = n$, the swapped state $|\psi'\rangle$ simplifies to $|\psi\rangle$, and the probability becomes:

$$P_{\Psi_3}(|0\rangle) = \frac{1}{2} + \frac{1}{2} |\langle\psi|\phi\rangle|^2. \quad (3.10)$$

This provides the squared inner product between the two states, with an uncertainty of $\mathcal{O}(\epsilon)$ after $\mathcal{O}(\epsilon^{-2})$ shots. The corresponding quantum circuit for the *Swap Test* method is shown in Fig. 3.5.

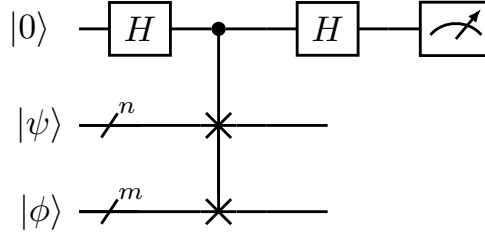


Figure 3.5: Quantum circuit *Swap Test*.

3.3.2 Grover's Algorithm

Grover's algorithm is a quantum search algorithm initially proposed by Lov K. Grover [51] that provides a quadratic speedup over classical methods for searching in unstructured databases. Consider the case of searching for a specific phone number in a randomly-ordered catalogue containing N entries. Using classical methods, one would need to check each entry one by one until the desired number is found. That takes an average of $N/2$ steps and in the worst case one would need to check all the N numbers. That means, the complexity of the classical algorithm to solve this problem is $\mathcal{O}(N)$. However, in [51] Grover presents a quantum algorithm that solves this problem in $\mathcal{O}(\sqrt{N})$ steps, hence providing a quadratic speedup. For example, if the catalogue contained 10000 items, a classical search might require up to 10000 checks, while Grover's quantum algorithm would need only around 100 quantum operations.

Before digging into Grover's algorithm workflow, let us first introduce one concept that constitutes the algorithm's core, the *oracle*. In Grover's algorithm, the quantum *oracle* is a black box that can recognize the solutions to the search problem. This oracle is represented as a unitary operator O whose action over the index x of an element of the list is:

$$O|x\rangle = (-1)^{f(x)}|x\rangle, \quad (3.11)$$

where $|x\rangle$ is the index register and by definition $f(x) = 1$ if x is a solution to the search problem, and $f(x) = 0$ otherwise. Looking at Eq. (3.11), we can say that the oracle marks the solutions to the search problem by introducing a -1 phase.

Another crucial component of Grover's algorithm is the *diffuser* or diffusion operator. The intuition behind this operator is that it will amplify the amplitudes of the states that are marked by the oracle. Mathematically, for a problem with $N = 2^n$ entries, where the indices are represented by n qubits, the diffusion operator, D , is defined as:

$$D = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} = 2|\psi\rangle\langle\psi| - I, \quad (3.12)$$

where $|\psi\rangle$ is the uniform superposition state:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle. \quad (3.13)$$

By joining the oracle and the diffuser, one builds the Grover operator, $G = DO$, which will mark and amplify the probability of measuring a solution to the problem. Moreover, the Grover operator has also a particularly interesting geometrical interpretation. It can be understood as a rotation in the two-dimensional space constituted by the initial state $|\psi\rangle$ and the state consisting of a uniform superposition of all the M solutions to the search problem $|\beta\rangle$. To do so, let us define $|\alpha\rangle$ and $|\beta\rangle$ as:

$$|\alpha\rangle \equiv \frac{1}{\sqrt{N-M}} \sum_x'' |x\rangle, \quad (3.14)$$

$$|\beta\rangle \equiv \frac{1}{\sqrt{M}} \sum_x' |x\rangle, \quad (3.15)$$

where the convention $\sum_x'$ indicates a sum over all x which are solutions and $\sum_x''$ indicates a sum over the rest x which are not solutions. Now, it is convenient to rewrite the initial state $|\psi\rangle$ in a new basis:

$$|\psi\rangle = \sqrt{\frac{N-M}{N}} |\alpha\rangle + \sqrt{\frac{M}{N}} |\beta\rangle. \quad (3.16)$$

In the orthogonal basis defined by $|\alpha\rangle$ and $|\beta\rangle$, the operator G can be understood as a combination of two reflections. The first one is a reflection of the oracle O about the state $|\alpha\rangle$ in the plane defined by $|\alpha\rangle$ and $|\beta\rangle$. Since $O(a|\alpha\rangle + b|\beta\rangle) = a|\alpha\rangle - b|\beta\rangle$. The second one is performed by the diffusion operator D about the state $|\psi\rangle$. This combination of two reflections results in a rotation. It also explains why the state $G^k|\psi\rangle$ remains in the subspace defined by $|\alpha\rangle$ and $|\beta\rangle$ for all k .

Furthermore, this process also gives us the rotation angle. Let $\cos \theta/2 = \sqrt{(N-M)/N}$, so that $|\psi\rangle = \cos \theta/2 |\alpha\rangle + \sin \theta/2 |\beta\rangle$. As illustrated in Fig. 3.6, the two reflections that form G , transform $|\psi\rangle$ into the following state:

$$G|\psi\rangle = \cos \frac{3\theta}{2} |\alpha\rangle + \sin \frac{3\theta}{2} |\beta\rangle, \quad (3.17)$$

so the rotation angle is θ . It can be demonstrated that applying G a number k of times gives us:

$$G^k|\psi\rangle = \cos \left(\frac{2k+1}{2} \theta \right) |\alpha\rangle + \sin \left(\frac{2k+1}{2} \theta \right) |\beta\rangle. \quad (3.18)$$

From Eq. (3.18), it can be understood that the operator G is a rotation that is making the initial state to get closer to the state $|\beta\rangle$ which represents the M solutions of our problem. However, the amplitude of the state $|\beta\rangle$ also depends on the number of iterations k of Grover's operator. Hence, there exists an optimal number of iterations, R , such that applying G^R results in $G^R|\psi\rangle \approx |\beta\rangle$. Beyond this point, for $k > R$, the state begins to deviate from $|\beta\rangle$. Thus, Grover's algorithm can also be viewed as a quantum preparation subroutine, where the initial state $|\psi\rangle$ evolves to approximate the solution $|\beta\rangle$.

At this point, we should ponder the question: how can we determine the number of rotations required to bring the state $|\psi\rangle$ close to the target state $|\beta\rangle$? In other words, how do we estimate R ? To answer this, it is helpful to first recognize that if the initial state is given by Eq. (3.16), a rotation by an angle $\theta' = \arccos \sqrt{\frac{M}{N}}$ will bring the state to $|\beta\rangle$. That can be verified by applying a rotation of angle θ' to $|\psi\rangle$ in the $|\alpha\rangle, |\beta\rangle$ basis:

$$R(\theta')|\psi\rangle = \begin{pmatrix} \cos \theta' & -\sin \theta' \\ \sin \theta' & \cos \theta' \end{pmatrix} \begin{pmatrix} \sqrt{\frac{N-M}{N}} \\ \sqrt{\frac{M}{N}} \end{pmatrix} = \begin{pmatrix} \cos \theta' \sqrt{\frac{N-M}{N}} - \sin \theta' \sqrt{\frac{M}{N}} \\ \sin \theta' \sqrt{\frac{N-M}{N}} + \cos \theta' \sqrt{\frac{M}{N}} \end{pmatrix}, \quad (3.19)$$

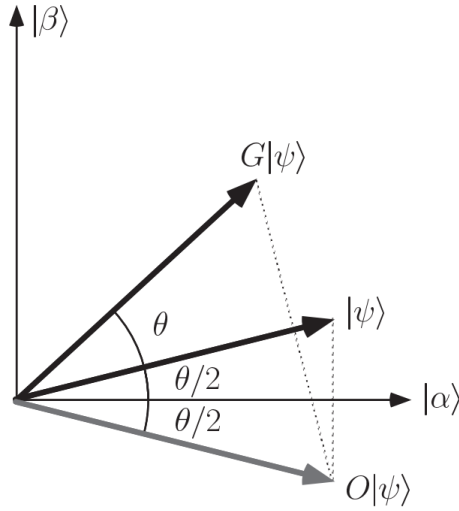


Figure 3.6: Geometrical interpretation of the action of Grover operator G over the initial state $|\psi\rangle$. Picture taken from [17].

applying $\cos \theta' = \sqrt{M/N}$ and $\sin \theta' = \sqrt{(N-M)/N}$ we obtain:

$$R(\theta')|\psi\rangle = \begin{pmatrix} \sqrt{\frac{M}{N}}\sqrt{\frac{N-M}{N}} - \sqrt{\frac{N-M}{N}}\sqrt{\frac{M}{N}} \\ \sqrt{\frac{N-M}{N}}\sqrt{\frac{N-M}{N}} + \sqrt{\frac{M}{N}}\sqrt{\frac{M}{N}} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |\beta\rangle. \quad (3.20)$$

However, we cannot reach the state $|\beta\rangle$ in a single step. It will require several iterations of the Grover operator. This is exactly what we aim to determine: the number of required iterations R . Essentially, we can decompose this large rotation of angle θ' into R smaller rotations of angle θ , which means that $R\theta = \theta'$. Hence, we can obtain R as:

$$R = \text{CI} \left(\frac{\arccos \sqrt{M/N}}{\theta} \right), \quad (3.21)$$

where $\text{CI}(x)$ denotes the integer closest to the real number x . The expression in Eq. (3.21) provides an exact formula for the number of oracle calls required to perform Grover's algorithm. However, in some cases, it will be more convenient to have a simpler expression that captures the essential behavior of R . In particular, from Eq. (3.21) we see that $R \leq \lceil \pi/2\theta \rceil$, which means that a lower bound on θ will give an upper bound on R . Assuming that $M \leq N/2$ we have:

$$\sqrt{\frac{M}{N}} = \sin \theta/2 = \frac{\theta}{2}. \quad (3.22)$$

Combining this with the upper bound we already had on R we obtain this elegant upper bound:

$$R \leq \left\lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} \right\rceil. \quad (3.23)$$

That is, $R = \mathcal{O} \left(\sqrt{\frac{N}{M}} \right)$ Grover iterations (and thus oracle calls) are needed to obtain a solution to the search problem with high probability, representing a quadratic improvement over the $\mathcal{O} \left(\frac{N}{M} \right)$ oracle calls classically required.

We now illustrate Grover's algorithm with a simple example. Let us consider the case where we are looking for one item in a list of four elements, whose indices are represented as: $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$. Without loss of generality we will consider $|x_0\rangle = |11\rangle$ as the solution. The quantum circuit that represents Grover's search is depicted in Fig. 3.7.

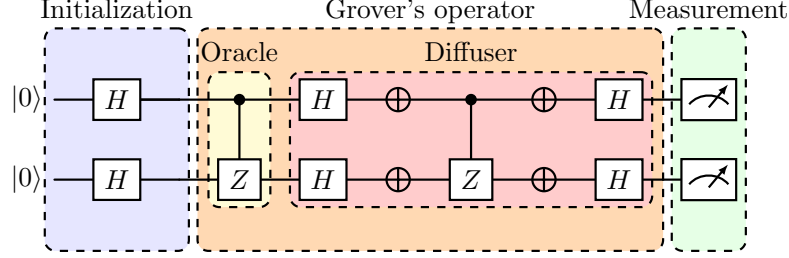


Figure 3.7: Quantum circuit representing Grover's algorithm for two qubits.

Note that the oracle of Fig. 3.7 marks the state $|11\rangle$ as the CZ will add a -1 phase if and only if both qubits are in the state $|1\rangle$. The quantum circuit of Fig. 3.7 performs the following operations:

- **Initialization:**

$$|\psi\rangle = \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle). \quad (3.24)$$

- **Apply Grover operator $R \approx \left\lceil \pi\sqrt{4}/4 \right\rceil \approx 1$ times:**

$$\begin{aligned} G|\psi\rangle &= DO|\psi\rangle = D \left[\frac{1}{2} (|00\rangle + |01\rangle + |10\rangle - |11\rangle) \right] \\ &= (2|\psi\rangle\langle\psi| - I) \left[\frac{1}{2} (|00\rangle + |01\rangle + |10\rangle - |11\rangle) \right] \\ &= \left(2|\psi\rangle\langle\psi| \frac{1}{4} (1 + 1 + 1 - 1) - \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle - |11\rangle) \right) \\ &= |\psi\rangle - \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle - |11\rangle) \\ &= \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) - \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle - |11\rangle) \\ &= \frac{1}{2} (|00\rangle - |00\rangle + |01\rangle - |01\rangle + |10\rangle - |10\rangle + |11\rangle + |11\rangle) \\ &= |11\rangle \end{aligned} \quad (3.25)$$

- **Measurement:**

We measure the $n = 2$ qubits and obtain $x_0 = |11\rangle$ with probability 1.

In this example, we showed how Grover's algorithm solves a relatively simple unstructured search problem. However, its applications extend far beyond this. Grover's algorithm can speed up tasks such as finding the minimum of a function [53], addressing encryption problems involving prime factorization [54], and one of its most fascinating uses: quantum Monte Carlo integration [55], which is achieved through the Quantum Amplitude Estimation (QAE) algorithm [56]. This application of Grover's algorithm in quantum integration will be explored in detail in Chapter 5.

3.3.3 Variational Quantum Algorithms

The NISQ (Noisy Intermediate-Scale Quantum) [57] era refers to the current stage of quantum computing², where quantum processors have tens to a few hundred qubits, which are enough to explore interesting problems but are still subject to noise and errors. These devices are not yet capable of performing fault-tolerant quantum computations, which are not restricted by the number of qubits or gates, but they offer a unique opportunity to experiment with quantum algorithms and explore quantum advantage in real-world applications. In this context, Variational Quantum Algorithms (VQAs) are an incredibly popular approach for quantum algorithms on near-term quantum devices.

VQAs rely on Parameterized Quantum Circuits (PQC), which are quantum circuits that contain gates with free parameters that can be tuned. Hence, the goal of a VQA is to find the optimal parameters that achieve the desired task. This parameter tuning is handled by a classical computer, making VQA a quantum-classical hybrid algorithm, with the PQC representing the quantum component, while the tuning of parameters is the classical component. The next question is, how to find the correct set of parameters that solve a particular problem? To do so, the so-called *cost function* quantifies how far the PQC is from achieving its goal. Therefore, the cost function is a function of the parameters that, when minimized, indicates that we have performed the task well enough.

In most cases, VQAs is divided into three main steps:

1. A PQC, often referred to as an *Ansatz*, that is defined with a set of free parameters.
2. Measurement of an observable and computation of a cost function based on these measurements.
3. Optimization of the free parameters using a classical computer, which interacts with the quantum device to refine the parameters iteratively. This optimization is usually performed by a gradient descent method, an optimization algorithm to find a local minimum of a differentiable function.

3.3.3.1 Ansatz

In physics and mathematics, the German word “Ansatz” or in plural “Ansätze” (“Ansatzes” in English), refers to an educated guess for solving a problem. In the context of VQAs, the Ansatz refers to the quantum circuit $U(\vec{\theta})$ we design to solve a specific task. But how do we choose such a circuit Ansatz? There are two main approaches. One approach is to use prior knowledge about the problem or physical intuition to design a tailored Ansatz, often referred to as a “problem-inspired Ansatz”.

What if we lack sufficient understanding of the problem to guide our design? In such cases, we can use universal, “problem-agnostic” Ansatzes that are applied without specific information about the problem. These Ansatzes are generally larger, containing many gates and parameters, which makes the optimization process more challenging. The better an Ansatz is tailored to the specific task, the easier it becomes for the optimizer to find the right parameters, increasing the likelihood of solving the problem in a reasonable amount of time.

²Part of the quantum computing community is starting to discuss the transition from the NISQ era to the early fault-tolerant era, or ISQ era [58], where noise becomes less of a problem due to advancements in error correction. This would enable the implementation of quantum algorithms with significantly deeper quantum circuits.

There are two key factors to consider when evaluating how well an Ansatz is suited to a particular problem: *expressibility* and *trainability*. *Expressibility* measures the range of functions the PQC can represent, which is often related to the amount of entanglement and the number of parameters in the circuit. While greater expressibility allows the PQC to describe more complex functions, it also makes the optimization process, and hence the *trainability*, more difficult. This is because a larger space of functions increases the complexity of finding the optimal parameters. *Trainability* also depends on the choice of the cost function and optimizer.

Typically, there is a trade-off between *expressibility* and *trainability*, since larger and more expressive PQCs are harder to train. Finding an Ansatz that balances enough *expressibility* with feasible *trainability* is essential for designing effective quantum circuits.

3.3.3.2 Cost functions

Once we have chosen the Ansatz circuit for our VQA, the next step is to determine the parameters required to perform the desired task. This task is addressed by defining a cost function, which is a mathematical function that quantifies how close the output of the PQC is to the desired result.

However, that does not mean that we necessarily know the answer. In supervised problems, we often know the solution for some specific inputs, even if not for all. For these known cases, we optimize the parameters to match the expected outcomes, assuming that the same parameters will work for unknown inputs.

In other cases, the goal could simply be to minimize a cost function without having prior knowledge of the solution. VQAs are well-suited for this task, as they iteratively adjust the parameters to find the minimum value of the cost function. The cost function is typically defined as the expectation value of a chosen observable $\hat{M}$:

$$C(\vec{\theta}) = \langle 0|U^\dagger(\vec{\theta})\hat{M}U(\vec{\theta})|0\rangle. \quad (3.26)$$

However, constructing an effective cost function $C(\vec{\theta})$ can be challenging. As explained in [59], some points can help us make the right choice:

- **Faithfulness:** The minimum of the cost function $C(\vec{\theta})$ should correspond to the solution of the problem.
- **Efficient Estimation:** It must be possible to estimate $C(\vec{\theta})$ efficiently through quantum measurements and potentially classical post-processing. Additionally, to maintain a quantum advantage, $C(\vec{\theta})$ should not be efficiently computable classically.
- **Operational Meaningfulness:** Lower values of $C(\vec{\theta})$ should correspond to better solutions.
- **Trainability:** It should be feasible to optimize the parameters efficiently.

By carefully considering these criteria, we can design cost functions that enhance the effectiveness and performance of VQAs.

3.3.3.3 Optimizer and Gradient Descent

Once the Ansatz and the cost function are defined, the next step is to optimize the trainable parameters. Mathematically, this optimization task is expressed as finding the set of

parameters θ^* that minimize the cost function $C(\theta)$:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} C(\theta). \quad (3.27)$$

There are numerous methods to perform this optimization, but gradient-based techniques are particularly advantageous. These methods leverage information about the gradient (or higher-order derivatives) of the cost function to iteratively update the parameters, driving the cost function towards a minimum.

At this point, a natural question arises: How can we compute the gradient of a PQC? At first glance, this might seem challenging due to the quantum nature of the system. However, thanks to the Parameter Shift Rule (PSR), it is possible to compute these gradients analytically, even for quantum circuits. The PSR is a powerful mathematical tool that enables gradient computation by evaluating the expectation value of the cost function at two specific points. While PQCs typically consist of multiple gates, we focus here on computing the gradient of a single gate for simplicity and pedagogical clarity. For a more general treatment of PQCs with multiple gates, we refer the reader to [60]. That said, let us consider a unitary gate $U(\theta)$ of the form:

$$U(\theta) = e^{-i\frac{\pi}{2}\theta\hat{G}_j}, \quad (3.28)$$

where $\hat{G}_i$ is the Hermitian generator of $U(\theta)$ and corresponds to a Pauli operator. By using Taylor expansion of the matrix exponential, the gradient of $U(\theta)$ is expressed as:

$$\nabla U(\theta) = -\frac{i}{2}\hat{G}_j U(\theta) = -\frac{i}{2}U(\theta)\hat{G}_j. \quad (3.29)$$

Then, we aim to compute the gradient of the following quantum circuit in Fig. 3.8, which corresponds to the function:

$$f(\theta) = \langle \hat{M} \rangle = \langle 0|U^\dagger(\theta)\hat{M}U(\theta)|0\rangle. \quad (3.30)$$

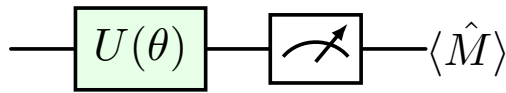


Figure 3.8: Quantum circuit measuring the expectation value of the observable $\hat{M}$.

The gradient of $f(\theta)$ is given by:

$$\begin{aligned} \nabla f(\theta) &= \nabla \langle \hat{M} \rangle = \langle 0|(\nabla U^\dagger(\theta)\hat{M}U(\theta) + U^\dagger(\theta)\hat{M}\nabla U(\theta))|0\rangle \\ &= \frac{i}{2}\langle 0|U^\dagger(\theta)(\hat{G}_j\hat{M} - \hat{M}\hat{G}_j)U(\theta)|0\rangle \\ &= \frac{i}{2}\langle 0|U^\dagger(\theta)[\hat{G}_j, \hat{M}]U(\theta)|0\rangle. \end{aligned} \quad (3.31)$$

This can be simplified using the following expression for commutators involving Pauli operators:

$$[\hat{G}_j, \hat{M}] = -i \left(U^\dagger \left(\frac{\pi}{2} \right) \hat{M} U \left(\frac{\pi}{2} \right) - U^\dagger \left(-\frac{\pi}{2} \right) \hat{M} U \left(-\frac{\pi}{2} \right) \right), \quad (3.32)$$

and substituting in previous expression we obtain the gradient of the circuit:

$$\nabla f(\theta) = \frac{1}{2} \langle 0|U^\dagger \left(\theta + \frac{\pi}{2} \right) \hat{M}U \left(\theta + \frac{\pi}{2} \right) |0\rangle - \frac{1}{2} \langle 0|U^\dagger \left(\theta - \frac{\pi}{2} \right) \hat{M}U \left(\theta - \frac{\pi}{2} \right) |0\rangle. \quad (3.33)$$

Finally, rewriting this in terms of $f(\theta)$, the gradient of the quantum circuit is:

$$\nabla f(\theta) = \frac{1}{2} \left[f \left(\theta + \frac{\pi}{2} \right) - f \left(\theta - \frac{\pi}{2} \right) \right]. \quad (3.34)$$

The PSR unlocks the possibility of using gradient-based optimization algorithms, such as Gradient Descent (GD), for variational quantum circuits. (GD) is a fundamental optimization technique used to iteratively minimize a function. It is widely employed in various fields, including Machine Learning, numerical optimization, and Quantum Computing. The goal of GD is to find the minimum of a cost function by iteratively adjusting parameters in the direction of the steepest descent of the function.

The GD algorithm can be summarized in the following steps:

1. **Initialization:** Start with an initial guess for the parameters $\boldsymbol{\theta}$, denoted as $\boldsymbol{\theta}^{(0)}$. This initial guess can be random or based on some heuristic.
2. **Compute gradient:** Calculate the gradient $\nabla f(\boldsymbol{\theta}^{(k)})$, which represents the vector of partial derivatives of $f(\boldsymbol{\theta})$ with respect to each parameter θ_k . For each parameter:

$$(\nabla f(\boldsymbol{\theta}))_k = \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_k}. \quad (3.35)$$

In practice, libraries like **PennyLane** [61] use efficient methods such as the PSR to compute these gradients for quantum circuits.

3. **Update parameters:** Adjust the parameters in the opposite direction of the gradient to minimize the function. The update rule is given by:

$$\boldsymbol{\theta}^{(kt+1)} = \boldsymbol{\theta}^{(k)} - \eta \nabla f(\boldsymbol{\theta}^{(kt)}), \quad (3.36)$$

where η is the learning rate, a small positive scalar that controls the size of the steps taken towards the minimum. Choosing an appropriate learning rate is crucial: too small learning rate can result in slow convergence, while too large learning rate can cause oscillations or even divergence.

4. **Iterate until convergence:** Repeat steps 2 and 3 until a stopping criterion is met. This criterion can be based on the change in the function value or the gradient magnitude falling below a threshold, indicating that further updates are not significantly improving the result. Alternatively, steps 2 and 3 can be executed for a fixed number of iterations.

A graphical interpretation of GD is shown in Figure 3.9. Essentially, the gradient $\nabla f(\boldsymbol{\theta})$ indicates the direction of the steepest increase of the function $f(\boldsymbol{\theta})$. Moving in the opposite direction ensures movement towards the function's minimum. The gradient is a vector of partial derivatives, representing how much the function value changes with respect to changes in its parameters. The learning rate η controls the size of the step taken in the direction of the gradient, balancing the trade-off between convergence speed and stability. A well-chosen learning rate ensures efficient and stable convergence to the function's minimum.

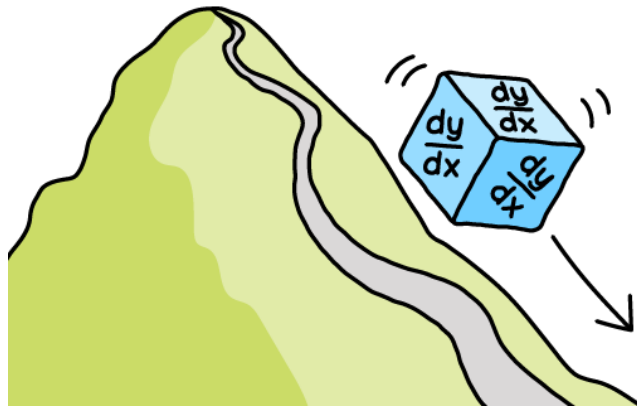


Figure 3.9: Graphical representation of Gradient Descent. The algorithm iteratively moves towards the minimum of the cost function by following the direction of the negative gradient. Picture taken from [62]

In Quantum Computing, GD is used to optimize parameters in PQCs. By tuning the variational parameters of a PQC, GD minimizes the cost function. The combination of the Parameter Shift Rule and GD provides a powerful framework for training VQAs, enabling efficient optimization of PQCs.

In summary, the PSR provides a practical and efficient way to compute gradients of PQCs, enabling the use of gradient-based optimization techniques. GD's ability to iteratively refine parameters towards a function's minimum, coupled with its efficiency and versatility, makes it indispensable for optimization tasks in modern computational problems. Whether optimizing classical Machine Learning models or fine-tuning quantum circuits, GD constitutes a fundamental technique for achieving effective and efficient optimization.

3.4 Quantum Machine Learning

Over the past years, the term ‘Quantum Machine Learning’ (QML) has become ubiquitous, appearing in various contexts and usually referring to the intersection of quantum information theory and machine learning, see Fig. 3.10.

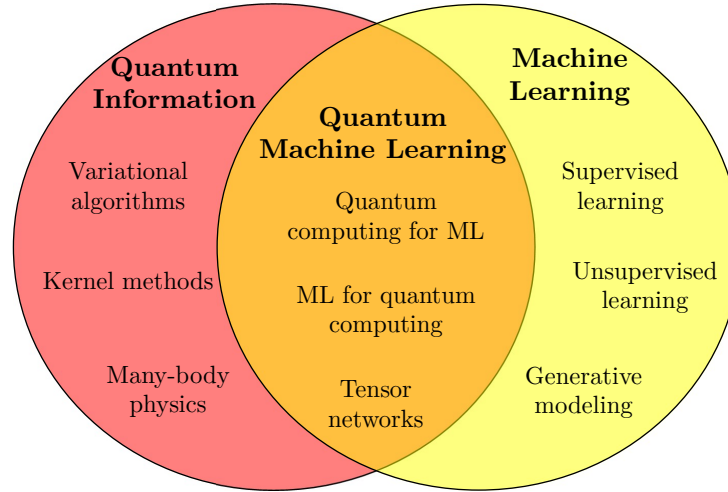


Figure 3.10: Venn diagram of the intersection between quantum information and Machine Learning (ML).

This broad field includes Machine Learning (ML) techniques applied to quantum information processing [63, 64, 65] and classical computational tools developed within the quantum community, such as the use of tensor networks from many-body physics to train neural networks [66], and the use of quantum algorithms to tackle ML tasks [67, 68].

Another interesting point of view to analyze the different ways of doing QML is to analyze the nature of the algorithms and the data, as depicted in Fig. 3.11

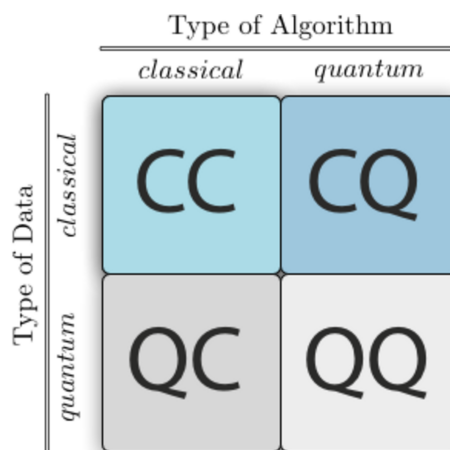


Figure 3.11: Quantum Machine Learning (QML) approaches by algorithm and data nature. Picture taken from [69].

In this thesis, we will primarily focus on quantum algorithms for tackling ML problems with classical data (first item on the overlapping area in Fig. 3.10 and top right corner of Fig. 3.11). Within this category, we consider three different categories that have been

widely explored: quantum kernels, quantum neural networks and quantum generative models. Nevertheless, before diving into these categories it is convenient to consider the important question of how to encode data in a quantum computer in the context of QML.

3.4.1 Encoding

Data is the core component of ML. Its efficient processing allows models to make predictions that might be otherwise unreachable. In QML, this role also remains crucial. However, the way data is processed and represented varies significantly. Quantum computers offer unique computational advantages, but to leverage their properties effectively, data must first be mapped onto quantum states. Encoding classical data into a quantum computer, is a key challenge in designing QML algorithms, as the choice of encoding method impacts efficiency, expressiveness, and the feasibility of the potential quantum advantage. Moreover, the best encoding strategy often depends on the specific problem at hand. The most common approaches to load data into quantum computers include basis encoding, amplitude encoding, angle encoding and dynamical encoding [68].

3.4.1.1 Basis encoding

Basis encoding (or basis embedding) maps a classical n -bit string (e.g., 0011) to a computational basis state of an n -qubit system (e.g., $|\psi\rangle = |0011\rangle$). This approach is relatively straightforward, as each classical bit is directly replaced by a qubit. Like classical computing, basis encoding relies on a binary representation of numbers. A quantum state $|x\rangle$, where $x \in \mathbb{R}$, refers to a binary representation of x using $\log x$ qubits.

Let us now consider a classical dataset D that we want to load into a quantum computer. For basis embedding, each item $x^{(m)}$ has to be a N -bit binary string; $x^{(m)} = (b_1, \dots, b_N)$ with $b_i \in \{0, 1\}$ for $i = 1, \dots, N$. Assuming all features are represented with unit binary precision (one bit), each input example $x^{(m)}$ can be directly mapped to the quantum state $|x^{(m)}\rangle$. This means that the number of quantum subsystems, n , must be at least equal to N . An entire dataset is represented in superpositions of computational basis states as

$$|D\rangle = \frac{1}{\sqrt{M}} \sum_{m=1}^M |x^{(m)}\rangle. \quad (3.37)$$

To illustrate this, consider the set $D = \{2, 3, 7, 9\}$, which corresponds to $D = \{0010, 0011, 0111, 1001\}$ in binary. Therefore, the quantum state is given by:

$$|D\rangle = \frac{1}{\sqrt{4}} (|0010\rangle + |0011\rangle + |0111\rangle + |1001\rangle). \quad (3.38)$$

This encoding technique is particularly useful in binary classification, quantum feature mapping, and quantum error correction, but it might not be ideal for other tasks such as regression.

3.4.1.2 Amplitude encoding

Amplitude encoding represents classical data by encoding each value in the amplitudes of a quantum state. Specifically, a classical dataset $D = \{x^{(1)}, x^{(2)}, \dots, x^{(M)}\}$, where each $x^{(m)}$ is a scalar, is mapped to a quantum state $|D\rangle$ such that each classical value $x^{(m)}$ appears as the amplitude of the corresponding computational basis state. The quantum state $|D\rangle$ is a superposition of basis states, where the amplitudes are directly proportional

to the values in the dataset. Mathematically, the quantum state is written as:

$$|D\rangle = \frac{1}{\sqrt{\sum_{m=1}^M |x^{(m)}|^2}} \sum_{m=1}^M x^{(m)} |x^{(m)}\rangle. \quad (3.39)$$

Let us now, consider the same dataset as before $D = \{2, 3, 7, 9\}$. The corresponding quantum state will now be:

$$|D\rangle = \frac{1}{\sqrt{2^2 + 3^2 + 7^2 + 9^2}} (2|00\rangle + 3|01\rangle + 7|10\rangle + 9|11\rangle). \quad (3.40)$$

This approach could be more efficient than basis encoding, as it allows the quantum system to represent a larger range of data with fewer qubits. Amplitude encoding is particularly useful for representing compactly and efficiently large datasets in quantum computing. It is commonly used in tasks like multivariable classification.

3.4.1.3 Angle encoding

Angle encoding maps classical data to a quantum computer by representing each value as the angle of a quantum rotation. This encoding technique is implemented encoding each angle in a different qubit via a rotation gate such as R_X, R_Y or R_Z . Specifically, a classical dataset $D = \{\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(M)}\}$, where each $\theta^{(m)}$ is a scalar, is mapped to a quantum state $|D\rangle$ such that each qubit codifies one angle. Therefore, if we consider rotations around the X axis the quantum state is written as:

$$|D\rangle = \bigotimes_{m=1}^M R_X |\theta_m\rangle = \cos\left(\frac{\theta_m}{2}\right) |0\rangle - i \sin\left(\frac{\theta_m}{2}\right) |1\rangle, \quad (3.41)$$

Now, if we want to encode the vector of the dataset $D = \{2, 3, 7, 9\}$ using R_X rotations, the resulting quantum state would be:

$$\begin{aligned} |D\rangle &= (R_X(2)|0\rangle \otimes R_X(3)|0\rangle \otimes R_X(7)|0\rangle \otimes R_X(9)|0\rangle) \\ &= \left((\cos(1)|0\rangle - i \sin(1)|1\rangle) \otimes (\cos(1.5)|0\rangle - i \sin(1.5)|1\rangle) \otimes \right. \\ &\quad \left. (\cos(3.5)|0\rangle - i \sin(3.5)|1\rangle) \otimes (\cos(4.5)|0\rangle - i \sin(4.5)|1\rangle) \right). \end{aligned} \quad (3.42)$$

Angle encoding is particularly efficient for tasks where the data needs to be represented in a way that is compatible with quantum operations. It can provide a more compact representation of data, offering an advantage in certain problems such as regression.

3.4.1.4 Dynamic encoding

Dynamic encoding is a technique that embeds classical or quantum information into the evolution of a quantum system. Instead of encoding data into static quantum states, this approach encodes information into unitary operators that govern the system's time evolution. A common strategy is to associate a given square matrix A with a Hamiltonian H , enabling the representation of A through quantum evolution. If A is non-Hermitian, one defines an extended Hermitian matrix as

$$\tilde{A} = \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}, \quad (3.43)$$

and use only part of the quantum system's output. This method allows quantum algorithms to manipulate matrices efficiently, including operations such as matrix-vector multiplication and matrix inversion, particularly in amplitude encoding schemes. Another perspective on dynamic encoding emerges when analyzing the evolution of a subsystem within a larger unitary system. This encoding strategy provides a flexible way to load information into a quantum computer, allowing for applications in quantum algorithms and quantum simulations.

3.4.2 Quantum Kernels

Quantum kernels have emerged as a promising tool in QML, offering potential advantages over classical kernel methods. Kernel methods work by mapping data into a higher-dimensional space, where patterns that may be complex in the original space become more easily separable. These techniques are widely used in supervised learning to transform nonlinear problems into linear ones by leveraging high-dimensional feature representations.

To understand how kernel methods work we follow [70] approach of considering the example of linear classification. One of the most well-known kernel methods for solving such problems is the support vector machine (SVM) [71], a supervised binary classifier that aims to determine the hyperplane that best divides data in two groups. The SVM algorithm will perform the binary classification of a datapoint according to the following expression:

$$y(\vec{x}) = \text{sign}(\langle \vec{w}, \vec{x} \rangle + b), \quad (3.44)$$

where $\langle \vec{w}, \vec{x} \rangle$ denotes the inner product of the vector $\vec{w}$, which is perpendicular to the hyperplane, and the input $\vec{x}$, while b determines the hyperplane's position. A sketch of this binary classification problem is depicted in Fig. 3.12.

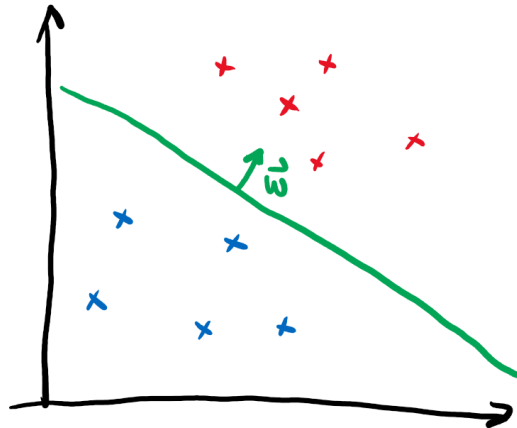


Figure 3.12: Binary classification problem using a Support Vector Machine (SVM). Picture taken from [72].

Then, following the kernel's definition, one remaps $\vec{x}$ into a larger feature space: $\vec{x} \rightarrow \phi(\vec{x})$. Also, one rewrites $\vec{w}$ as a linear combination of the remapped datapoints: $\vec{w} = \sum_i \alpha_i \phi(\vec{x}_i)$. Putting this all together, the previous formula reads:

$$y(\vec{x}) = \text{sign} \left(\sum_i \alpha_i \langle \phi(\vec{x}_i), \phi(\vec{x}) \rangle + b \right). \quad (3.45)$$

At this point, the prediction only depends on inner products in the higher-dimensional feature space. These inner products, which measure the similarity between two datapoints, are what we will call kernel functions:

$$k(\vec{x}_i, \vec{x}_j) = \langle \phi(\vec{x}_i), \phi(\vec{x}_j) \rangle. \quad (3.46)$$

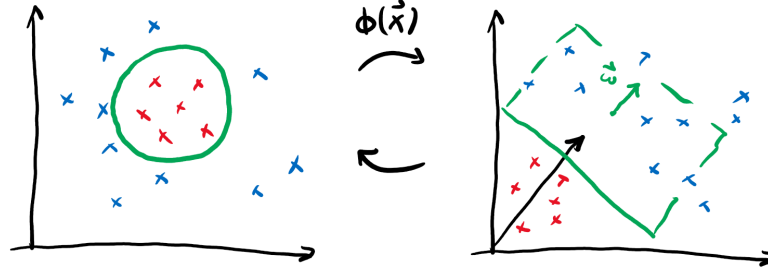


Figure 3.13: A nonlinear embedding used to enhance the capabilities of a linear classifier. Linear classification in the embedding space translated to nonlinear decision boundaries in the original space. Picture taken from [72]

It is now relatively straightforward to generalize this kernel concept to a more general framework, introducing the notion of a *quantum* kernel. This connection to quantum computing naturally arises when we interpret the feature space as a Hilbert space and the mapping $\phi(\vec{x})$ as a quantum state within it:

$$|\phi(\vec{x})\rangle = S(\vec{x})|0\rangle, \quad (3.47)$$

where $S(\vec{x})$ represents the embedding routine used to prepare this state. With this perspective, we can define the *quantum* kernel as:

$$k(\vec{x}_i, \vec{x}_j) = |\langle \phi(\vec{x}_i) | \phi(\vec{x}_j) \rangle|^2. \quad (3.48)$$

This quantum kernel is then be used as a subroutine to efficiently compute similarities between datapoints in a quantum-enhanced SVM, leading to the so-called Quantum Support Vector Machine (QSVM) [73].

To implement a quantum kernel in a quantum circuit, we prepare the two states $|\phi(x_i)\rangle$ and $|\phi(x_j)\rangle$ on different sets of qubits using angle-embedding routines $S(x_i)$ and $S(x_j)$, and measure their overlap with the SWAP test subroutine explained in Section 3.3.2. However, we only need half the number of qubits n if we prepare $|\phi(x_i)\rangle$ and then apply the inverse embedding with x_j on the same qubits. We then measure the projector onto the initial state $|0^{\otimes n}\rangle\langle 0^{\otimes n}|$. This corresponds to the quantum circuit of Fig. 3.14.

We can mathematically verify that this quantum circuit, when measuring the observable $\hat{M} = |0^{\otimes n}\rangle\langle 0^{\otimes n}|$, produces the desired quantum kernel:

$$\begin{aligned} \langle 0^{\otimes n} | S(x_j) S(x_i)^\dagger \hat{M} S(x_j)^\dagger S(x_i) | 0^{\otimes n} \rangle &= \langle 0^{\otimes n} | S(x_j) S(x_i)^\dagger | 0^{\otimes n} \rangle \langle 0^{\otimes n} | S(x_j)^\dagger S(x_i) | 0^{\otimes n} \rangle \\ &= |\langle 0^{\otimes n} | S(x_j)^\dagger S(x_i) | 0^{\otimes n} \rangle|^2 \\ &= |\langle \phi(x_j) | \phi(x_i) \rangle|^2 \\ &= \kappa(x_i, x_j). \end{aligned} \quad (3.49)$$

As expected, the quantum circuit computes the quantum kernel, which is inserted to enhance ML algorithms. Thus, we have shown how quantum kernels can be leveraged in QML.

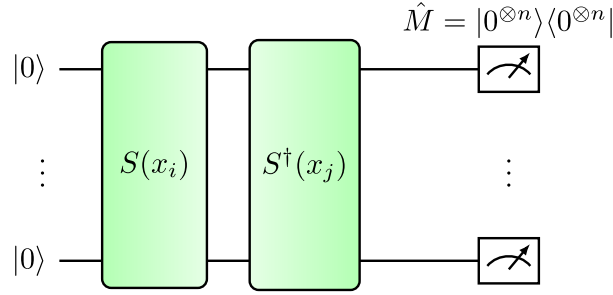


Figure 3.14: Quantum kernel circuit for states encoded by $S(x_i)$ and $S(x_j)$.

3.4.3 Quantum Neural Networks

The term Quantum Neural Network (QNN) has been widely used in recent years [74, 75, 76, 77, 78] to describe ML models that combine elements from quantum computing and artificial neural networks. The term originates from the fact that these models mimic, to some extent, the behavior of Neural Networks (NN). They share with their classical analogues a modular structure, with quantum gates functioning similarly to layers in classical neural networks, and the optimization of trainable parameters using techniques such as GD. Nevertheless, their internal functioning is fundamentally different. To provide a comprehensive comparison and explanation of these topics, let us begin by introducing how neural networks work.

A NN consists of a set of nodes or neurons that communicate with each other by transmitting information [79]. Input data passes through the network, where it undergoes various transformations and produces the algorithm's predictions. The architecture of a classification NN is as follows (see Fig. 3.15): an input layer with as many neurons as the number of input variables, one (simple NN) or several (deep NN) hidden layers with n neurons, where n is a hyperparameter of the model introduced by the user and an output layer with as many neurons as classes.

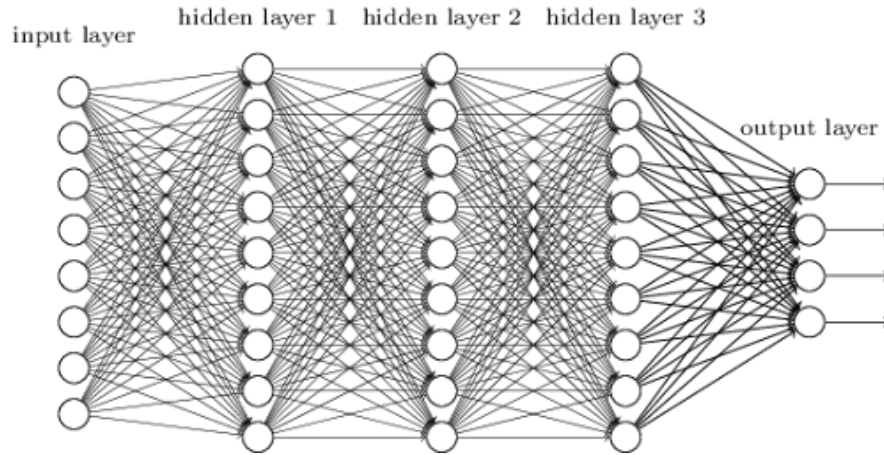


Figure 3.15: Diagram of a classical Neural Network (NN) with three hidden layers. Picture taken from [80].

The NN works as follows. First, the neurons in the first layer process the value corresponding to the input variables. Then, in the rest of the layers, each neuron has a weighted sum of the values of the neurons in the previous layer [81]. The weights of this weighted sum are initially randomized, but through backpropagation learning, they are

modified in successive iterations of the training process to make predictions increasingly accurate [82].

This type of algorithm has several hyperparameters introduced by the user that affect the functioning and learning process of ML algorithms. Some of the hyperparameters of NN algorithms are [83]:

- **hidden layers:** number of hidden layers.
- **neurons:** number of neurons in each layer.
- **dropout:** percentage of neurons randomly eliminated in each hidden layer during each iteration of the training process. This procedure prevents overfitting, as patterns are not established between subsets of neurons that memorize training data instead of learning.
- **batch size:** number of training data samples used to train in each iteration of the process.
- **validation split:** fraction of training data used as validation data. The model separates this fraction of data for the training process and subsequently performs a validation of the trained model to adjust the weights of the neurons, obtaining a more accurate model.
- **epochs:** number of times the entire training set is used for training.

On the other hand, QNN have a workflow that presents similarities and differences. QNNs can load classical data (inputs) into quantum states, which can be later processed by using quantum gates parametrized by trainable parameters (weights). Fig. 3.16 depicts a generic QNN example including the data loading and processing steps. After this processing, the quantum state is measured and the result is employed to compute a loss function. This loss function serves as the metric to be minimized during the training process. The weights are adjusted through optimization routines, which can be either gradient-free or gradient-based, such as GD. These optimization routines are usually the same as those used in classical NNs.

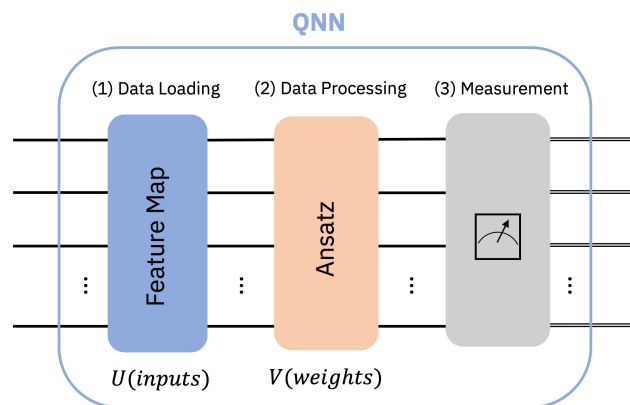


Figure 3.16: Generic Quantum Neural Network (QNN) structure. Picture taken from [84].

While the QNN structure depicted in Fig. 3.16 provides a good general representation, it is worth noting that other architectures have also demonstrated effectiveness, sometimes surpassing this generic model in specific applications. One remarkable example, which will be explored further in this thesis, is the data-reuploading scheme [85]. The

data-reuploading approach to QNNs addresses a significant challenge in implementing these algorithms on NISQ devices. Traditional QNN schemes, as illustrated in Fig. 3.16, often require a large number of qubits to effectively learn and represent the intricate underlying patterns in complex high-dimensional datasets [86]. The data-reuploading technique offers an elegant solution to this limitation. By repeatedly "re-uploading" input data into the quantum system, this approach enhances the flexibility and expressivity of quantum models. This enables the approximation of complex functions without necessitating large quantum systems or sophisticated encoding strategies. The data-reuploading QNN architecture consists of a sequence of alternating trainable and encoding blocks. In particular, the QNN consists of L layers. Each containing a data-encoding block $S(\vec{x})$ and a trainable block $\mathcal{A}(\vec{\theta})$. The quantum unitary that represents the quantum circuit is expressed as:

$$U(\vec{x}, \vec{\theta}) = \mathcal{A}(\vec{\theta}_L)S(\vec{x})\mathcal{A}(\vec{\theta}_{L-1}) \dots \mathcal{A}(\vec{\theta}_1)S(\vec{x})\mathcal{A}(\vec{\theta}_0). \quad (3.50)$$

The training blocks $\mathcal{A}(\vec{\theta})$ depend on the parameters $\vec{\theta}$ that are optimized classically. A sketch of this method is depicted in Fig. 3.17. Another key motivation behind data

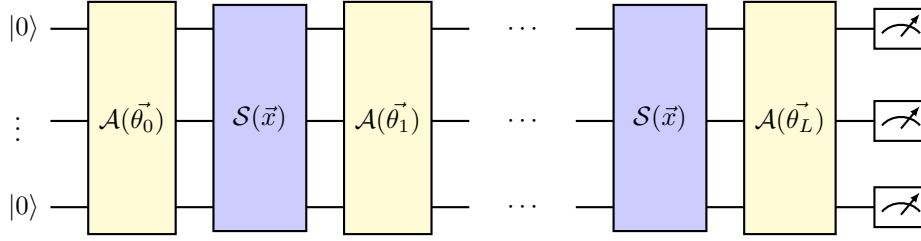


Figure 3.17: Quantum neural network following a data-reuploading scheme.

re-uploading is overcoming the limitations imposed by the quantum no-cloning theorem, which prevents the duplication of quantum states. Unlike classical NNs, which can reuse input data freely, quantum circuits face restrictions on copying quantum information. The re-uploading scheme overcomes this issue by repeatedly feeding the data into the quantum circuit at different layers alongside additional quantum gates with trainable parameters. This approach draws a parallel with classical NNs, where the same input is passed through multiple layers to extract hierarchical features. As a result, the re-uploading framework provides a natural bridge between quantum circuits and classical NNs, reinforcing the relevance of quantum models in ML tasks.

Furthermore, the re-uploading technique has implications beyond QML algorithms. This strategy allows quantum information stored in registers to be efficiently processed through controlled operations. This flexibility makes it a valuable tool for designing larger quantum algorithms where classical data needs to be integrated with quantum processing, such as loading a function into a quantum state.

Another relevant aspect of QNNs is how the chosen encoding scheme influences the set of functions they can represent. In this section, we demonstrate how a specific encoding enables QNNs to represent Fourier series [87, 88]. For simplicity, we focus on the case of univariate functions with inputs $x \in \mathbb{R}$, although this can be generalized as outlined in Appendix A of [87]. We define a quantum model $f_\theta(x)$ as the expectation value of an observable, where the state is prepared through a PQC:

$$f_\theta(x) = \langle 0 | U^\dagger(x, \theta) \hat{M} U(x, \theta) | 0 \rangle, \quad (3.51)$$

where $|0\rangle$ is the n -th qubit initial state of the quantum computer, $U(x, \theta)$ is a quantum circuit that depends on the input x and a set of parameters θ , and $\hat{M}$ is some observable.

Such quantum circuit is constructed from L layers, each consisting of a data encoding circuit block $S(x)$ and a trainable circuit block $W(\theta)$. The data encoding block is the same in every layer and consists of single or multiple qubit gates of the form $\mathcal{G}(x) = e^{-ixH}$, where H is a Hamiltonian that generates the “time evolution” used to encode the data.

Since we want to focus on the role of the data encoding, and to avoid further assumptions on how the trainable circuit blocks are parametrized, we view the trainable circuit blocks as arbitrary unitary operations, $W(\theta) = W$, and drop the subscript θ of f_θ from here on. With this assumption, the overall quantum circuit has the form:

$$U(x) = W^{(L+1)}S(x)W^{(L)} \dots W^{(2)}S(x)W^{(1)}. \quad (3.52)$$

Our goal is to prove that $f(x)$ can be written as a partial Fourier series:

$$f(x) = \sum_{n \in \Omega} c_n e^{inx}, \quad (3.53)$$

with integer-valued frequencies. The first step is to note that one can always find an eigenvalue decomposition of the generator Hamiltonian

$$H = V^\dagger \Sigma V \quad (3.54)$$

where Σ is a diagonal operator that contains the eigenvalues of H , $\lambda_1, \dots, \lambda_d$ on its diagonal. Now for simplicity, we consider the case where $S(x) = \mathcal{G}(x)$, although it can easily be generalized to the more general case where $S(x) = \mathcal{G}_1(x) \otimes \dots \otimes \mathcal{G}_N(x)$. In this case, the data encoding unitary becomes

$$S(x) = V^\dagger e^{-ix\Sigma} V, \quad (3.55)$$

and we can “absorb” V and $V^\dagger$ into the arbitrary unitaries $W' = VWV^\dagger$. Hence, without loss of generality, we will assume that H is diagonal. This allows us to separate the data-dependent expressions from the rest of the circuit in each component i of the quantum state $U(x)|0\rangle$,

$$[U(x)|0\rangle]_i = \sum_{j_1, \dots, j_L=1}^d e^{-i(\lambda_{j_1} + \dots + \lambda_{j_L})x} W_{ij_L}^{(L+1)} \dots W_{j_2 j_1}^{(2)} W_{j_1 1}^{(1)}. \quad (3.56)$$

To ease the notation, we introduce the multi-index $j = \{j_1, \dots, j_L\} \in [d]^L$, where $[d]^L$ refers to the set of L integers between $1, \dots, d$. We then rewrite the sum of eigenvalues for a given j as $\Lambda_j = \lambda_{j_1} + \dots + \lambda_{j_L}$

$$[U(x)|0\rangle]_i = \sum_{j \in [d]^L} e^{-i\Lambda_j x} W_{ij_L}^{(L+1)} \dots W_{j_2 j_1}^{(2)} W_{j_1 1}^{(1)}. \quad (3.57)$$

Now, computing the adjoint of this expression we rewrite the Eq. (3.51) that describes the quantum model:

$$f(x) = \sum_{k, j \in [d]^L} e^{i(\Lambda_k - \Lambda_j)x} a_{k, j}, \quad (3.58)$$

where $a_{k, j}$ contain the terms coming from the parametrized unitaries and the observable,

$$a_{k, j} = \sum_{i, i'} (W^*)_{1k_1}^{(1)} (W^*)_{j_1 j_2}^{(2)} \dots (W^*)_{j_L i}^{(L+1)} M_{i, i'} W_{i' j_L}^{(L+1)} \dots W_{j_2 j_1}^{(2)} W_{j_1 1}^{(1)}. \quad (3.59)$$

At this point Eq. (3.58) almost look like a Fourier series. What is left now is to put together all the terms in Eq. (3.58) whose basis function $e^{(\Lambda_k - \Lambda_j)x}$ have the same frequency $\omega = \Lambda_k - \Lambda_j$. All frequencies accessible to the quantum model are contained in its frequency spectrum

$$\Omega = \{\Lambda_k - \Lambda_j, \quad k, j \in [d]\}. \quad (3.60)$$

With that, we finally obtain

$$f(x) = \sum_{\omega \in \Omega} c_{\omega} e^{i\omega x}, \quad (3.61)$$

where the coefficients are obtained by summing over all $a_{k,j}$ contributing to the same frequency

$$c_{\omega} = \sum_{\substack{k,j \in [d]^L \\ \Lambda_k - \Lambda_j = \omega}} a_{k,j}. \quad (3.62)$$

The frequency spectrum Ω has key properties: $0 \in \Omega$, for every $\omega \in \Omega$, $-\omega \in \Omega$, and since $c_{\omega} = c_{-\omega}^*$, Eq. (3.51) is a real-valued function. The size of the spectrum is denoted by $K = (|\Omega| - 1)/2$, indicating the number of independent nonzero frequencies and the largest frequency $D = \max(\Omega)$ is the spectrum's degree. The coefficients c_{ω} are determined by gates $W^{(1)} \dots W^{(L+1)}$ and the measurement observable $\hat{M}$. Thus, the frequency spectrum depends on the eigenvalues of the data encoding gates, while Fourier coefficients depend on the entire quantum circuit. For integer-valued eigenvalues $\lambda_1, \dots, \lambda_d$, frequencies in Ω are integer-valued, yielding a partial Fourier series as in Eq. (3.53).

The implications of interpreting QNNs as Fourier series will be explored further in Section 5, where we demonstrate how to utilize this property to build a Quantum Monte Carlo integrator.

3.4.4 Quantum Generative Models

Generative models have emerged as a transformative tool to revolutionize how we handle information in the XXI century by creating new data that mimic real-world distributions [89, 90]. These models have significantly advanced artificial intelligence by enabling the creation of datasets that are often indistinguishable from real data. Their applications range from generating text, images, videos or music. Large Language Models (LLMs), such as GPT are a good example of generative models that leveraging transformer architectures, are capable of capturing complex patterns in language and reproducing it to a disturbing level of accuracy.

In this regard, quantum generative models aim to leverage the strengths of quantum computing to enhance data generation tasks [91, 92]. Generally, these models learn a target distribution from training data and generate new samples upon measurement. This aligns well with one of the tasks that quantum computers are better at: sampling. In fact, one of the few demonstrations of quantum supremacy, where a quantum device performs a task faster than any classical counterpart, was Google's 2019 experiment [93], which was fundamentally a sampling problem, showcasing the potential of quantum computers to outperform classical systems. Given that generative modeling is a cornerstone of classical ML and sampling is one of the strongest capabilities of quantum computers, the intersection of these fields has gained significant attention within the quantum computing community.

Several quantum generative models have shown particular promise, following a general workflow based on variational quantum circuits. These circuits contain tunable parameters that are optimized to minimize a cost function that encodes the problem that we aim to

solve. Typically, this cost function is designed to ensure that the output distribution obtained from measuring the quantum circuit matches the target distribution used for training. By exploring an exponentially larger Hilbert space, these models offer a different approach to generative learning, potentially leading to more expressive and efficient representations compared to classical methods.

There exist different type of learning strategies that allows us to differentiate into different groups of quantum generative models. Two relevant examples are Quantum Circuit Born Machines (QCBMs) and Quantum Generative Adversarial Networks (QGANs), each employing distinct mechanisms for training and sample generation.

QCBMs aim to directly learn a target probability distribution. They achieve this by optimizing a cost function, such as the negative log-likelihood, to fit the probability of measurement outcomes from a quantum circuit to the given data distribution. Given a dataset $\mathcal{D} = \{x\}$ consisting of independent and identically distributed samples from an unknown target distribution $\pi(x)$, a QCBM is employed to generate new samples that resemble the target distribution. The QCBM maps an initial product state $|0\rangle$ to a parameterized quantum state $|\psi_\theta\rangle$. Measuring this state in the computational basis produces a sample of bits, following the distribution $x \sim p_\theta(x)$ [94].

$$p_\theta(\mathbf{x}) = |\langle x | \psi_\theta \rangle|^2. \quad (3.63)$$

The goal is to optimize the model so that the generated probability distribution p_θ approximates the target distribution π as closely as possible. A general workflow of the QCBM is depicted in Fig. 3.18

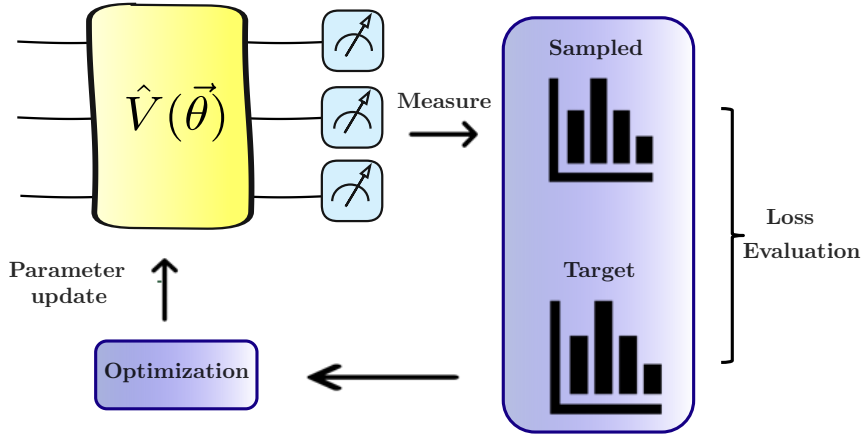


Figure 3.18: Quantum Circuit Born Machine (QCBM) workflow.

On the other hand, QGANs are especially good at generating data with multiple distinct patterns that follow different behaviors. This is due to its learning structure, which consists of two pieces:

1. A generator (quantum) creates fake data samples, trying to make them look real.
2. A discriminator (usually classical) checks whether a sample is real (from actual data) or fake (generated).

These two components compete, which helps the generator learn to focus on different patterns in the data rather than just averaging everything out, like sometimes other models such as QCBM tend to do. This makes QGANs optimal for capturing complex, multi-modal

distributions, meaning they can generate diverse and structured data instead of blurry or unrealistic results. This ability to separate different modes of data makes them very useful for tasks like realistic image synthesis, financial modeling, and anomaly detection. A general workflow of the QGAN is depicted in Fig. 3.19

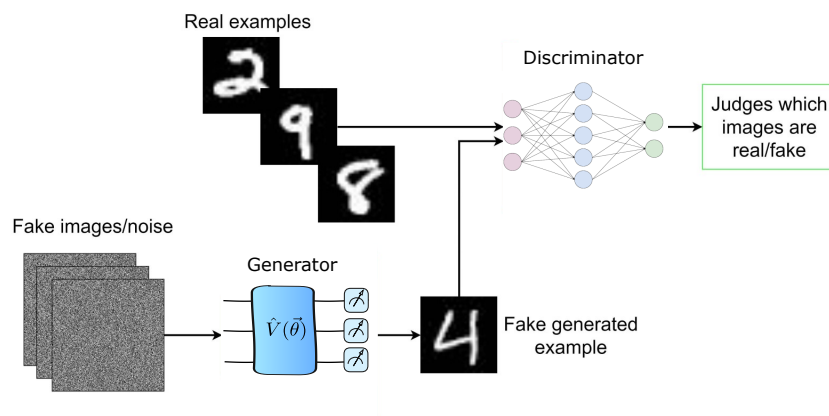


Figure 3.19: Quantum Generative Adversarial Network (QGAN) workflow.

4 Quantum clustering and jet reconstruction at the LHC

4.1 Introduction

Quantum computing devices, based on the principles of Quantum Mechanics, offer the potential to solve certain problems efficiently that become intractable for classical computers due to exponential or super-polynomial scaling. Quantum algorithms leverage superposition, entanglement and interference to achieve a computational advantage over their classical counterparts. In recent years, these algorithms have rapidly advanced, finding applications in diverse fields such as optimization problems in fintech [95], quantum chemistry [96], nuclear physics and Monte Carlo simulations [97, 98, 99], combinatorial optimization [100], and state diagonalization [101, 102]. More recently, they have also been applied to challenges in high-energy physics (HEP) [7, 103]. The Large Hadron Collider (LHC) at CERN, along with its Run 3 and the future high-luminosity phase (HL-LHC), generates vast amounts of data requiring intensive analysis and precise theoretical predictions [104]. The computational demands will only grow with planned future colliders [34, 39, 41].

Quantum algorithms have been explored for various HEP subfields, including jet clustering [105, 106, 107], jet quenching [108], parton density determination [109], parton shower simulation [110, 111, 112], heavy-ion collisions [113], quantum machine learning [114, 115, 116, 117, 118, 119, 120, 121], lattice gauge theories [122, 123, 124, 125], and multi-loop Feynman integrals [126, 127, 128, 129, 2, 3, 9].

In this chapter, we focus on the problem of clustering and jet reconstruction from collider data, a computationally expensive task involving optimization over large sets of final-state particles. Current state-of-the-art jet clustering algorithms require months to process one year’s worth of LHC data [130]. The upcoming HL-LHC will further increase event rates and pile-up [131, 132], increasing the computational challenge. Without improvements, processing times could last to several decades. This evidences the need for faster and more efficient jet clustering algorithms.

In this chapter, we explore the potential of quantum algorithms to accelerate jet identification, focusing on three well-established classical methods: **K-means** clustering [133, 134], Affinity Propagation (AP) [135], and the k_T -jet clustering algorithm in its various forms [136, 137, 138, 139, 140]. In particular, we propose [6, 8] quantum analogs for each: quantum **K-means**, quantum AP, and quantum k_T -based algorithms.

Clustering is a fundamental problem in ML and computational geometry, widely used in marketing, data mining, bioinformatics, image processing, and pattern recognition, as well as in HEP. The **K-means** method [134, 133] groups N datapoints into K clusters, where

each point belongs to the nearest mean. Finding the optimal solution is NP-hard [141], but the widely used Lloyd’s algorithm [142] provides an iterative heuristic approach. The K-means++ variant [143] improves performance by selecting better initial centers. In HEP, K-means has been applied to improve mass resolution in top quark and W boson studies [144], boosted object tagging [145], and minijet identification at low p_T [146].

The AP algorithm clusters data by exchanging messages between points to identify exemplars. It has been applied in face recognition, gene identification [147, 148, 149], astrophysics [150], and parton density studies in HEP [151]. Unlike K-means, it does not require to specify the number of clusters beforehand, making it interesting for jet clustering.

Hierarchical clustering builds on a recursive approach without fixing the number of clusters in advance. The k_T -based algorithms [152] fall into this category, relying on a distance metric to iteratively merge clusters. These methods are the most widely used for jet reconstruction at the LHC.

Quantum versions of K-means clustering have been introduced in Refs. [153, 106] for HEP, following earlier studies [154]. These implementations use Euclidean distance for clustering. In this chapter, we present a novel quantum K-means algorithm based on Minkowski distance, making it the first quantum clustering method adapted to relativistic kinematics. For the quantum AP algorithm, we define a similarity metric based on invariant sum squared and compute it using a quantum subroutine identical to quantum K-means. Additionally, we introduce the first quantum k_T -based algorithm, featuring a novel quantum search method for finding minimum distances. This new quantum method is generalizable to other applications, which we discuss further in the chapter.

This chapter is organized as follows. In Section 4.4.1.2 we introduce our notation and we define the Euclidean and Minkowskian quantum distances. In Section 4.4.2 we present our new quantum algorithm in order to search the approximate maximum in a set of a given number of elements. We consider the classical versions of the K-means clustering, AP and k_T -based algorithms in Section 4.3. In Section 4.4 we present our results considering the quantum simulations of these algorithms and a proof-of-concept implementation with Gaussian datasets as well as with simulated LHC physical events. We also compare their performance in detail. We discuss their differences and conceptual similarities and we compare them with their classical counterparts. A brief summary of our results is presented in Section 4.6.

The code to reproduce the results of this chapter is available in [155].

4.2 Jets

The main goal of particle colliders such as the Large Hadron Collider (LHC) at CERN is to study the fundamental interactions of elementary particles with the greatest detail and to search for new physics, including the discovery of unknown particles. Among the richest processes that include more relevant information are those involving quarks and gluons, collectively known as partons, as they constitute other particles known as hadrons that are color-neutral states. When partons are produced in high-energy collisions, they undergo a cascade of radiation, emitting secondary quarks and gluons mostly collinear to the parent parton through a process called parton showering. This is a prediction of perturbative QCD at high energies. Because quarks and gluons cannot exist as free particles due to QCD’s color confinement they finally hadronize into color-neutral bound states by combining with “sea” partons spontaneously created from the quantum vacuum. The avalanche of hadrons

created is observed in the detector as highly collimated bunches of particles see Fig. 4.1. These bunches of particles are known as hadronic jets, as their constituents are hadrons. If we look inside a jet we might find protons and neutrons, the constituents of atomic nuclei, and other less known hadrons such as pions and kaons [156, 157].

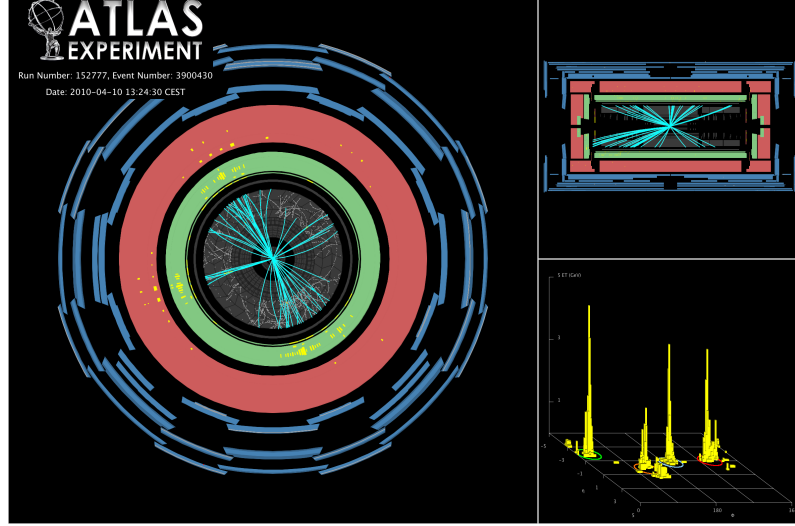


Figure 4.1: Event with four reconstructed hadronic jets observed by the ATLAS detector at CERN. Picture taken from the ATLAS public event display repository [158]

At this point, it has been clear that the partons produced in the collisions cannot be directly observed since they are not free particles. However, jets provide a relevant information on the underlying quarks and gluons produced in high-energy collisions. Analyzing individual hadrons would obscure the original parton dynamics, as hadronization is a non-perturbative process that introduces significant complexity. On the other hand, jets preserve key properties of the originating parton due to the preference for the production of secondary particles in the original direction, as predicted by QCD. This allows for meaningful comparisons with QCD predictions and facilitates searches for new physics, where deviations from expected jet properties could indicate the presence of beyond the Standard Model (BSM) physics [159].

Jets are fundamental objects in HEP analysis [159, 160, 161, 139]. The transverse momentum (p_T) of jets, measured perpendicular to the beam axis, distinguishes hard-scattering processes (high p_T) from background interactions (low p_T). Also the jet mass, calculated as the invariant mass of all constituent particles, helps differentiate jets produced by quarks (typically lighter) from those produced by gluons or boosted jets. Furthermore, the angular substructure, reveals radiation patterns critical for identifying QCD splittings or resonant decays. Jets originated from bottom quarks are specifically tagged via displaced secondary vertices. Finally, the jet radius parameter (R), defining the angular scale $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ for particle clustering, balances resolution and noise: smaller radius ($R \leq 0.4$) mitigates pileup in proton-proton collisions, while larger radius R ($R \sim 1.0$) captures the broad substructure of boosted decays. Together, these properties enable jets to serve as effective tools to identify and describe the partons that participate in the hard scattering at very short distances.

Accurately identifying jets in high-energy collisions is a daunting task that is essential for understanding the fundamental processes at particle colliders. Since jets are the primary experimental signatures of quarks and gluons, their correct reconstruction directly impacts precision tests of QCD and searches for new physics. A well-designed jet identification

algorithm must efficiently group the particles originating from a parent parton while minimizing contamination from unrelated radiation or detector noise. However, this is challenging due to the complex and dynamic nature of hadronization, the presence of overlapping jets, and fluctuations in the detector response. Poor jet reconstruction can lead to noisy measurements of key observables, such as jet energy and angular distributions, reducing the sensitivity to deviations from the SM. In searches for new physics, subtle modifications in jet substructure could provide hints of novel phenomena, making robust jet algorithms a critical component of high-energy collider experiments.

4.3 Clustering algorithms

4.3.1 K-means algorithm

The **K-means** algorithm is an unsupervised learning method that classifies the elements of a dataset into K groups called clusters [134, 133]. The goal of this algorithm is to maximize similarity within clusters while maximizing dissimilarity between them. The algorithm inputs and outputs are:

- Input: N d -dimensional data points and cluster count K ($K \leq N$)
- Output: K centroids computed as cluster means, and the N points assigned to their respective centroids.

The iterative procedure follows these steps, and is also depicted in Fig. 4.2:

1. **Initialization:** Select K initial centroids, either randomly or via smart initialization, such as **k-means++** [143].
2. **Assignment:** Associate each point to its nearest centroid using a predefined distance metric, such as the Euclidean distance.
3. **Update:** Recompute centroids as the mean of all points in each cluster.
4. **Termination:** Repeat steps 2 and 3 until centroids stabilize and convergence is achieved.

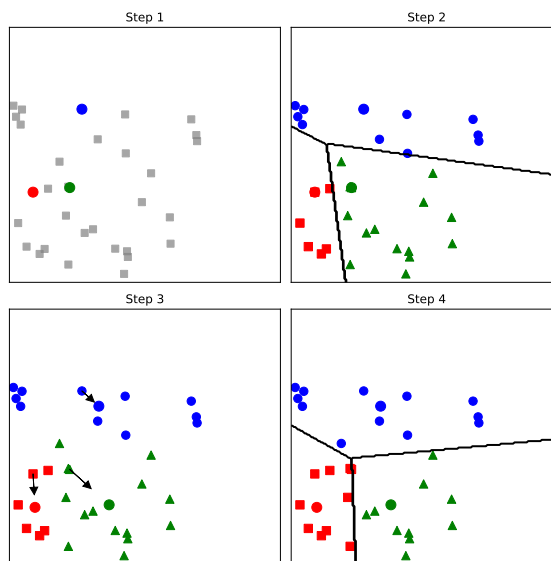


Figure 4.2: Workflow of the K-means algorithm.

The time complexity of this algorithm is estimated by analysing the time complexity of its components. The number of distances that have to be calculated required $\mathcal{O}(N)$ operations, the search of a minimum distance for every data point with respect to the centroids would be $\mathcal{O}(K)$, and the calculation of each distance itself would require $\mathcal{O}(d)$, this results in a complexity of $\mathcal{O}(NKd)$.

4.3.2 Affinity Propagation algorithm

Although K-means is an effective algorithm for clustering data, it requires the number of clusters, K , to be predetermined, which is often not practical in HEP applications. In contrast, the Affinity Propagation (AP) algorithm [135], another unsupervised machine learning technique, does not require the number of clusters as an input. Instead, AP only requires the data points that need to be classified. Let $x_1, \dots, x_N$ represent a set of data points. A similarity function s is then computed to quantify the relationship between points, such that $s(i, j) \geq s(i, k)$ if and only if x_i is more similar to x_j than to x_k . A common similarity metric is the negative squared distance between two points: $s(i, j) = -|x_i - x_j|^2$.

The diagonal elements of the similarity matrix, $s(i, i)$, are particularly important as they represent “preferences” that indicate the likelihood of a point becoming an exemplar, or a cluster center, similar to K-means centroids. During the first iteration, each diagonal element, $s(i, i)$, is initialized to a fixed value, typically the median similarity of all point pairs.

Next, two matrices are computed, representing message exchanges between data points [135]. The responsibility matrix R contains values $r(i, k)$, which measure the suitability of point k as the exemplar for point i , compared to other candidate exemplars. The availability matrix A contains elements $a(i, k)$, which indicate how suitable it would be for point i to select point k as its exemplar, based on the preferences of other points. Both matrices can be interpreted as log-probability ratios. The steps of the AP algorithm are as follows and are also depicted in Fig. 4.3:

1. The matrices R and A are initialized to zero.
2. The responsibility matrix is computed:

$$r(i, k) = s(i, k) - \max_{q \neq k} \{a(i, q) + s(i, q)\}. \quad (4.1)$$

3. The availability matrix is computed:

$$a(i, k) = \min \left(0, r(k, k) + \sum_{q \notin \{i, k\}} \max(0, r(q, i)) \right) \text{ for } i \neq k, \text{ and} \quad (4.2)$$

$$a(k, k) = \sum_{q \neq k} \max(0, r(q, k)). \quad (4.3)$$

4. Steps 2 and 3 are repeated until either the cluster boundaries remain unchanged for several iterations, or a predetermined number (of iterations) is reached.

After convergence, the exemplars (clusters) are identified from the final matrices as those for which $r(i, i) + a(i, i) > 0$.

This algorithm requires $\mathcal{O}(N^2)$ operations to fill the similarity matrix, and computing each element takes $\mathcal{O}(d)$ time, as it involves calculating the distance between two d -dimensional

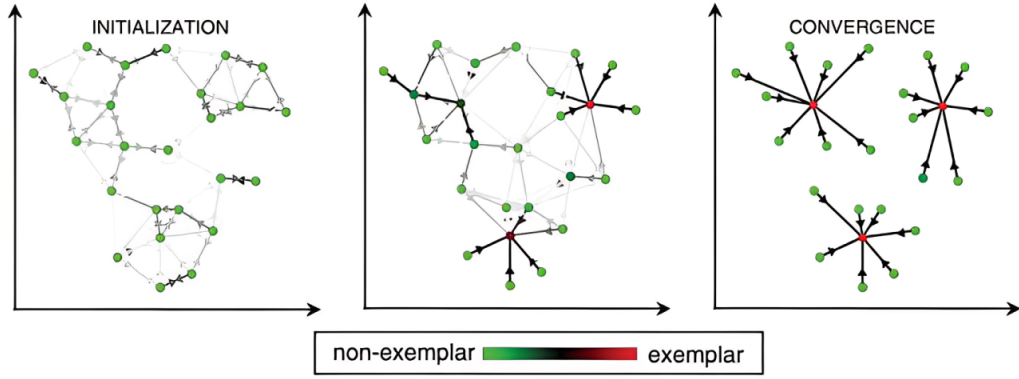


Figure 4.3: Workflow of the AP algorithm. Picture taken from [162]

points. Furthermore, steps 2 and 3 are repeated T times, resulting in an overall complexity of $\mathcal{O}(N^2 T d)$.

4.3.3 Generalised k_T -jet algorithm

The inclusive variant of the generalised k_T -jet algorithm is defined as follows [152]:

1. For each pair of particles i and j , a distance measure is calculated according to:

$$d_{ij} = \min(p_{T,i}^{2p}, p_{T,j}^{2p}) \Delta R_{ij}^2 / R^2, \quad (4.4)$$

where $\Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$. Here, $p_{T,i}$, y_i , and ϕ_i are the transverse momentum (relative to the beam direction), rapidity, and azimuth angle of particle i , respectively. The parameter R defines the characteristic jet radius and is typically set to a value around 1. In addition, a distance to the beam is defined for each particle i as $d_{iB} = p_{T,i}^{2p}$.

2. The smallest distance, d_{min} , is then identified among all d_{ij} and d_{iB} . If d_{min} corresponds to a d_{ij} , particles i and j are merged by summing their four-momenta (using the E-scheme recombination). Alternatively, if d_{min} is a d_{iB} , particle i is designated as a final jet and removed from further consideration.
3. Step 1 is repeated until no particles remain.

It is worth noting that for specific choices of the parameter p in Eq. (4.4), the generalised k_T -jet algorithm simplifies to well-known jet algorithms: the k_T algorithm for $p = 1$, the Cambridge/Aachen algorithm for $p = 0$, and the anti- k_T algorithm for $p = -1$.

As mentioned in Ref. [163], the traditional implementation of the k_T -jet algorithm has a computational complexity of $\mathcal{O}(N^3)$. This stems from the fact that the algorithm's performance is limited by the need to scan a table of size $\mathcal{O}(N^2)$ containing all distances d_{ij} and d_{iB} , an operation that must be performed N times.

However, the **FastJet** [152] algorithm achieves a reduced complexity of $\mathcal{O}(N^2)$. This improvement relies on identifying the geometrical nearest neighbor for each particle, which avoids the need to construct the full size- N^2 table of d_{ij} . Instead, it is sufficient to consider only the size- N array $d_{i\mathcal{G}_i}$, where $\mathcal{G}_i$ represents i 's geometrical nearest neighbor. Furthermore, **FastJet** offers opportunities for additional optimization using Voronoi diagrams, leading to a further reduction in time complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$.

4.4 Quantum algorithms

In this section we design specific quantum algorithms intended to be inserted as subroutines in the clustering algorithms mentioned in the previous section with the goal of speeding up some bottleneck tasks.

4.4.1 Quantum distance

In quantum computing, it is crucial to have efficient tools to measure quantum overlap between two states, as in many cases it determines the possibility of obtaining a quantum advantage [Foulds:2021]. One approach to efficiently measure similarity between quantum states is the *SwapTest* method [164] (see Eq. (A.5) in Appendix A for more details). The quantum distances (Euclidean distance or Minkowski invariant sum squared) presented in this subsection, make use of the *SwapTest* procedure.

4.4.1.1 Euclidean quantum distance

Let us start by considering N data points or vectors in an Euclidean d -dimensional space, $\{\mathbf{x}_i\}_{i=1,\dots,N}$, which are encoded as quantum states of the form

$$|x_i\rangle = |\mathbf{x}_i|^{-1} \sum_{\mu=1}^d x_{i,\mu} |\mu\rangle, \quad (4.5)$$

where $|\mathbf{x}_i| = \sqrt{\sum_{\mu=1}^d (x_{i,\mu})^2}$ is the modulus of the vector $\mathbf{x}_i$, and $x_{i,\mu}$ are its components. Since we rely on amplitude encoding, each vector requires $n \geq \log_2 d$ qubits to be encoded, i.e. for $d = 3$ we need two entangled qubits where one of its states remains free and is not used. The Euclidean distance between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined classically as

$$d_E^{(C)}(\mathbf{x}_i, \mathbf{x}_j) = |\mathbf{x}_i - \mathbf{x}_j|, \quad (4.6)$$

where E is a subscript indicating Euclidean distance, and the superscript C specifies that it is the classical version.

To obtain the quantum counterpart of Eq. (4.6), we employ the controlled *SwapTest* approach. In order to define the Euclidean quantum distance connecting the d -dimensional vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, we entangle their corresponding quantum states, $|x_i\rangle$ and $|x_j\rangle$. Auxiliary states are then defined as

$$|\psi_1\rangle = \frac{1}{\sqrt{2}} (|0, x_i\rangle + |1, x_j\rangle), \quad |\psi_2\rangle = \frac{1}{\sqrt{Z_{ij}}} (|\mathbf{x}_i||0\rangle - |\mathbf{x}_j||1\rangle), \quad (4.7)$$

where $Z_{ij} = |\mathbf{x}_i|^2 + |\mathbf{x}_j|^2$ acts as a normalization constant, and $|0\rangle$ and $|1\rangle$ are the states of an auxiliary qubit. Furthermore, let us define the swapped state $|\psi'_1\rangle$ as

$$|\psi'_1\rangle = \frac{1}{\sqrt{2}} (|x_i, 0\rangle + |x_j, 1\rangle). \quad (4.8)$$

The inner products involving the quantum states defined in Eqs. (8.2) and (4.8) is expressed as

$$\langle\psi'_1|\psi_2\rangle = \frac{1}{\sqrt{2Z_{ij}}} (|\mathbf{x}_i|\langle x_i| - |\mathbf{x}_j|\langle x_j|), \quad \langle\psi_2|\psi_1\rangle = \frac{1}{\sqrt{2Z_{ij}}} (|\mathbf{x}_i||x_i\rangle - |\mathbf{x}_j||x_j\rangle). \quad (4.9)$$

From this, it follows that

$$\langle \psi'_1 | \psi_2 \rangle \langle \psi_2 | \psi_1 \rangle = \frac{1}{2Z_{ij}} |\mathbf{x}_i - \mathbf{x}_j|^2. \quad (4.10)$$

Thus, the quantum Euclidean distance (as detailed in Eq. (A.5) within Appendix A) is given by

$$d_E^{(Q)}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{2Z_{ij}(2P_{\Psi_3}(|0\rangle) - 1)}, \quad (4.11)$$

where the superscript Q indicates the quantum formulation of the distance d_E , and the subscript Ψ_3 of the probability P refers to the probability of measuring the ancillary qubit in state $|0\rangle$ at the last stage of the *SwapTest* method.

4.4.1.2 Quantum invariant sum squared in Minkowski space

In HEP, vectors are typically defined within a four-dimensional spacetime in the Minkowski metric. These vectors take the form $x_i = (x_{i,0}, \mathbf{x}_i)$, where $x_{i,0}$ is the temporal component and $\mathbf{x}_i$ represent the three spatial components. We will subsequently assume a spacetime dimension of d , with $d - 1$ indicating the number of spatial components. The analogue of the Euclidean classical distance, in the Minkowski space is the invariant sum squared $s_{ij}^{(C)}$, often known as the invariant mass squared for particle four-momenta:

$$s_{ij}^{(C)} = (x_{0,i} + x_{0,j})^2 - |\mathbf{x}_i + \mathbf{x}_j|^2. \quad (4.12)$$

This Lorentz-invariant quantity serves as a similarity test. It also corresponds to the distance employed in certain traditional jet-clustering algorithms at e^+e^- colliders [165, 166, 167]. Computing this Minkowski-type distance via a quantum algorithm necessitates two applications of the *SwapTest* subroutine (detailed in Appendix A): one for the spatial and one for the temporal components.

The spatial distance is calculated following the method outlined in the previous section, with a minor adjustment to Eq. (4.9) (a sign change in the term multiplying the qubit $|1\rangle$):

$$|\psi_2\rangle \longrightarrow |\psi_2\rangle = \frac{1}{\sqrt{Z_{ij}}} (|\mathbf{x}_i||0\rangle + |\mathbf{x}_j||1\rangle), \quad (4.13)$$

The temporal distance is obtained by calculating the overlap between these states:

$$|\varphi_1\rangle = H|0\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle), \quad |\varphi_2\rangle = \frac{1}{\sqrt{Z_0}} (x_{0,i}|0\rangle + x_{0,j}|1\rangle), \quad (4.14)$$

where $Z_0 = x_{0,i}^2 + x_{0,j}^2$. Applying the *SwapTest* to these states results in:

$$P(|0\rangle|_{time}) = \frac{1}{2} + \frac{1}{2} |\langle \varphi_1 | \varphi_2 \rangle|^2, \quad (4.15)$$

where the overlap $|\langle \varphi_1 | \varphi_2 \rangle|^2$ is trivially given by

$$|\langle \varphi_1 | \varphi_2 \rangle|^2 = \frac{1}{2Z_0} (x_{0,i} + x_{0,j})^2. \quad (4.16)$$

Therefore:

$$(x_{0,i} + x_{0,j})^2 = 2Z_0(2P_{\Psi_3}(|0\rangle|_{time}) - 1). \quad (4.17)$$

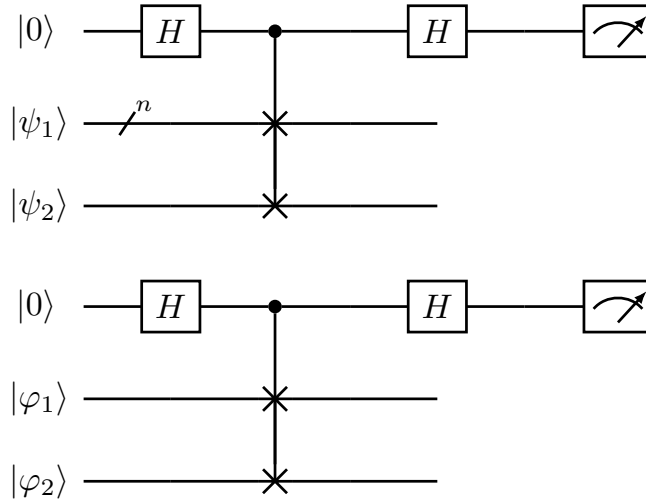


Figure 4.4: Quantum circuit to obtain the invariant sum squared between two d -dimensional vectors in Minkowski space.

At this juncture, the quantum version of the invariant sum squared emerges from the combination of Eqs. (4.11) and (4.17):

$$s_{ij}^{(Q)} = 2(Z_0(2P_{\Psi_3}(|0\rangle|_{time}) - 1) - Z_{ij}(2P_{\Psi_3}(|0\rangle|_{spatial}) - 1)). \quad (4.18)$$

The quantum circuit used to implement the invariant sum-squared distance is presented in Fig. 4.4. For the first three wires, the *SwapTest* is executed on the spatial components, where we assume that the states ψ_1, ψ_2 have been loaded from a quantum Random Access Memory (qRAM) in $\mathcal{O}(\log(d-1))$, since the state ψ_1 is encoded in $\log_2(d-1)$ qubits. On the other hand, from the fourth wire onward, the *SwapTest* is applied to the temporal components. In this case, it takes $\mathcal{O}(1)$, since we only have 1-dimensional qubit states.

4.4.2 Quantum maximum searching algorithm

Finding a specific element within a dataset is a common and computationally demanding challenge in data analysis. However, quantum computing provides efficient tools to accelerate data queries. In particular, Grover's algorithm [168] is well known for achieving a quadratic speedup.

In this section, we introduce a significantly simpler algorithm designed only for identifying the maximum value in a list. Despite its simplicity, this algorithm is sufficiently precise for the applications that we will address in Section 4.4. To the best of our knowledge, this method has not been previously reported in the literature.

Let $L[0, \dots, N-1]$ be an unsorted list of N elements. The problem of finding the maximum requires determining the index y such that $L[y]$ is the largest value. The quantum algorithm leveraging amplitude encoding follows two steps:

1. The list of N elements is encoded into a $\log(N)$ -qubit state as follows:

$$|\Psi\rangle = \frac{1}{\sqrt{L_{sum}}} \sum_{j=0}^{N-1} L[j] |j\rangle, \quad (4.19)$$

where $L_{sum} = \sum_{j=0}^{N-1} L[j]^2$ is a normalization constant. This amplitude encoding is implemented using qRAM.

2. The final state is measured multiple times to mitigate statistical uncertainty. The most frequently observed state corresponds to the maximum value.

The quantum circuits that represents this algorithm is shown in Fig. 4.5, where $n = \log(N)$ qubits are needed to encode a list of N (real) elements.

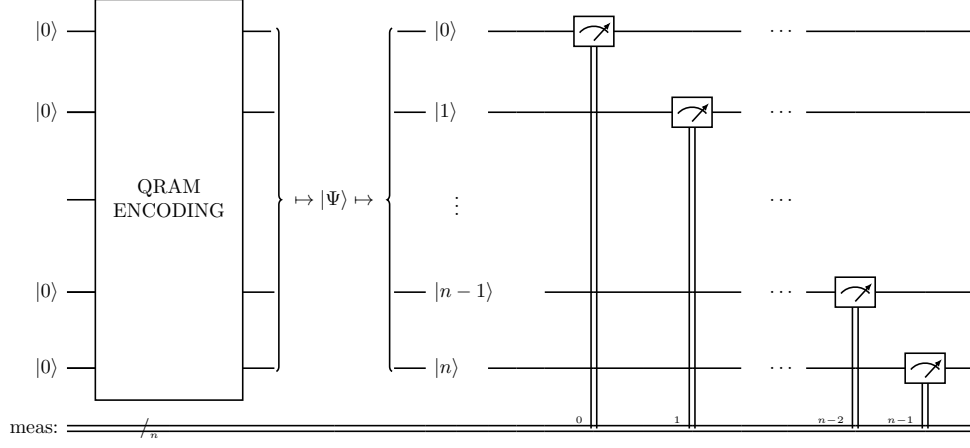


Figure 4.5: Quantum circuit for quantum maximum searching by qRAM amplitude encoding.

The bottleneck of this procedure underlies in encoding data into a quantum state. Assuming data is stored in a qRAM, as would be the case on a true universal quantum computer, encoding takes $\mathcal{O}(\log(N))$ steps [169, 170, 171, 172, 173, 174, 175, 176]. The best classical algorithm used to obtain the minimum of an unsorted list of N items is of order $\mathcal{O}(N)$. Therefore, under these assumptions, the improvement introduced by this quantum algorithm is exponential.

The well-known quantum minimum search algorithm by Dürr and Høyer [53] achieves a complexity of $\mathcal{O}(\sqrt{N})$. Following its theoretical proposal [53], it has been studied and implemented in quantum simulators (see Ref. [177]). However, prior implementations [177] indicate potential areas for improvement, including the large number of qubits required, the infeasibility of hardcoding an oracle for each element, the high number of *shots* needed, and, in some cases, suboptimal performance. The quantum maximum search algorithm presented here, based on amplitude encoding through qRAM, aims to address these challenges and enhance the efficiency of quantum search methods.

Nonetheless, the new algorithm introduced in this paper and the Dürr and Høyer quantum method share a probabilistic nature that could lead to misidentification of the absolute maximum or minimum. These challenges typically arise when the dataset has a very low standard deviation, or when the largest and smallest values are nearly indistinguishable. In such cases, the probability of measuring multiple candidates becomes almost identical, making it difficult to select the correct maximum/minimum.

However, as we will show in Section 4.4, these potential challenges do not significantly affect the practical implementation of our quantum algorithm in the context of jet clustering.

Beyond HEP and jet clustering, our quantum algorithm could also be valuable in other fields, particularly in Extreme Value Theory (EVT)[178]. According to Gumbel (1958)[179], EVT focuses on analyzing the probability distributions of extreme values, aiming to predict rare events based on historical data. Since predictive models require identifying extreme values within large datasets, our algorithm could be well-suited for statistical analysis in this domain. This includes applications in actuarial and financial

sciences, meteorology, material sciences, engineering, climatology, geology, hydrology, and highway traffic analysis [180, 181, 182].

4.5 Quantum clustering algorithms

In this Section, we use the open-source IBM Quantum software *Qiskit* [84] to build the quantum algorithms presented in Section 4.4. Specifically we design the quantum circuits to calculate the invariant sum squared described in Section 4.4.1.2 for the **K-means** and the **AP** algorithm, and to finding the minimum distance in the **K-means** and the k_T -jet algorithm. Then, these quantum subroutines have been introduced into their respective classical algorithm substituting the classical part they are replacing. The algorithms presented here have been executed on a quantum simulator that offers an unrestricted and noise-free environment. A quantum implementation in an existing quantum device taking advantage of the claimed speed-up is today not possible, as a qRAM architecture does not exist yet. Nonetheless, the quantum simulations in this section show a satisfactory performance and clustering efficiencies comparable to those of their classical counterparts.

4.5.1 Quantum K-means with Minkowski-type distance

The **K-means** algorithm has a precedent of a quantum version that differs from its classical counterpart [183] in two points [184]. First, the quantum **K-means** introduces a quantum method to calculate the distance between data points. Second, the quantum version also includes a procedure for obtaining the minimum distance of each data point with respect to the K centroids, which is achieved by Dürr and Høyer’s algorithm [53].

We focus on a new quantum version of the **K-means** algorithm, where the calculation of distances is made quantumly and the minimum distance of each data point to the centroids is obtained with the quantum maximum searching algorithm¹ explained in Section 4.4.2. Other quantum versions of the **K-means** algorithm have been studied in Refs. [153, 106] and [154], where an Euclidean distance was used to separate the particles from each other. Instead, we analyse for the first time an implementation of the **K-means** algorithm with a Minkowski-type quantum distance, as defined in Section 4.4.1.2.

The time complexity of this quantum algorithm is estimated by analysing the time complexity of its components. The distances that have to be calculated are $\mathcal{O}(N)$, the search of a minimum distance for every data point with respect to the centroids would be $\mathcal{O}(\log K)$, and the calculation of each distance itself would require $\mathcal{O}(\log(d-1))$ qubits assuming the data is stored in a qRAM. This results in a speedup from $\mathcal{O}(N K d)$ in the classical version to $\mathcal{O}(N \log K \log(d-1))$ in our quantum version. Therefore an exponential speed-up in the number of clusters and in the vector dimensionality would be achieved.

We now present our implementation of the quantum **K-means** algorithm using the invariant sum squared as a distance metric, along with a maximum searching algorithm, and compare their performance with classical counterparts. To do so, we generate 15 Gaussian-clustered datasets of $N = 300$ three-dimensional vectors² with different noise and clustering levels using *Scikit-learn*’s *make_blobs* function, which provides *true_labels* for validation. These

¹We may apply this algorithm for finding the minimum since obtaining the minimum amongst the distances is equivalent to obtaining the maximum of their inverses: s_{ij}^{-1} .

²In general, it is possible to relate this generated set of three-dimensional vectors, to a physical event at the LHC. It is enough to consider the set of n three-dimensional vectors as massless partons recoiling against a small number of tagged particles.

labels allow us to assess clustering accuracy by computing the true efficiency, ε_t , defined as the fraction of data points correctly classified by the algorithm.

Both the hybrid and classical K-means algorithms were applied to each dataset. The analyzed data represent particle four-momenta, with the three-dimensional vectors corresponding to spatial components, while temporal components are inferred under the assumption that all particles are massless and on-shell. Results are shown in Figs. 4.6 and 4.7.

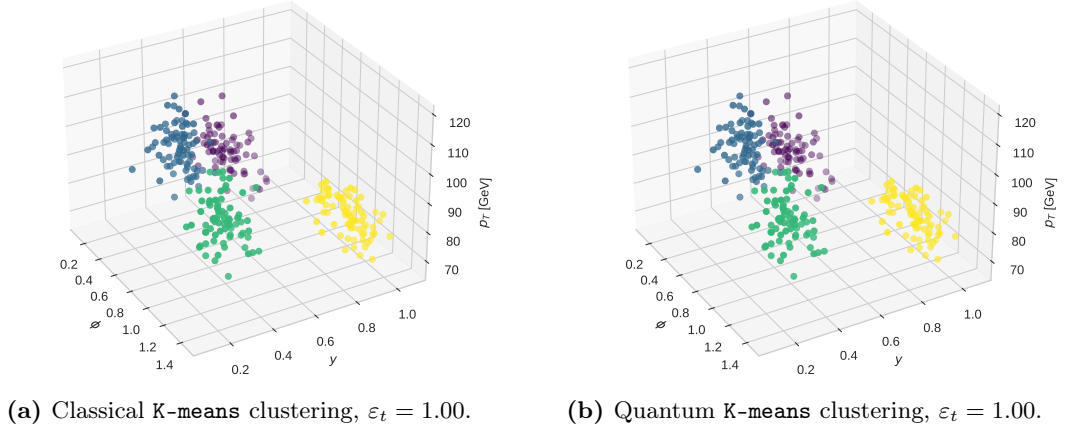


Figure 4.6: Clusters identified after five iterations by the classical and quantum versions of the K-means algorithm are shown in different colors. The dataset consists of Gaussian-distributed points generated with a random seed and a standard deviation of 2.0 from the cluster centroids. Clustering was performed using a Minkowski-type distance, assuming all particles are massless and on-shell, with both algorithms achieving an efficiency of $\varepsilon_t = 1.00$.

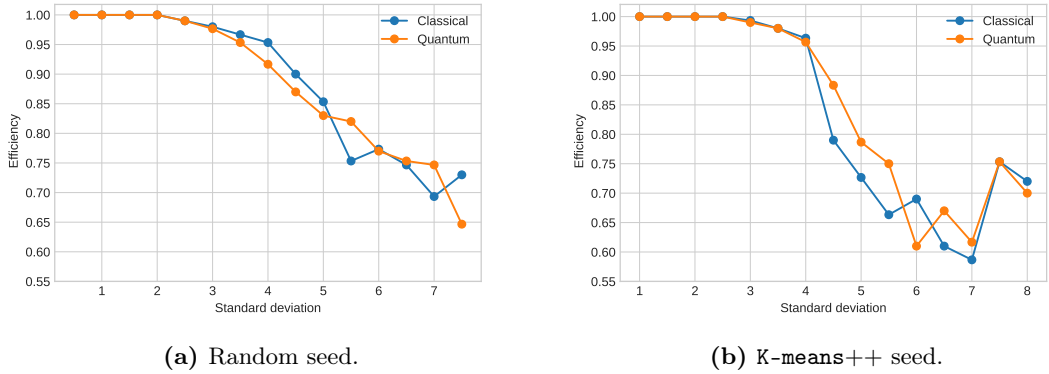


Figure 4.7: Cluster efficiency of the K-means algorithm as a function of the data's standard deviation from the centroids. Both the classical and quantum versions were tested on 15 datasets with standard deviations ranging from 0.5 to 7.5.

Fig. 4.6 illustrates that both the classical and quantum versions of the K-means algorithm cluster data similarly in the three-dimensional space of transverse momentum (p_T), rapidity (y), and azimuthal angle (ϕ). On the other hand, Fig. 4.7 presents the clustering efficiency as a function of the standard deviation used to generate the data, assessing whether clustering occurs as expected. For small standard deviations, both algorithms perform well, achieving efficiencies close to one. However, as the standard deviation increases—introducing more noise—efficiencies decline for both approaches.

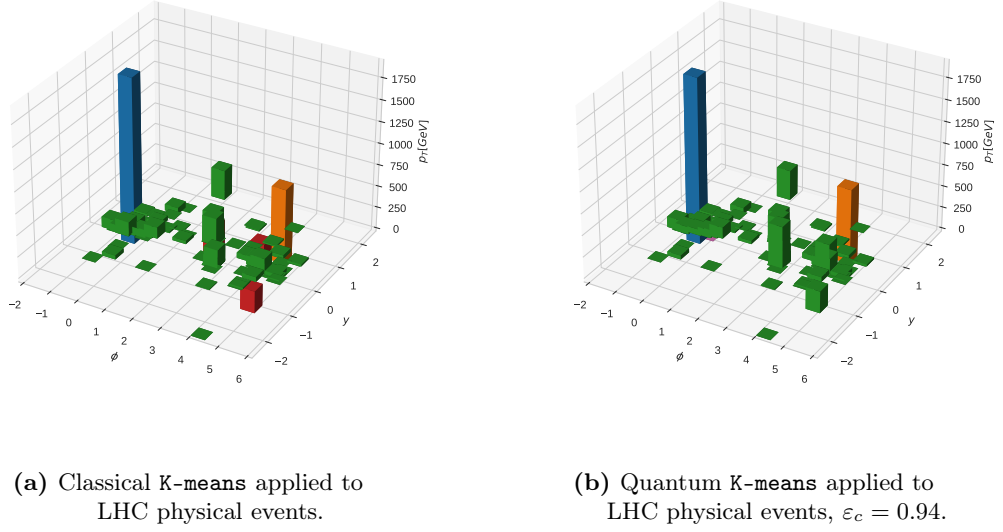


Figure 4.8: A sample parton-level event generated as described in the text and clustered with the classical and the quantum version of the K-means++ algorithm, taking $K = 8$.

The performance of K-means is further studied by comparing two centroid initialization strategies: a random seed (Fig. 4.7a) and the K-means++ method (Fig. 4.7b), which selects initial centroids to be as far as possible. The random seed approach presents a linear efficiency decrease with increasing standard deviation, with classical and quantum versions performing similarly. In contrast, the K-means++ variant shows distinct behavior: from a standard deviation of 4 onward, the quantum version generally outperforms the classical one. In this case, both algorithms experience a drop in efficiency from a standard deviation of 4 to 7, followed by a slight recovery from 7 to 8. Comparing both methods, K-means++ achieves higher efficiencies for low standard deviations (< 4), whereas the random seed approach performs better for larger standard deviations.

We now apply our quantum K-means method to LHC physical events. A key challenge arises from the fact that a negative vector $-\vec{x}$ represents the same quantum state $|x\rangle$ as its positive counterpart $\vec{x}$, differing only by a global phase. To address this, we preprocess the data by rescaling each component of every data point to the range $\{1, 10\}$, ensuring all values remain positive³. Unlike the previous analysis, LHC event data lack *true_labels*, making it impossible to compute ε_t . Instead, we introduce the efficiency metric ε_c , defined as the quotient of the number of particles clustered in the same way as their classical counterpart and the total number of particles to be classified.

We simulate an n -particle event at the LHC using a custom phase-space event generator implemented in C++ and based on ROOT [185]. This generator can handle n -particle events with n reaching tens of thousands, allowing for a flexible choice of final-state configurations, including massless QCD partons, QCD partons with photons, massive vector bosons, or top quarks. The accuracy of each generated event is validated by enforcing kinematical constraints between the initial state and the n -particle final state. The required precision⁴

³The value 0 is excluded to prevent numerical and statistical fluctuations

⁴The kinematical constraint is evaluated over the three-momentum space vector, assuming all momenta are outgoing. The test is performed at the highest multiplicity, ensuring that the precision is at least 10^{-2} , with further improvements as the number of final-state particles decreases.

is consistently better than 10^{-2} .

We study the production of n -particle massless final states in proton-proton⁵ collisions at a center-of-mass energy of $\sqrt{s} = 14$ TeV. The final-state selection follows these criteria: jets are identified using the k_T -jet algorithm with a minimum transverse momentum of $p_{T,\min} \geq 10$ GeV and a radius of $R = 1$. We focus on events with $n = 128$ massless particles in the final state.

The application of the quantum K-means++ algorithm to LHC events is shown in Fig. 4.8. Despite setting $K = 8$ clusters initially, Fig. 4.8 reveals that the algorithm consistently identifies only 3 or 4 distinct clusters (jets). This behavior arises because, although the algorithm starts with K centroids, it converges to a local minimum where some clusters remain empty. Fig. 4.8 also illustrates that both the classical and quantum algorithms classify the data similarly, with the quantum algorithm achieving an efficiency close to one. These results suggest that the quantum approach performs successfully when applied to physical LHC data.

4.5.2 Quantum Affinity Propagation algorithm

The quantum (hybrid) Affinity Propagation (AP) algorithm [135] uses the invariant sum squared as a metric in the similarity matrix and calculates it through a quantum subroutine, as the quantum K-means algorithm just described. Then, a speedup would be achieved, since computing the distances only requires $\mathcal{O}(\log(d-1))$ qubits. So, the quantum AP algorithm, which is as far as we know completely original, would have a time complexity of $\mathcal{O}(N^2 T \log(d-1))$ instead of the classical $\mathcal{O}(N^2 T d)$.

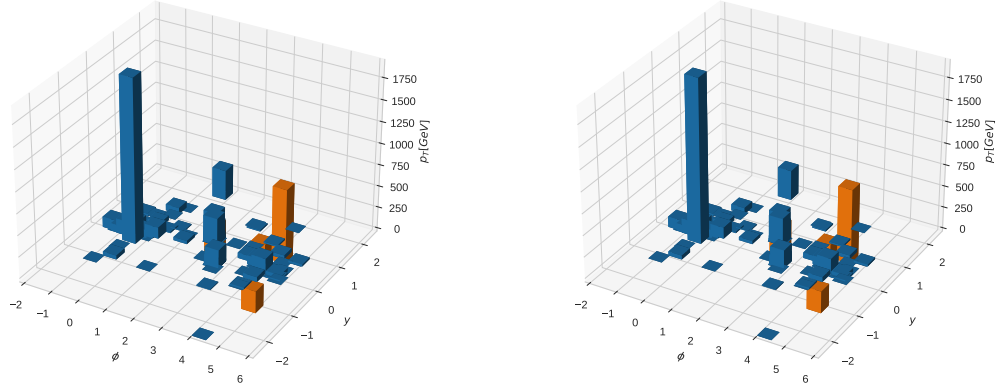
We present a simulation of the quantum AP algorithm. First, we apply the algorithm to Gaussian datasets with varying numbers of clusters, generated using a standard deviation of 0.6. This particular value for the standard deviation was chosen for convenience. The resulting efficiencies for both the classical and quantum versions are shown in Table 4.1. Table 4.1 illustrates that both the classical AP algorithm and its quantum counterpart successfully clustered the low-noise Gaussian datasets.

Number of clusters K	Efficiency classical AP (ε_t)	Efficiency quantum AP (ε_t)
4	1.00	0.99
5	1.00	1.00
6	0.99	0.98
7	1.00	0.98
8	0.98	0.94

Table 4.1: Efficiencies of AP algorithms for Gaussian datasets with different number of clusters.

Now, we apply the quantum AP algorithm to the physical dataset described in Section 4.5.1, which has been preprocessed as explained there. The results are presented in Fig. 4.9. In Fig. 4.9b, the same clustering is performed as in Fig. 4.9a (note that the efficiency of the quantum version is $\varepsilon_c = 1.00$). Although both algorithms show similar results, the AP algorithm identifies only 2 clusters, whereas the K-means algorithm finds 3 or 4 clusters (see Fig. 4.8). Nevertheless, both algorithms correctly identify the most energetic jets of the event (the blue and orange clusters), while the majority of the remaining particles are not classified in the same way, probably because they are soft particles.

⁵Since we consider unweighted events, our study is also applicable to e^+e^- colliders.



(a) Classical AP algorithm applied to LHC physical events.

(b) Quantum AP algorithm applied to LHC physical events, $\varepsilon_c = 1.00$.

Figure 4.9: A sample parton-level event, generated as described in the text, is clustered into $K = 2$ distinct clusters using both the classical and quantum versions of the AP algorithm.

4.5.3 Quantum k_T jet algorithm

The quantum maximum search algorithm discussed in Section 4.4.2 can be effectively applied to the k_T -jet algorithm. Even if the quantum algorithm does not select the absolute maximum during one of the multiple iterations, it is unlikely to impact the overall jet clustering process. This is because QCD predicts that secondary radiation is emitted mostly collinear to the parent parton. Therefore, a small deviation in finding the true maximum would alter the order in which two particles are merged, but in many cases, the final result remains unaffected by this permutation.

It is also worth noting that the quantum version of the k_T -jet algorithm would have a complexity of $\mathcal{O}(N^2 \log(N))$. This is because computing all the distances requires $\mathcal{O}(N^2)$, while finding the minimum takes $\mathcal{O}(\log(N))$, compared to the classical version, which has a complexity of $\mathcal{O}(N^3)$ [163]. Additionally, the quantum minimum search could be integrated into the **FastJet** [139] algorithm, which has a complexity of $\mathcal{O}(N^2)$. In this case, the resulting quantum algorithm would have a complexity of $\mathcal{O}(N \log(N))$, matching the efficiency of the **FastJet** algorithm with Voronoi diagrams, the most efficient clustering algorithm known to date.

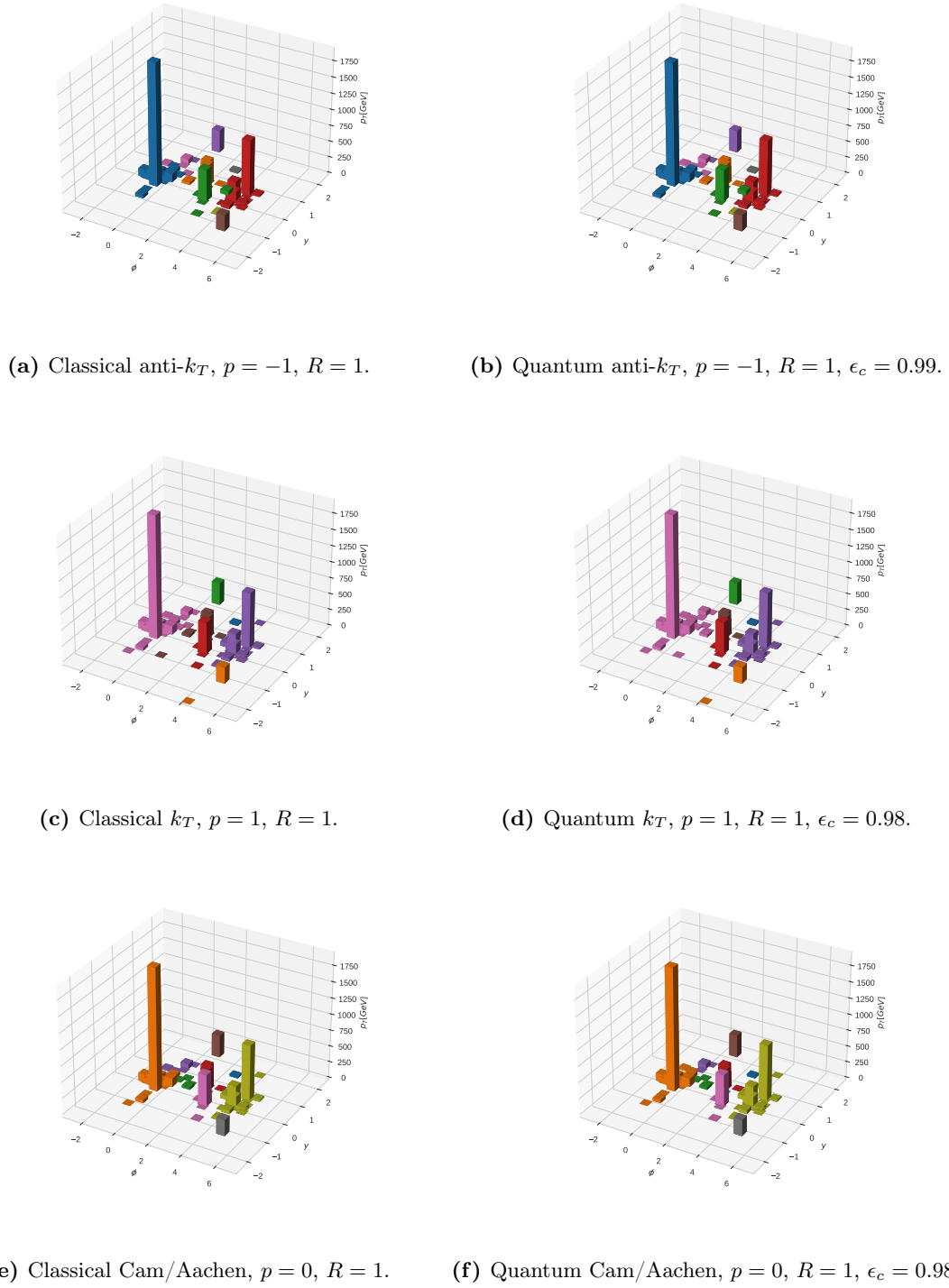


Figure 4.10: A sample parton-level event generated as described in the text and clustered with three different k_T -jets algorithms as well as its quantum versions.

We apply the quantum version of **FastJet** with the LHC physical datasets of the two previous sections. Figure 4.10 shows the performance of both classical and quantum k_T jet algorithms, illustrating the jet clustering process for each of the k_T algorithm variants: anti- k_T , k_T , and Cambridge/Aachen. Both the classical and quantum versions yield identical jet clusterings.

When comparing Figs. 4.8, 4.9, and 4.10, we observe that the k_T algorithm performs a cleaner clustering with a larger number of jets. This difference is primarily due to the fact that jet clustering is represented graphically in three dimensions, which aligns with the dimensionality of the k_T metric. In contrast, the K-means and AP algorithms use a four-dimensional Minkowski distance, which leads to a different clustering representation.

Furthermore, we analyze the efficiencies and the number of *shots* required for all quantum versions as a function of the a parameter. These results are presented in Table 4.2. Table 4.2 demonstrates that the efficiencies of the quantum algorithms are close to one, meaning they classify particles nearly identically to their classical counterparts. Additionally, it can be observed that as the parameter a increases, the number of *shots* required to achieve a successful efficiency decreases. Specifically, by setting a to 5, we attain the desired efficiencies with at most 10 *shots*. For other problems involving larger datasets, using a higher value of a can further separate the data points and maximize efficiency while minimizing the number of *shots* needed.

a	Efficiency anti- k_T	<i>Shots</i> anti- k_T	Efficiency k_T	<i>Shots</i> k_T	Efficiency Cam/Aachen	<i>Shots</i> Cam/Aachen
1	0.96	50	0.98	50	0.96	70
2	0.99	40	0.99	45	0.98	60
3	1.00	25	0.98	20	0.97	40
4	1.00	15	0.95	15	1.00	20
5	0.99	5	1.00	8	0.98	10

Table 4.2: Efficiencies and number of *shots* of the quantum k_T -jet algorithms as a function of the parameter a .

A summary of the time complexity of all the jet clusterings we consider alongside their quantum versions is presented in Table 4.3.

Jet clustering algorithm	Quantum subroutine	Classical version	Quantum version
K-means	Both	$\mathcal{O}(NKd)$	$\mathcal{O}(N \log K \log(d-1))$
AP	Distance	$\mathcal{O}(N^2Td)$	$\mathcal{O}(N^2T \log(d-1))$
anti- k_T Jet	Maximum	$\mathcal{O}(N^2)$	$\mathcal{O}(N \log N)$
anti- k_T FastJet	Maximum	$\mathcal{O}(N \log N)$	$\mathcal{O}(N \log N)$

Table 4.3: Time complexity of quantum and classical jet clustering algorithms when one or both subroutines are applied.

4.6 Conclusions

In this chapter, we have developed quantum versions of three well-known clustering algorithms [6, 8]: K-means, Affinity Propagation, and k_T -jet clustering. These quantum algorithms are built upon two novel quantum techniques. The first is a quantum subroutine designed to compute distances that satisfy the Minkowski metric, and the second is a quantum circuit used to track the maximum value from a list of unsorted data.

For the K-means algorithm, the quantum version builds upon the classical algorithm by incorporating a quantum procedure to compute Minkowski distances and a quantum circuit for assigning each particle to the closest centroid. We found that the quantum

K-means algorithm achieves clustering efficiency on par with its classical counterpart, while also offering an exponential speed-up in computational time, particularly as the vector dimensionality d and the number of clusters K increase, when executed on a quantum device with qRAM.

Next, we examined the quantum version of the Affinity Propagation algorithm, an unsupervised machine learning method. Similar to the K-means case, the quantum procedure computes the similarity using the same quantum technique for Minkowski distance. This results in an exponential speed-up over the classical version in terms of the vector dimensionality d , while preserving clustering efficiency.

Finally, we presented the quantum adaptations of the k_T -jet clustering algorithms. On a true universal quantum device, these algorithms would demonstrate an exponential speed-up in finding the minimum distance. Specifically, while the classical algorithm requires $\mathcal{O}(N^3)$ computational time, where N is the number of particles, the quantum version would only require $\mathcal{O}(N^2 \log(N))$. It is important to note that this comparison is made between the classical *non-optimal* and unoptimized version and its quantum counterpart. Further improvements can be made by applying a geometrical nearest-neighbor optimization procedure to the quantum algorithm, similar to that used in **FastJet**. This would result in a quantum version with a complexity of $\mathcal{O}(N \log(N))$, matching the fully optimized **FastJet** version.

In summary, the quantum simulations presented in this chapter demonstrate excellent performance and clustering efficiencies across all considered algorithms in comparison with their classical counterparts, while providing theoretical speedups when executed on quantum devices.

5 Quantum integration

5.1 Introduction

Quantum Field Theories (QFT) have emerged as highly successful for understanding the behavior of nature at sub-atomic scales. This achievement necessarily requires the development of robust computational methods to extract reliable predictions from QFTs. The perturbative approach has become a cornerstone technique in high-energy particle physics, since higher-order contributions within the perturbative expansion lead systematically to more precise and accurate theoretical predictions. However, the computation of higher-order contributions in perturbative QFT is not straightforward. The primary challenge lies in the virtual quantum fluctuations, which involve dealing with multiloop-multileg Feynman integrals. Loop Feynman integrals are an essential component in the calculation of quantum contributions to scattering and decay processes of elementary particles at high-energy colliders [186]. Their accurate evaluation is crucial for making precise theoretical predictions for experimental measurements. The difficulty of computing these integrals stems from their multi-dimensional nature, the dependence on multiple scales, and the emergence of ultraviolet (UV), infrared (IR) and threshold singularities, which often requires the use of regularization and renormalization techniques. Traditional methods can be computationally demanding, making it a challenging task.

The standard approach to computing multiloop scattering matrix elements starts with the corresponding Feynman diagrams and their associated scattering amplitudes. The corresponding Feynman integrals appearing in the expression are written in terms of bases of master integrals (MIs) using integration-by-parts identities (IBPs) relations [187, 188]. These MIs can be obtained analytically or numerically. One of the most widely used methods for calculating MIs is the so-called differential equation method [189, 190]. The success in completing the analytical formulae is based, roughly, on our knowledge of the *Multiple Polylogarithmic* functions (MPLs) [191]. More difficult cases involving elliptic functions [192] are challenging. Recently, much effort has been devoted to understanding the analytic structure of Feynman integrals that do not admit expressions in terms of MPLs [193, 194, 195, 196, 197, 198, 199, 200]. However, our understanding of the analytic structure of Feynman integrals beyond the MPLs class is not the same as for the MPLs case. Even when an analytic solution is available, the numerical evaluation of the functions associated with such a solution can be extremely challenging for phenomenological applications [201, 202, 203, 204, 205]. To overcome the difficulties described above, the generalized power series method [206, 207, 208] has recently been developed to solve the system of differential equations associated with the MIs. This technique has attracted much interest due to its wide range of applicability, and it has been successfully exploited in several phenomenological applications [209, 210, 211, 212].

Both approaches to multiloop scattering amplitudes (analytical or numerical/semi-analytical) produce results that are very slow to evaluate under phase-space integration. In the case of the two-loop scattering amplitude for three-photon production [201] the CPU cost is 29 years in a single core. Another example is the case of three-jet production [202], the time needed to complete the calculation at the second order in strong coupling is 115 CPU years¹. The preceding processes represent the current two-loop technical frontier, which refers to $2 \rightarrow 3$ particle collisions. It is important to note that all precedent CPU costs represent the time required for a single phenomenological study at the LHC. Each of them requires multi-cluster computers running continuously for several months (or even years, as in the case of three-jet production [202]). It is clear that our current state-of-the-art methodologies will not be able to cope with the huge demand for phenomenological studies that the LHC will require in the current Run 3 and future phases.

This evidences the need for computing Feynman integrals more efficiently, surpassing the current limitations imposed by hardware capabilities. In this regard, there is a growing interest in the development of innovative strategies based on the new paradigm brought by quantum computing with the potential to tackle traditionally challenging problems across various domains. The potential speedup offered by quantum computers has sparked numerous ideas, such as employing Grover’s algorithm for efficient database querying [168], Shor’s algorithm for factorization of large integers [213], or quantum annealing for Hamiltonian minimization [214]. In the realm of particle physics, quantum algorithms have found applications in different areas, such as solving problems associated with lattice gauge theories [215, 216]. Furthermore, quantum algorithms have been applied to various tasks in the context of high-energy colliders [7]. These include jet identification and clustering [217, 218, 6, 8], jet properties in a medium [219, 220, 108], determining parton densities (PDFs) [221], simulating parton showers [222], detecting anomalies [223, 224, 225], unveiling the causal structure of multiloop Feynman diagrams [126, 127, 128, 129], integrating elementary particle processes at tree level [5, 226], and optimizing existing particle accelerator beam lines [227]. The list of applications continues to expand rapidly as quantum algorithms demonstrate their suitability for a range of purposes. A very recent status of quantum computations in high-energy physics, split into theoretical and experimental target benchmark applications that can be addressed in the near future, can be found in [dimeglio2023quantum].

In view of these recent successes, the aim of the work presented in this Chapter is to investigate the potential of quantum algorithms for the efficient computation of loop Feynman integrals [2, 3]. In particular, we apply a novel quantum Monte Carlo integration algorithm called Quantum Fourier Iterative Amplitude Estimation (QFIAE) [5] to numerically evaluate several benchmark one-loop Feynman integrals and decay rates at second order in perturbation theory in the Loop-Tree Duality (LTD) [228, 229, 230, 231]. We use the LTD representation of the loop integrals because of its several advantages compared to other more conventional representations, as will be explained below.

The underlying basis of QFIAE are other well-known quantum algorithms, upon which QFIAE introduces a suitable solution to successfully solve some of the bottlenecks arising from these methods. Quantum Amplitude Estimation (QAE) [232] is a quantum algorithm that estimates the amplitude value of a quantum state. It uses amplitude amplification, a generalization of Grover’s searching algorithm [168], to enhance the likelihood of measuring the desired state over the non-desired state. The QAE method finds practical applications

¹Due to the high gluon multiplicity in the final state (up to 5 gluons), the most time-consuming contribution in this case is the double real emission.

in various domains, including our field of interest, i.e. numerical integration [233, 234]. However, the dependence of the original QAE algorithm on the resource-intensive Quantum Phase Estimation (QPE) subroutine [235], involving operations considered expensive for current Noisy Intermediate Scale Quantum (NISQ) devices, potentially undermines the speedup provided by Grover’s algorithm. To cope with this problem [236], several solutions have been suggested [237, 238, 239, 240, 241, 242, 243]. Among them, the Iterative Quantum Amplitude Estimation (IQAE) variant proposed in [243] stands as a successful candidate that maintains the speedup.

In the context of numerical integration, a Quantum Monte Carlo Integration (QMCI) method that utilizes QAE, or any of its variants, to achieve a quadratic speedup over the number of queries compared to its classical counterpart has been presented in [233]. However, this proposal does not address properly the challenge of preparing the quantum initial state, and in certain scenarios, it requires an extensive amount of quantum arithmetic operations, which may potentially diminish the quantum advantage. In view of this, the Fourier Quantum Monte Carlo Integration (FQMCI) method [244] introduces a unique approach to QMCI that harnesses the power of quantum computing without relying on arithmetic or phase estimation. This method stands out by accomplishing multiple goals simultaneously, which was not seen in previous proposals. The key idea is to use Fourier series decomposition to approximate the integrand and then estimate each component separately using QAE. Nevertheless, FQMCI relies on certain assumptions regarding the acquisition of Fourier coefficients, which may not hold in general. When these assumptions are not met, the quantum speedup might be wiped out.

The main novelty introduced by QFIAE [5] with respect to FQMCI [244] involves decomposing the target function into its Fourier series using a Quantum Neural Network (QNN) and subsequently integrating each trigonometric component using IQAE. As a result, this approach constitutes an end-to-end quantum algorithm that offers a viable strategy to maintain the quadratic speedup achieved by IQAE.

This chapter is organized as follows. First, in Section 5.2 the loop Feynman integrals in LTD are introduced. Then in Section 5.3, the QFIAE method is introduced and described in detail. In Section 5.4, we motivate the application of quantum algorithms to specific loop Feynman diagrams, and present numerical results in simulators and hardware of the quantum integration of different loop topologies: tadpole, bubble, triangle and pentagon. Then, in Section 5.5, we present the decay rates of Higgs bosons, photons, and scalars computed using QFIAE on both simulators and real quantum devices. Finally, Section 5.6 provides a brief summary and discussion of the results.

5.2 Loop Feynman integrals in LTD

QFT stands as one of the most accurate theories for producing theoretical predictions at high-energy colliders. However, its inherent complexity makes analytical calculations of physical processes exceptionally challenging. The perturbative approach in QFT expresses physical observables (such as cross sections or decay rates) as power series in the interaction couplings, with each term in the expansion corresponding to tree-level and loop Feynman integrals of increasing complexity. These mathematical expressions involve sophisticated multi-dimensional integrations over internal momenta, creating significant computational challenges due to UV and IR divergences that require careful regularization and renormalization. Regarding their physical interpretation, terms at high perturbative orders are elegantly represented through Feynman diagrams, where loop

diagrams specifically capture quantum fluctuations arising from virtual particles circulating in closed paths [245].

Despite the technical difficulties in estimating their value, loop Feynman integrals are essential to deliver precision calculations in particle physics. They enable theoretical predictions for scattering amplitudes, decay rates, and cross-sections that can be directly compared with experimental measurements.

5.2.1 Loop-Tree Duality

In this context, the Loop-Tree Duality (LTD) [228, 229, 230, 231] is an innovative methodology to deal with multiloop Feynman integrals and scattering amplitudes. Specifically, LTD transforms loops defined in the Minkowski space of the loop four-momenta into trees defined in the Euclidean space of their spatial components, and reinterprets virtual states as configurations that resemble real-radiation processes. Among other advantages, this transformation provides a more intuitive understanding of the singular structure of loop integrals [246, 247]. In particular, the most remarkable property of LTD is the existence of a manifestly causal representation [228, 248, 249, 250, 251, 252], i.e., an integrand representation where certain nonphysical singularities are absent and therefore yields integrands that are numerically more stable.

The fact that the integration domain in LTD is Euclidean and not Minkowski also brings additional advantages, both for analytic applications such as asymptotic expansions [253, 254], where the hierarchy of scales is well defined, and numerical applications [255, 256] because the number of loop integration variables is independent of the number of external particles. For example, at one loop the number of independent integration variables is always three, the number of spatial components of the loop momentum, although for certain kinematic configurations this number can be reduced when the dependence on any of these variables is trivial. This is the case for tadpole and bubble diagrams at one loop, as we will see in more detail in Section 5.4.

Moreover, LTD offers a unified framework for cross-section calculations, since the dual representation of loop integrals in Euclidean domains allows a direct combination of virtual and real contributions at the integrand level, resulting in a fully local cancellation of IR singularities, the so-called Four-Dimensional Unsubtraction (FDU) [257, 258, 259, 260, 261] and UV singularities by appropriate UV local counterterms [253, 262], without the need for additional regulators, such as Dimensional Regularization (DREG) [263, 264]. Also, it is worth to remark that quantum algorithm have already been used to bootstrap the causal structure of multiloop Feynman diagrams in LTD [126, 127, 128, 129].

To illustrate mathematically how the LTD formalism work let us consider a generic L -loop scattering amplitude with N external legs $\{p_j\}^N$ and n sets of internal lines, each defined by specific dependencies on loop momenta, can be expressed as [261]

$$\mathcal{A}_N^{(L)}(1, \dots, n) = \int_{\ell_1, \dots, \ell_L} \mathcal{A}_F^{(L)}(1, \dots, n), \quad (5.1)$$

where

$$\mathcal{A}_F^{(L)}(1, \dots, n) = \mathcal{N}(\{\ell_s\}^L, \{p_j\}^N) \times G_F(1, \dots, n). \quad (5.2)$$

This integral in the Minkowski space involves the L -loop momenta $\{\ell_s\}^L$, the product of Feynman propagators $G_F(q_i) = (q_i^2 - m_i^2 + i0)^{-1}$, and numerators $\mathcal{N}(\{\ell_s\}^L, \{p_j\}^N)$ derived from the Feynman rules of the theory. In Dimensional Regularization (DREG),

the d -dimensional integration measure is defined as

$$\int_{\ell_s} = -i\mu^{4-d} \int \frac{d^d \ell_s}{(2\pi)^d}. \quad (5.3)$$

The standard Feynman propagator for a single particle reads

$$G_F(q_{i_s}) = \frac{1}{q_{i_s,0}^2 - \left(q_{i_s,0}^{(+)}\right)^2}, \quad (5.4)$$

where

$$q_{i_s,0}^{(+)} = \sqrt{\vec{q}_{i_s}^2 + m_{i_s}^2 - i0} \quad (5.5)$$

represents the positive on-shell energy of the loop momentum q_{i_s} in terms of its spatial components $\vec{q}_{i_s}$, mass m_{i_s} , and the infinitesimal Feynman complex prescription $i0$. Also, s denotes the set of internal propagators with momenta $q_{i_s} = \ell_s + k_{i_s}$, depending on the loop momentum ℓ_s or specific linear combinations of loop momenta, along with external momenta k_{i_s} , where $i_s \in s$. The Feynman propagators can be raised to arbitrary powers.

The LTD representation is derived through iterative application of Cauchy's residue theorem. This process integrates out one degree of freedom for each loop momentum, with the Cauchy contour closed below the real axis to select poles with negative imaginary components in the complex plane. This procedure modifies the infinitesimal complex prescription of the Feynman propagators, requiring careful treatment to preserve causality. Starting from Eq. 5.1, we set on-shell the propagators dependent on the first loop momentum q_{i_1} and define

$$\mathcal{A}_D^{(L)}(1; 2, \dots, n) \equiv \sum_{i_1 \in 1} \text{Res} \left(\mathcal{A}_F^{(L)}(1, \dots, n), \text{Im}(q_{i_1,0} < 0) \right), \quad (5.6)$$

where taking the residue effectively integrates out the energy component of the loop momenta. We then construct the nested residue, iterating until the r -th set as

$$\mathcal{A}_D^{(L)}(1, \dots, r; r+1, \dots, n) = \sum_{i_s \in s} \text{Res} \left(\mathcal{A}_D^{(L)}(1, \dots, r-1; r, \dots, n), \text{Im}(q_{i_s,0} < 0) \right). \quad (5.7)$$

In this representation, all sets preceding the semicolon contain one propagator set on-shell and are linearly independent, while remaining propagators stay off-shell. This transforms the loop amplitude into non-disjoint trees. Then, after integration of the energy component, the integration measure becomes

$$\int_{\vec{\ell}_s} \equiv -\mu^{d-4} \int \frac{d^{d-1} \ell_s}{(2\pi)^{d-1}}, \quad (5.8)$$

converting the d -dimensional Minkowski space into a $(d-1)$ -dimensional Euclidean space.

5.3 Quantum Fourier Iterative Amplitude Estimation

This section is based on [5] where we present a Quantum Monte Carlo Integrator dubbed Quantum Fourier Iterative Amplitude Estimation (QFIAE). The QFIAE algorithm is build upon Quantum Amplitude Estimation and uses a Quantum Neural Network to decompose the target function to be integrated into a Fourier series to ensure efficient encoding into a quantum circuit. A tutorial on using this quantum integrator, and the source code used to produce the results in this section, are available in [265] and [266], respectively.

5.3.1 Iterative Quantum Amplitude Estimation

Quantum Amplitude Estimation (QAE) is a quantum algorithm that relies on Grover's amplitude amplification subroutine to efficiently estimate the amplitude of a quantum state. One of their applications include integration, as long as one can encode the desired integral as the amplitude of what we will call the "good state". However, the general implementation [232] relies on Quantum Phase Estimation (QPE) which involves many quantum gates and makes it difficult to be implemented in current NISQ devices. In this context the Iterative Quantum Amplitude Estimation (IQAE) algorithm proposed in [243] is an alternative implementation of QAE that relies exclusively on Grover's algorithm. This variant replaces the QPE subroutine with an efficient classical post-processing methodology, thereby reducing the necessary quantum resources in terms of qubits and gate operations. The IQAE algorithm incorporates two essential components: quantum amplitude amplification and a methodical iterative search to identify the optimal number k of amplifications required for accurate estimation. The amplitude amplification [267, 268, 232] technique, which extends the Grover's quantum search algorithm [269], achieves a quadratic enhancement in performance relative to classical approaches. Let us consider a unitary operator $\mathcal{A}$ operating on $n + 1$ qubits such that:

$$|\psi\rangle = \mathcal{A}|0\rangle_{n+1} = \sqrt{a}|\tilde{\psi}_1\rangle|1\rangle + \sqrt{1-a}|\tilde{\psi}_0\rangle|0\rangle, \quad (5.9)$$

where $a \in [0, 1]$ represents the parameter we aim to estimate, while $|\tilde{\psi}_1\rangle$ and $|\tilde{\psi}_0\rangle$ denote the n -qubit good and bad states, respectively. Through Eq. 5.9, parameter a could be determined from the ratio of good to bad states, but this approach provides no computational advantage as its query complexity would match that of classical methods. The quantum advantage materializes through amplitude amplification, enabling a quadratic reduction in query complexity through the application of the amplification operator:

$$\mathcal{Q} = -\mathcal{A}S_0\mathcal{A}^{-1}S_\chi, \quad (5.10)$$

where S_0 marks the $|0\rangle_{n+1}$ state with a negative sign while preserving other states, and S_χ inverts the sign of the good state, specifically $S_\chi|\tilde{\psi}_1\rangle|1\rangle = -|\tilde{\psi}_1\rangle|1\rangle$. By defining a parameter $\theta_a \in [0, \pi/2]$ where $\sin^2 \theta_a = a$, we can express:

$$|\psi\rangle = \mathcal{A}|0\rangle_{n+1} = \sin \theta_a |\tilde{\psi}_1\rangle|1\rangle + \cos \theta_a |\tilde{\psi}_0\rangle|0\rangle. \quad (5.11)$$

Brassard et al. demonstrated in [232] that applying $\mathcal{Q}$ repeatedly k times to ψ yields:

$$\mathcal{Q}^k |\psi\rangle = \sin((2k+1)\theta_a) |\tilde{\psi}_1\rangle|1\rangle + \cos((2k+1)\theta_a) |\tilde{\psi}_0\rangle|0\rangle. \quad (5.12)$$

Eq. 5.12 indicates that after k applications of operator $\mathcal{Q}$, the probability of obtaining the good state increases by approximately $4k^2$ times compared to the probability from Eq. 5.11 for sufficiently small values of a . This represents the quadratic acceleration achieved through amplitude amplification, as $2k$ measurements from $\mathcal{A}|0\rangle_{n+1}$ would only enhance the probability of obtaining the good state by a factor of $2k$. Thus, if we can determine the proportion of good states following amplitude amplification, we can estimate a based on the query count required to achieve this proportion. The procedure for identifying the optimal value of k constitutes the innovative contribution of the IQAE algorithm. This process involves a confidence interval $[\theta_l, \theta_u] \subseteq [0, \pi/2]$ with $\theta_l < \theta_a < \theta_u$, a power k of $\mathcal{Q}$, and an estimate for $\sin^2((2k+1)\theta_a)$. Applying the trigonometric identity $\sin^2(x) = (1 - \cos(2x))/2$, our estimates for $\sin^2((2k+1)\theta_a)$ are converted to estimates for $\cos((4k+2)\theta_a)$, considering whether the argument falls within $[0, \pi]$ or $[\pi, 2\pi]$. The objective is to identify the maximum k for which the interval $[(4k+2)\theta_l, (4k+2)\theta_u] \bmod 2\pi$ remains entirely within either the

upper or lower half-plane. This establishes an upper bound for k , and the core innovation of this algorithm lies in the methodology used to determine the next value of k given the interval $[\theta_l, \theta_u]$.

5.3.2 Quantum Monte Carlo Integration

Monte Carlo integration is a powerful technique widely used in fields such as particle physics, finance, and cosmology. It estimates definite integrals using random sampling:

$$I = \int_{x_{min}}^{x_{max}} p(x)f(x)dx. \quad (5.13)$$

To compute this integral using the Monte Carlo method, we draw $2n$ independent and identically distributed (i.i.d.) samples $x_i \in \{0,1\}^n$, with $i = 0, \dots, 2n-1$, from the probability distribution $p(x)$ over the interval $[x_{min}, x_{max}]$. The integral is then approximated by the mean value of $f(x)$ over these samples.

The main objective of discrete Monte Carlo integration is to compute the expected value of a real function $0 \leq f(x) \leq 1$ defined for n -bit inputs $x \in \{0,1\}^n$ with probability $p(x)$:

$$\mathbb{E}[f(x)] = \sum_{x=0}^{2^n-1} p(x)f(x). \quad (5.14)$$

Although Monte Carlo integration has been successfully applied across various fields, certain complex integrals remain highly challenging. This is particularly true for multivariate functions with sparse probability distributions, which are common in HEP. These challenges in Monte Carlo integration methods have motivated the development of a quantum counterpart [233]. The key idea behind this quantum approach is to encode the integral as the amplitude of a quantum state and then apply QAE to estimate this amplitude, and thus the integral.

The convergence of Quantum Monte Carlo Integration (QMCI), measured by the Mean Square Error (MSE), offers a quadratic advantage in the number of samples drawn from the probability distribution. This speedup comes from the use of the Grover's quantum amplitude amplification subroutine, which underlies QAE.

As discussed in [237], the QMCI algorithm prepares the operator $\mathcal{A}$ from two components:

$$\mathcal{P}|0\rangle_n = \sum_{x=0}^{2^n-1} \sqrt{p(x)}|x\rangle_n, \quad (5.15)$$

which encodes the probability distribution $p(x)$ into the state $|0\rangle_n$, and

$$\mathcal{R}|x\rangle_n|0\rangle = |x\rangle_n \left(\sqrt{f(x)}|1\rangle + \sqrt{1-f(x)}|0\rangle \right), \quad (5.16)$$

which encodes the function $f(x)$ into an ancillary qubit. Applying $\mathcal{A}$ to the $(n+1)$ -qubit initial state results in:

$$\begin{aligned} |\psi\rangle &= \mathcal{A}|0\rangle_{n+1} = \mathcal{R}(\mathcal{P} \otimes \mathbb{I}^1)|0\rangle_{n+1} \\ &= \sum_{x=0}^{2^n-1} \sqrt{p(x)}|x\rangle_n \left(\sqrt{f(x)}|1\rangle + \sqrt{1-f(x)}|0\rangle \right), \end{aligned} \quad (5.17)$$

where $\mathbb{I}^1$ is the identity operator acting on the ancillary qubit. We now define:

$$a = \sum_{x=0}^{2^n-1} p(x)f(x) = \mathbb{E}[f(x)], \quad (5.18)$$

$$|\tilde{\psi}_1\rangle = \frac{1}{\sqrt{a}} \sum_{x=0}^{2^n-1} \sqrt{p(x)}\sqrt{f(x)}|x\rangle_n, \quad (5.19)$$

$$|\tilde{\psi}_0\rangle = \frac{1}{\sqrt{1-a}} \sum_{x=0}^{2^n-1} \sqrt{p(x)}\sqrt{1-f(x)}|x\rangle_n, \quad (5.20)$$

and express $|\psi\rangle$ as:

$$|\psi\rangle = \sqrt{a}|\tilde{\psi}_1\rangle|1\rangle + \sqrt{1-a}|\tilde{\psi}_0\rangle|0\rangle. \quad (5.21)$$

which looks identical to Eq. 5.9.

As motivated before, we can now interpret Monte Carlo integration as an amplitude estimation problem, where the probability of the target state, a , corresponds to the integral to be computed, $\mathbb{E}[f(x)]$. What is left now is the operator $\mathcal{Q}$, which is constructed using U_ψ and U_{ψ_0} :

$$\mathcal{Q} = U_\psi U_{\psi_0}, \quad (5.22)$$

where

$$U_{\psi_0} = \mathbb{I}_{n+1} - 2\mathbb{I}_n|0\rangle\langle 0|, \quad U_\psi = \mathbb{I}_{n+1} - 2|\psi\rangle\langle\psi|, \quad \mathbb{I}_n \equiv \mathbb{I}^{\otimes n}. \quad (5.23)$$

Defining $a = \sin^2 \theta_a$ and applying Eq. 5.21, the IQAE algorithm described earlier can be used for Monte Carlo integration. The quantum circuit implementing this procedure is shown in Fig. 5.1.

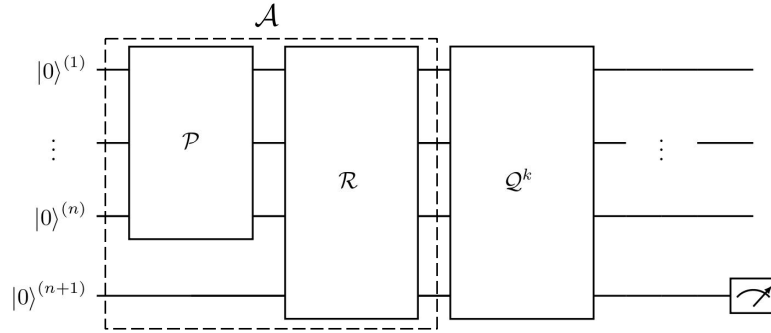


Figure 5.1: Quantum circuit for amplitude amplification and estimation in Monte Carlo integration.

5.3.3 Fourier Quantum Monte Carlo Integration

Although it is possible to integrate any function $f(x)$ that follows a probability distribution $p(x)$ using the QMCI described in Sec. 5.3.2, this comes at the cost of creating the quantum circuit $\mathcal{R}$, which encodes the function $f(x)$. If one wants to consider the most general scenario, where $f(x)$ has a complicated behavior, the computational cost of building such

circuit $\mathcal{R}$ could be prohibitively complex, potentially killing the quantum speedup provided by Grover's amplitude amplification subroutine.

Even the trivial case where $f(x) = x$ (i.e., just finding the mean of $p(x)$) would imply significant amounts of arithmetic to be performed quantumly. However, an important exception is the specific case where $f(x) = \sin^2(mx + c)$ for some constants m and c , which can be encoded by a bank of R_y rotation gates.

With this in mind, the author in [244] introduced the Fourier Quantum Monte Carlo Integration (FQMCI) method that is based on the idea of decomposing the desired function $f(x)$ into sines and cosines. The procedure starts by extending $f(x)$ as a periodic piecewise function. Let us consider $\mathbf{f}(x)$, which repeats with period $x_{\bar{u}} - x_l$:

$$\mathbf{f}(x) = \begin{cases} f(x) & \text{if } x_l \leq x \leq x_u \\ \tilde{f}(x) & \text{if } x_u \leq x \leq x_{\bar{u}}, \end{cases} \quad (5.24)$$

where $\tilde{f}(x)$ is itself sufficiently smooth and chosen such that $f(x_l) = \tilde{f}(x_{\bar{u}})$ and $f(x_u) = \tilde{f}(x_u)$, which holds true for their derivatives w.r.t to x . Since $\mathbf{f}(x)$ is periodic, it has a Fourier series:

$$\mathbf{f}(x) = c + \sum_{n=1}^{\infty} (a_n \cos(n\omega x) + b_n \sin(n\omega x)), \quad (5.25)$$

where $\omega = 2\pi/T$ and T is the period of the periodic piecewise function. Those various trigonometric components of $\mathbf{f}(x)$ can be estimated individually, using IQAE, and then all the individual integrals are collected to obtain the final result.

Furthermore, it is shown that the quadratic quantum advantage is indeed retained with this method under certain conditions. Those include the ability to prepare a specific probability distribution as a shallow-depth quantum state and apply a function to random samples such that the mean cannot be calculated analytically. Additionally, it is important to be cautious when evaluating Fourier coefficients as numerical integration may shift the computational load rather than reduce the complexity. In [244], the author claims that for commonly used functions such as the mean ($f(x) = x$), Fourier coefficients can be calculated symbolically. Moreover, he adds that if the coefficients cannot be found symbolically, for commonly used functions it may be reasonable to assume that they have been pre-computed and stored in advance. Therefore they would not add to the complexity of the overall cost.

Nevertheless, in the most general possible scenario, the Fourier coefficients need to be calculated and we cannot rely on them being symbolically computable or stored in advance. Hence, to attain the desired speedup for the whole computation, a new procedure to estimate the Fourier coefficients without relying on numerical integration must be introduced. This leads us to propose in Sec. 5.3.5 a novel method for obtaining the Fourier series of a function using a QNN.

5.3.4 Quantum Fourier Series

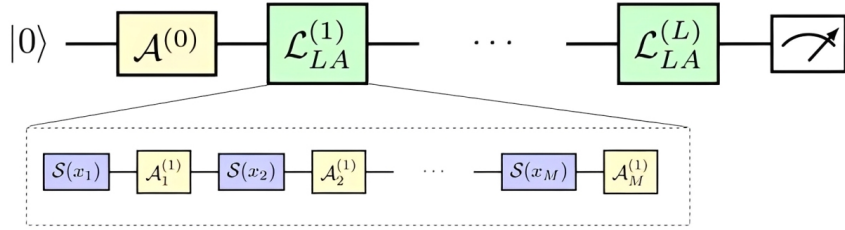
As explained in Sec. 3.4.3, a QNN typically consists of a layered architecture that encodes input data into quantum states and processes them in a high-dimensional feature space. The encoding strategy and circuit Ansatz play a key role in outperforming classical NNs. While complex data encoding with feature maps that are hard to simulate may enable a quantum advantage, too expressive circuits can lead to flat cost landscapes, making the model difficult to train [270].

With this in mind, it is possible to implement a QNN to fit a multivariate function $f(\vec{x})$ and extract its Fourier series from the trained quantum circuit. The architecture considered as introduced in [85], consists of interleaved encoding and trainable circuit blocks. The general form of the Ansatz is:

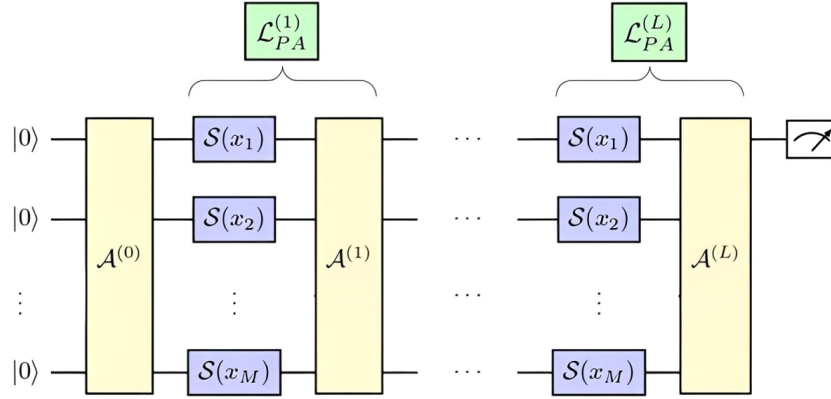
$$U_0 \equiv \mathcal{A}(\vec{\theta}_0) , \quad U_l \equiv \mathcal{A}(\vec{\theta}_l) \mathcal{S}(\vec{x}) . \quad (5.26)$$

More specifically, the circuit consists of three main components: an initial operator, $\mathcal{A}(\vec{\theta}_0)$, that prepares a superposition state; an encoding layer, $\mathcal{S}(\vec{x})$, that maps the input data; and a trainable layer, $\mathcal{A}(\vec{\theta}_l)$, optimized according to a chosen metric. The cost function is defined as the squared difference between the circuit's output (expected value in the computational basis) and the target function.

We consider two strategies to encode the data sampled from the target function. In the first approach, the circuit follows a linear Ansatz, where each feature, i.e., each x value, is encoded in a single qubit, as shown in Fig. 5.2(a). The second approach, the circuit follows a parallel Ansatz, where each feature, i.e. each x value, is encoded in a different qubit as shown in Fig. 5.2(b).



(a) Linear Ansatz



(b) Parallel Ansatz

Figure 5.2: Quantum circuit Ansätze of the QNN. The Linear Ansatz (a) encodes each data feature in a single qubit. Thus the circuit depth grows linearly with the total number of variables M . The Parallel Ansatz (b) encodes the M variables in M qubits. In both Ansätze the circuit depth grows linearly with the number of layers L . Note that for $M = 1$, both the Linear and Parallel Ansatz converge to the same linear circuit.

Following the explanation in Sec. 3.4.3, the expectation value of this quantum model corresponds to a universal multidimensional Fourier series representation [271, 272, 273]:

$$\langle M \rangle(\vec{x}, \vec{\theta}) = \sum_{\vec{w} \in \Omega} c_w e^{i\vec{x} \cdot \vec{w}} , \quad (5.27)$$

where the Fourier components arise from the circuit elements. The encoding gate for each variable $\mathcal{S}(x) = e^{xH}$ determines the frequency w , given a Hamiltonian H . For a single qubit with $H = \frac{1}{2}\sigma_z$, the frequency spectrum corresponds to the eigenvalues of H . The Fourier coefficients c_w depend on the trainable parameters $\vec{\theta}_l$ of the unitary gates $\mathcal{A}(\vec{\theta}_l)$.

The choice of data encoding influences the maximum frequency in the output Fourier series, which scales with the number of layers L and the computational space dimension $D = \max(\Omega) = (d - 1)L$. In the qubit setup considered in this Section, where $d = 2$, increasing L enables the circuit to represent more complex functions.

5.3.5 Quantum Fourier Iterative Amplitude Estimation

The quantum algorithm we are introducing in this section presents an elegant solution to the problem presented in the FQCMi method. The core idea of the Quantum Fourier Iterative Amplitude Estimation (QFIAE) algorithm is to exploit an appropriate choice of the encoding to design a QNN that represent a truncated Fourier series, as demonstrated in [87, 88] and explained in Sections 3.4.3 and 5.3.4. Then, if a quantum subroutine is used to obtain the Fourier series, this QNN can be integrated with the FQCMi algorithm to establish a fully quantum pipeline for Monte Carlo integration. The workflow of the QFIAE algorithm is presented in Fig. 5.3, and a tutorial using the open-source software for quantum computation Qibo [274] is available in [265].

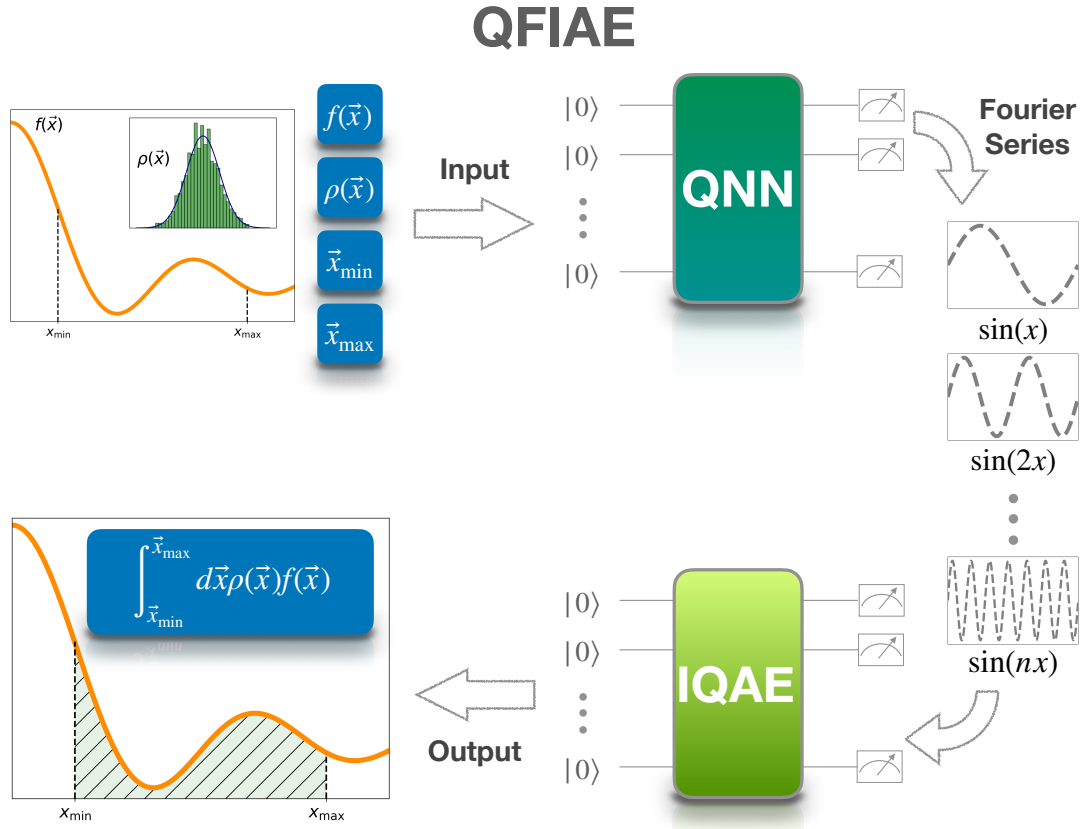


Figure 5.3: Workflow of QFIAE. The input consists of the target function $f(\vec{x})$, the probability distribution $\rho(\vec{x})$, and the integration domain $\{\vec{x}_{\min}, \vec{x}_{\max}\}$. The QNN fits $f(\vec{x})$ and extracts its Fourier series from the quantum circuit. Next, IQAE estimates the integral for each trigonometric term in the Fourier series. Finally, these integrals are added with their corresponding coefficients to obtain the final integral result.

Let us see how this algorithm performs in practice with a simple example.

5.3.6 Benchmark example: integral of $1 + x^2$

To benchmark the QFIAE method for integration, we consider the following integral:

$$I = \int_0^1 (1 + x^2) dx. \quad (5.28)$$

This choice is motivated by its simplicity, which provides a clear pedagogical demonstration of the method while being relevant to a well-known particle physics case. The integrand, $f(x) = 1 + x^2$, represents a simple yet non-trivial form of the differential cross-section for the scattering process $e^+e^- \rightarrow q\bar{q}$ at the Leading Order (LO). In quantum electrodynamics (QED), this process is well described by parametrizing the phase space with two angles, leading to the total cross-section expression²:

$$\sigma \sim \int_{-1}^1 \int_0^{2\pi} d\cos\theta d\phi (1 + \cos^2\theta) \quad (5.29)$$

This integral was previously computed using the IQAE algorithm with a different approach in [226]. There, a quantum Generative Adversarial Network (QGAN) was employed to load the normalized distribution $1 + x^2$. The QGAN was trained using classically generated samples to encode the function $f(x)$ into a quantum state, after which the integral was estimated using IQAE. While [226] demonstrated a quadratic speedup over classical Monte Carlo integration in terms of query complexity, the total circuit depth, given by the complexity of the QGAN architecture and the IQAE part makes this approach unsuitable for NISQ devices. In contrast, the QFIAE method we propose maintains a feasible circuit depth for NISQ hardware, making it more practical for real-world applications in near-term devices.

Starting from the mentioned example, we evaluate the performance of our quantum algorithm in computing the same integral. Since the QNN can only fit functions with values in the range $[-1, 1]$, we first normalize the function $f(x) = 1 + x^2$ before applying the QFIAE method.

The first step of the quantum integration algorithm involves fitting the function using a QNN, as shown in Fig. 5.4. The QML model achieves remarkable accuracy, exceeding 99%, demonstrating the reliability of our method for function fitting. The model's performance can be optimized by tuning hyperparameters such as the number of layers, which is related to the number of Fourier terms n_{Fourier} , the size of training and test datasets, and the number of optimizer iterations, *nepochs*. Adjusting these parameters allows balancing between learning time and accuracy.

The next step extracts the Fourier coefficients from the quantum circuit and reconstructs the Fourier series:

$$\begin{aligned} f(x) \approx & 0.476 + 1.169 \cos(x) - 0.263 \cos(2x) + \dots \\ & - 0.017 \cos(9x) + 0.004 \cos(10x) \\ & - 0.125 \sin(x) - 0.278 \sin(2x) + \dots \\ & - 0.029 \sin(9x) - 0.004 \sin(10x). \end{aligned} \quad (5.30)$$

²Note that Eq. (5.28) and Eq. (5.29) differ in their integration intervals. However, since the function is even, they are equivalent up to a factor of 2.

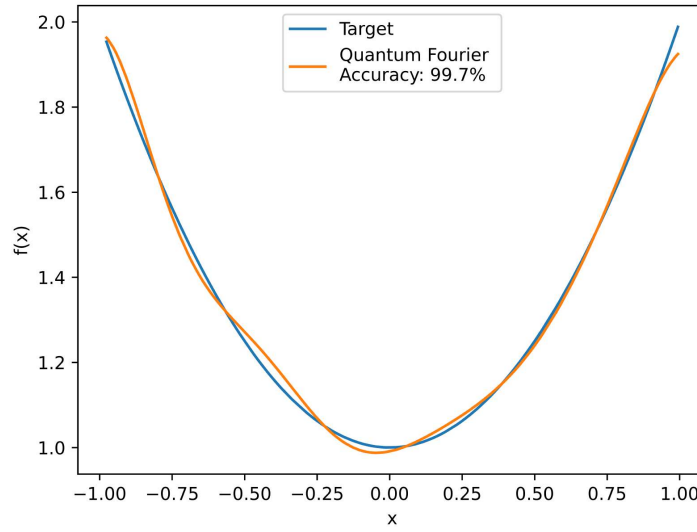


Figure 5.4: Quantum simulation using a Linear Ansatz in the QNN to fit the function $1 + x^2$. The setup uses $n_{\text{qubits,QF}} = 1$ and $L = 10$, resulting in a Fourier series with $n_{\text{Fourier}} = 10$ coefficients. A gradient descent method, the Adam optimizer, is employed with 200 data points in the range $[-1, 1]$. The optimizer hyperparameters are $\text{learning_rate} = 0.05$ and $\text{epochs} = 100$.

Once the function is expressed as a Fourier series, as shown in Eq. (5.30), the trigonometric components are transformed into the $\sin^2(ax + b)$ form and integrated using IQAE. The probability distribution considered in this example is $p(x) = 1/2^n$, which is prepared as a shallow-depth quantum state to preserve the quadratic quantum advantage. Specifically, it is generated by applying an n -dimensional Hadamard gate, $\mathcal{H}^{\otimes n}$.

At this point, one can compute the integrals, which are weighted by their corresponding coefficients, and summed them to estimate the integral in Eq. (5.28). The results, along with a comparison to the analytical value, are presented in Table 5.1.

In Table 5.1, we present an estimate of the integral of the function $1 + x^2$ over the interval $[0, 1]$. The integral is calculated using two methods: the QFIAE method and the FQMCI method introduced in [244]. The key distinction between these two algorithms lies in the calculation of the Fourier series. In QFIAE, the Fourier series is estimated using a QNN, while in FQMCI, it is computed classically through numerical integration. The integrals are evaluated for different values of the number of Fourier coefficients, n_{Fourier} , and the number of shots used in the IQAE algorithm, shots . In all cases, the IQAE algorithm employs $n_{\text{qubits,IQAE}} = 4$ qubits, with a confidence interval of $\alpha = 0.05$ and an estimated error of $\epsilon = 0.01$. The third column of Table 5.1 compares the obtained values with the exact result ($I_{\text{exact}} = 4/3$). Each integral is computed 50 times to minimize statistical fluctuations, and the results are averaged to compare the method's performance and identify the optimal set of parameters. The errors reported by the IQAE method in **Qibo** could be reduced by adjusting the ϵ parameter. From the analysis in the third column, we observe that QFIAE performs at least as well as FQMCI, demonstrating that estimating the Fourier series with a QNN does not produce any loss in accuracy. Additionally, the best performance is achieved with $n_{\text{Fourier}} = 10$ and $\text{shots} = 1000$, confirming the intuition that a higher number of Fourier coefficients increases the expressibility of the Fourier series, and also a larger number of shots improves precision.

method = QFIAE	I_{est}	ϵ
$n_{Fourier} = 5,$ $shots = 100$	1.32 ± 0.05	0.991
$n_{Fourier} = 10,$ $shots = 100$	1.34 ± 0.06	1.006
$n_{Fourier} = 5,$ $shots = 1000$	1.33 ± 0.05	0.998
$n_{Fourier} = 10,$ $shots = 1000$	1.33 ± 0.04	0.999
method = FQMCI	I_{est}	ϵ
$n_{Fourier} = 5,$ $shots = 100$	1.35 ± 0.07	1.010
$n_{Fourier} = 10,$ $shots = 100$	1.34 ± 0.07	1.003
$n_{Fourier} = 5,$ $shots = 1000$	1.34 ± 0.07	1.007
$n_{Fourier} = 10,$ $shots = 1000$	1.34 ± 0.06	1.002

Table 5.1: Integration of $1 + x^2$ from $[0, 1]$ using QFIAE and FQMCI. With $\epsilon = I_{est}/I_{exact}$.

$QF_{depth} = \mathcal{A}_{depth} + layers(\mathcal{A}_{depth} + \mathcal{S}_{depth})$			
$\mathcal{A}_{depth}$	$\mathcal{S}_{depth}$	$layers$	QF_{depth}
3	1	10	43
$IQAE_{depth} = \mathcal{A}_{depth} + k\mathcal{Q}_{depth}$			
$\mathcal{A}_{depth}$	$\mathcal{Q}_{depth}$	k	$IQAE_{depth}$
4	12	9^3	112

Table 5.2: Depths of the Quantum Fourier (QF) and Iterative Quantum Amplitude Estimation (IQAE) parts of the QFIAE method.

To evaluate the feasibility of our algorithm on current NISQ devices, we consider the Quantum Volume (QV) metric introduced by IBM in [275]. This metric quantifies the maximum size of square quantum circuits that can be successfully executed by a quantum computer. However, it is important to note that the QV metric may not accurately predict the performance of circuits that are deep and narrow or wide and shallow [276]. This limitation is particularly relevant to our analysis, as shown in Table 5.2, where two circuits are much deeper than they are wide, with $n_{qubits,QF} = 1$ and $n_{qubits,IQAE} = 4$. Consequently, the suitability of our circuits is further assessed by referencing a recent study conducted by IonQ [276], which is detailed in [277].

In their study, a set of quantum algorithms was tested and compared using QED-C benchmarks. The results indicate that our algorithm, with a quantum depth of 112 and a low qubit count (≤ 4), can be executed with a high probability of success on IonQ and Quantinuum devices. This suggests that running our proposed algorithm on current NISQ processors, is not only feasible but also a promising endeavor.

In the following sections, Sec. 5.4 and Sec. 5.5, we demonstrate how the QFIAE algorithm is fully or partially implemented on quantum computers and successfully computes integrals

³The value of k is calculated by the algorithm and it changes in every iteration and integral. Hence the different values have been averaged, and the mean value is presented in the table.

that are highly relevant to HEP processes.

5.4 Loop Feynman integration on a quantum computer

In this section, we apply the QFIAE method to estimate a selection of loop Feynman integrals. While quantum algorithms have been previously applied to loop integrals [126, 127, 128, 129] to unveil their causal structure, and quantum integration methods have been employed in elementary particle physics processes at tree level [226], this represents the first application of a quantum algorithm that computes Feynman loop integrals. This achievement holds great significance as these integrals present both technical and conceptual challenges. From a technical standpoint, they involve cumbersome multi-dimensional functions with UV, IR and threshold singularities. In terms of their physical importance, loop Feynman integrals serve as fundamental components in precision physics. Therefore, developing an efficient method to calculate them is crucial for facilitating rapid computations that have the potential to drive new discoveries and enhance our understanding of the fundamental principles governing the universe.

5.4.1 QFIAE implementation

In this subsection we will explain the details of how we will implement the QFIAE method introduced in Sec.5.3 in hardware and simulators.

5.4.1.1 Simulators implementation

At this point, we can implement the QFIAE algorithm using two different simulation frameworks. PennyLane [61] will be used for QNN implementation, whereas Qibo [274] will be used for applying IQAE to the Fourier series.

In particular, the linear Ansatz corresponding to each layer $\mathcal{L}_{LA}^{(l)}(\vec{x}, \vec{\theta})$ for training the QNN to fit a M -dimensional function is the following [272]:

$$\mathcal{L}_{LA}^{(l)}(\vec{x}, \vec{\theta}) = \prod_{i=1}^M \mathcal{S}(x_i) \mathcal{A}_i^{(l)}(\vec{\theta}_{l,i}) , \quad (5.31)$$

where $\mathcal{S}(x_i)$ and $\mathcal{A}_i^{(l)}$ are chosen as:

$$\mathcal{S}(x_i) = R_z(x_i) , \quad \mathcal{A}_i^{(l)}(\vec{\theta}_{l,i}) = R_z(\theta_{l,i,1}) R_y(\theta_{l,i,2}) R_z(\theta_{l,i,3}) . \quad (5.32)$$

Once the Fourier coefficients are obtained from the QNN, we implement the IQAE algorithm. To uphold the claimed quantum advantage provided by Grover's amplitude amplification, certain conditions must be fulfilled. First, the probability distribution of the functions to be integrated should be encodable into a shallow quantum circuit. In view of this requirement, we will use the distribution $p(x_i) = 1/2^n$ generated by applying an n -dimensional Hadamard gate, denoted as $\mathcal{H}^{\otimes n}$, which corresponds to a quantum circuit of depth 1. The second condition is that the target function has to be encodable with a minimum number of quantum arithmetic operations. That will be achieved selecting the target function as a $\sin^2(x_i)$ to be integrated in $[x_{i,min}, x_{i,max}]$. Then the integrals of the Fourier terms are obtained from the integral of the sine function.

Under these considerations and choosing $n_{qubits} = 5$, the quantum circuits corresponding to the $\mathcal{A}$ and $\mathcal{Q}$ operators are shown in Fig. 5.5. Note that in Fig. 5.5(a) the rotation angles

encode the information about the limits of integration $x_{i,min}$ and $x_{i,max}$. In particular, they are defined as:

$$\theta_0 = (x_{max} - x_{min})/2^n + 2x_{min}, \quad \theta_i = 2^{(i+1)}(x_{max} - x_{min})/2^n, \quad n = n_{qubits}. \quad (5.33)$$

For more information, a tutorial on QFIAE implementation can be found at [265].

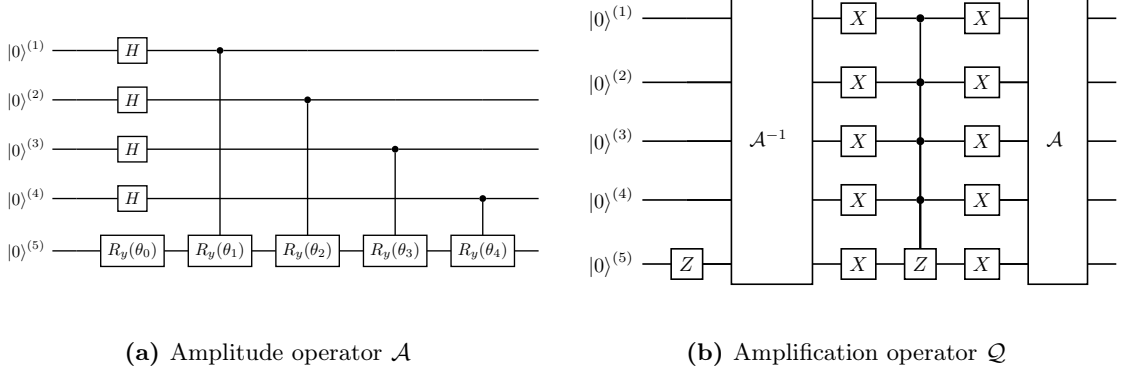


Figure 5.5: Quantum circuits of the operators for the IQAE component of the QFIAE algorithm.

5.4.1.2 Hardware implementation

Although the implementation of Monte Carlo integration in quantum simulators is of great interest for proof of concept purposes, the potential quantum advantage will only be materialized when the quantum algorithm is run on a real quantum device. To this aim, we have addressed the challenge of implementing an end-to-end quantum integrator method into two different quantum devices.

First, the QNN has been trained using an updated version of the Adam gradient descent method first presented in [278] and recently improved in [279]. In this new version of the Adam algorithm, the authors propose a Real-Time Quantum Error Mitigation (RTQEM) procedure, that allows to mitigate the noise in the QNN parameters during training. We use the full-stack Qibo [274] framework. The high-level algorithm has been written using Qibo, while Qibolab [280] and Qibocal [281] are used to respectively control and calibrate the 5-qubit superconducting quantum device hosted in the Quantum Research Centre (QRC) of the Technology Innovation Institute (TII).

In this case, a more hardware-friendly linear Ansatz has been chosen to construct the QNN in one qubit. In particular each layer $\mathcal{L}_{LA}^{(l)}(\vec{x}, \vec{\theta})$ is defined as:

$$\mathcal{L}_{LA}^{(l)}(\vec{x}, \vec{\theta}) = \prod_{i=1}^M R_z(\theta_3 x_i + \kappa \theta_4) R_y(\theta_1 x_i + \theta_2), \quad \text{with} \begin{cases} \kappa = 1 & \text{if } l \text{ is the last layer,} \\ \kappa = 0 & \text{otherwise.} \end{cases} \quad (5.34)$$

On the other hand, the IQAE has been executed using Qiskit [282] on the IBM Quantum 27-qubits device *ibmq_mumbai*. To mitigate quantum noise during the execution of this algorithm, we employed a pulse-efficient transpilation technique [283]. This technique effectively reduces the number of two-qubit gate operations by harnessing the hardware-native cross-resonance interaction, potentially leading to a reduction in quantum noise. Furthermore, we also applied two more error mitigation techniques, Dynamical Decoupling (DD) and Zero Noise Extrapolation (ZNE), which are automatized within the Qiskit Runtime Estimator primitive [284].

5.4.2 Tadpole loop integrals

Let us start with the simplest one-loop Feynman integral, a tadpole-like, see Fig. 5.6, where the Feynman propagator is raised to a given power,

$$\mathcal{A}_a^{(1)}(m) = \int_{\ell} (G_F(\ell))^a = \int_{\ell} \frac{1}{(\ell^2 - m^2 + i0)^a}, \quad a \in \mathbb{N}. \quad (5.35)$$

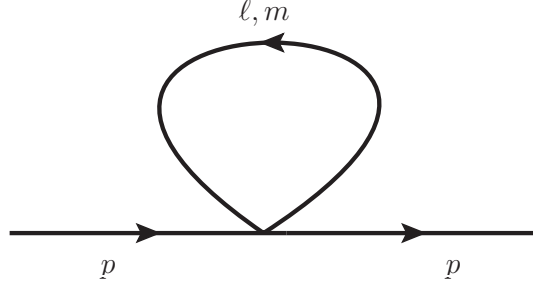


Figure 5.6: Tadpole Feynman diagram of an external particle with momentum p and an internal loop momentum ℓ with mass m .

For $a \in \{1, 2\}$, this integral is singular in the UV. So, we renormalize it by introducing a local UV counterterm

$$\mathcal{A}_a^{(1,R)}(m; \mu_{UV}) = \mathcal{A}_a^{(1)}(m) - \mathcal{A}_{UV,a}^{(1)}(\mu_{UV}), \quad a = \{1, 2\}, \quad (5.36)$$

where μ_{UV} is the renormalization scale. Regarding the integration measure, we have in d spacetime dimensions

$$\int_{\ell} \equiv -i \mu^{4-d} \int \frac{d^d \ell}{(2\pi)^d}. \quad (5.37)$$

We work with the LTD representation of the integrals in Eq. (5.35) and Eq. (5.36)

$$\mathcal{A}_a^{(1)}(m) = \int_{\ell} \frac{c_a}{(2q_{1,0}^{(+)})^{2a-1}}, \quad c_a = \{-1, 2, -6, 20, -70, \dots\}, \quad (5.38)$$

where the on-shell energy is $q_{1,0}^{(+)} = \sqrt{\ell^2 + m^2 - i0}$, and $q_{UV,0}^{(+)} = \sqrt{\ell^2 + \mu_{UV}^2 - i0}$ for the UV counterterm. The UV counterterms are

$$\mathcal{A}_{UV,1}^{(1)}(\mu_{UV}) = \int_{\ell} \frac{c_1}{2q_{UV,0}^{(+)}} \left(1 + \frac{\mu_{UV}^2 - m^2}{2(q_{UV,0}^{(+)})^2} \right), \quad (5.39)$$

$$\mathcal{A}_{UV,2}^{(1)}(\mu_{UV}) = \int_{\ell} \frac{c_2}{(2q_{UV,0}^{(+)})^3}. \quad (5.40)$$

Notice that $\mathcal{A}_1^{(1)}(m)$ is quadratically singular in the UV, so we need to subtract up to logarithmic order. Therefore, the extra term in the UV expansion in Eq. (5.39). The integration measure involves now only the spatial components of the loop momentum. In $d = 4 - 2\epsilon$ spacetime dimensions

$$\int_{\ell} \equiv \mu^{4-d} \int \frac{d^{d-1} \ell}{(2\pi)^{d-1}} = \frac{\mu^{2\epsilon}}{(2\pi)^{3-2\epsilon}} \int (\ell^2)^{1-\epsilon} d|\ell| d\Omega^{(2-2\epsilon)}. \quad (5.41)$$

As the integrals we consider are renormalized, we can fix the spacetime dimensions to $d = 4$, or $\epsilon = 0$. The integral does not depend on the solid angle, so the angular integration is straightforward, $\Omega^{(2)} = 4\pi$, and we are left with an integral in one variable, the modulus of the loop three-momentum.

This change of variable remaps the integration domain to the interval $[0, 1)$,

$$|\ell| = \frac{mz}{1-z}, \quad z = [0, 1). \quad (5.42)$$

These tadpole integrals can be computed using the QFIAE method. The integrals that are particularly interesting are those with $a = 1, 2$, since we have to deal with UV singularities by introducing appropriate UV counterterms. The corresponding analytic integrated expressions are

$$\mathcal{A}_1^{(1,R)}(m; \mu_{UV}) = \frac{1}{16\pi^2} \left(-m^2 \log \left(\frac{m^2}{\mu_{UV}^2} \right) + m^2 - \mu_{UV}^2 \right), \quad (5.43)$$

$$\mathcal{A}_2^{(1,R)}(m; \mu_{UV}) = \frac{1}{16\pi^2} \left(-\log \left(\frac{m^2}{\mu_{UV}^2} \right) \right). \quad (5.44)$$

5.4.2.1 Tadpole integration on simulator

Applying the QFIAE on the quantum simulators of **PennyLane** and **Qibo** for the QNN and the IQAE respectively, we obtain the results presented in Fig. 5.7.

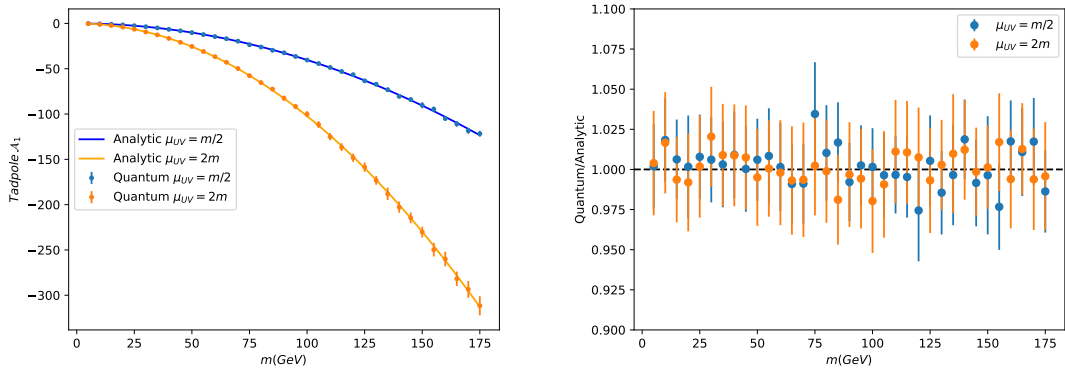


Figure 5.7: Quantum integration using QFIAE of the renormalized tadpole integral $\mathcal{A}_1^{(1,R)}(m; \mu_{UV})$ as a function of the mass m and the renormalization scale μ_{UV} (left) and comparison with the analytic result (right). The parameters used in the quantum implementation are: $max_steps = 300$, $step_size = 0.06$, $layers = n_{Fourier} = 5$ for the quantum Fourier part and $n_{qubits} = 5$, $n_{shots} = 10^3$, $\epsilon = 0.01$, $\alpha = 0.05$ for the IQAE part.

The Fig. 5.7 (left) shows the result of the quantum integration with QFIAE of the tadpole loop integral with $a = 1$ for different values of the mass m and the renormalization scale μ_{UV} . We do not show results for $a = 2$ because the integral is constant when we fix μ_{UV} as a multiple of m .

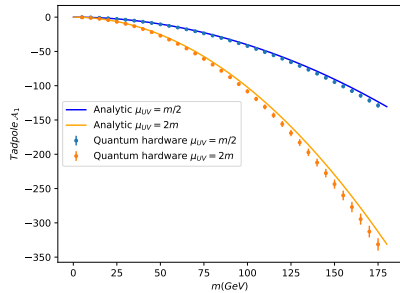
The uncertainties of the integrals have been calculated as the quadratic sum of the individual uncertainties provided by the IQAE algorithm when integrating every

trigonometric piece separately. Regarding the uncertainty of each integral, it has been set up beforehand to be smaller than $\epsilon = 0.01$ for each contribution with a confidence interval of $\alpha = 0.05$. These uncertainties are statistical. There will be also systematic errors in each integral due to the quantum estimation of the Fourier coefficients. In Fig. 5.7 (right) the accuracy of the results obtained in comparison to the analytical expression of the integral is depicted. Showing a relatively small deviation ($\sim 2\%$), which is in concordance with the uncertainties we have estimated, from the ideal scenario $\mathcal{A}_{\text{quantum}} = \mathcal{A}_{\text{analytical}}$.

5.4.2.2 Tadpole integration on hardware

Since the results obtained in the simulation are quite successful, we can go one step further and run QFIAE on hardware to compute for the first time a loop Feynman integral into a quantum computer.

To this aim, we have implemented the QFIAE on hardware. In particular, for the QNN part we have trained the model with considerably lighter hyperparameters than for simulation $n_{\text{data}} = 15$ points and we have used $\text{layers} = 3$, $\text{step_size} = 0.07$, $\text{max_steps} = 60$. However the results obtained show that with these values, the QNN is sufficiently expressive to fit the function in the integration domain $[0,1]$. In particular, since the access to the hardware is limited, the way to proceed here has been to fit the tadpole integrand and extract its Fourier series for an arbitrary value of the mass, $m_1 = 5$ and for $\mu_{UV} = m_1/2, 2m_1$ and then to obtain the Fourier coefficients for other values as a simple multiplication by the quotient m^2/m_1^2 . Since if we set $\mu_{UV} = km_1$, where k is a real number, we obtain the following relation, $\mathcal{A}_1^{(1,R)}(m)/\mathcal{A}_1^{(1,R)}(m_1) = (m/m_1)^2$, that allows us to extrapolate the results for the entire range of mass $[5,175]$ GeV. The results in Fig. 5.8 show a relatively small deviation from the analytical value for both $\mu_{UV} = m/2$ and $\mu_{UV} = 2m$. This represents a noteworthy achievement as it is the first application of an end-to-end quantum algorithm executed on a quantum computer for estimating loop integrals and the results obtained are satisfactory. Also, it further demonstrates the effectiveness of the error mitigation techniques applied, in particular: pulse transpilation, which involves decomposing circuits into hardware-native gates (as detailed in Sec.5.4.1.2). While this technique has shown promise in the context of Variational Quantum Circuits (VQC) [285], its potential for fault-tolerant applications remains largely unexplored. The positive results in Fig. 5.8 suggest that this work could serve as a valuable starting point for extending pulse transpilation to fault-tolerant quantum algorithms.



	ϵ (%) $\mu_{UV} = m/2$	ϵ (%) $\mu_{UV} = 2m$
CF & CI	0.0	0.1
CF & QI	5.2	2.8
QF & CI	9.4	3.3
QF & QI	4.4	5.9

Figure 5.8: (Left) Quantum integration on Qibo and IBM Quantum hardware of the renormalized tadpole integral $\mathcal{A}_1^{(1,R)}(p, m; \mu_{UV})$ as a function of the mass m and the renormalization scale μ_{UV} . (Right) Deviations from the analytical value of the integral when the Fourier (F) decomposition or the IQAE integration (I) are performed in a classical (C) or a quantum (Q) computer.

5.4.3 Bubble loop integral

Let us now consider the renormalized one-loop integral corresponding to a bubble topology, see Fig. 5.9,

$$\mathcal{A}^{(1,R)}(p, m_1, m_2; \mu_{UV}) = \mathcal{A}^{(1)}(p, m_1, m_2) - \mathcal{A}_{UV,2}^{(1)}(\mu_{UV}) , \quad (5.45)$$

where

$$\mathcal{A}_2^{(1)}(p, m_1, m_2) = \int_{\ell} \prod_{i=1}^2 G_F(q_i) = \int_{\ell} \frac{1}{(\ell^2 - m_1^2 + i0) ((\ell + p)^2 - m_2^2 + i0)} . \quad (5.46)$$

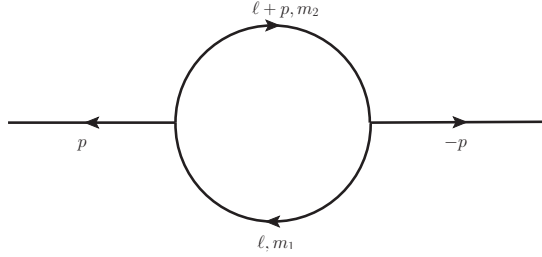


Figure 5.9: Bubble Feynman diagram with external momentum p and internal loop momenta $q_1 = \ell$ and $q_2 = \ell + p$ with masses m_1 and m_2 , respectively.

The UV counterterm is the same as for the tadpole integral with $a = 2$ in Eq. (5.40). The analytic expression with which we will compare the result of the numerical integration, for equal masses, is

$$\mathcal{A}_2^{(1,R)}(p, m, m; \mu_{UV}) = \frac{1}{16\pi^2} \left(-\log \left(\frac{m^2}{\mu_{UV}^2} \right) + \beta \log \left(\frac{\beta - 1}{\beta + 1} \right) + 2 \right) , \quad \beta = \sqrt{1 - \frac{4m^2}{p^2 + i0}} . \quad (5.47)$$

The LTD representation of Eq. (5.46) is [228]

$$\mathcal{A}_2^{(1)}(p, m_1, m_2) = \int_{\ell} \frac{1}{x_2} \left(\frac{1}{\lambda^+} + \frac{1}{\lambda^-} \right) , \quad (5.48)$$

where

$$x_2 = \prod_{i=1,2} 2q_{i,0}^{(+)} , \quad \lambda^{\pm} = \sum_{i=1,2} q_{i,0}^{(+)} \pm p_0 . \quad (5.49)$$

The on-shell energies are given by

$$q_{i,0}^{(+)} = \sqrt{\ell^2 + m_i^2 - i0} , \quad i = \{1, 2\} , \quad (5.50)$$

assuming the external momentum has vanishing spatial components, $p = (p_0, \mathbf{0})$.

If $p_0^2 < (m_1 + m_2)^2$ the integral is purely real. Otherwise, it gets an imaginary contribution from the unitary threshold singularity at $\lambda^- \rightarrow 0$, assuming $p_0 > 0$. To deal with this threshold singularity, it is convenient to introduce the following contour deformation in the complex plane, which will smooth the behavior of the function in the vicinity of the pole without altering the result of the integral,

$$\ell = \ell' \left(1 - \frac{i\delta}{\sqrt{\ell'^2}} \right) , \quad (5.51)$$

where δ is a parameter with dimensions of mass that controls the size of the deformation. With this transformation, the modulus of the loop momentum and the Jacobian of the transformation read

$$\ell^2 = \left(\sqrt{\ell'^2} - i\delta \right)^2, \quad \left| \frac{d^3\ell}{d^3\ell'} \right| = \frac{\ell^2}{\ell'^2}, \quad (5.52)$$

which is consistent with the $i0$ prescription of the on-shell energies.

To compute this bubble integral, we use the change of variable introduced in Eq. (5.42). Also, we consider $m_i = m$, for $i = 1, 2$. With these considerations, the way to proceed to estimate the integrals is using two different QNN to fit separately the real and the imaginary parts of the bubble integrand and then integrating each Fourier series using IQAE. The results obtained are depicted in Fig. 5.10, where the real and imaginary parts are displayed on the left and right plots, respectively. It is important to mention that

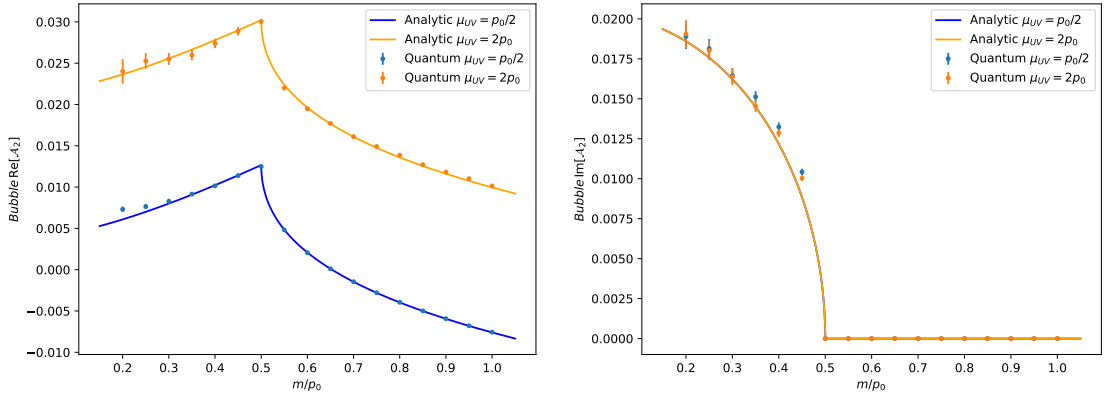


Figure 5.10: Quantum integration of the real (left) and imaginary (right) part of the renormalized bubble integral $\mathcal{A}_2^{(1,R)}(p, m, m; \mu_{UV})$ as a function of the ratio of the mass m to the energy component of the external momentum set at $p_0 = 100$ GeV, and the renormalization scale μ_{UV} . For the values $m/p_0 \geq 0.3$, the parameters used in the quantum implementation are: $max_steps = 200$, $step_size = 0.02$, whereas for $m/p_0 < 0.3$ more precision was needed, because of the threshold singularities, so the parameters used are: $max_steps = 450$, $step_size = 0.095$. For both cases, some parameters are equal: $layers = n_{Fourier} = 10$, $n_{qubits} = 1$ for the quantum Fourier part and $n_{qubits} = 5$, $n_{shots} = 10^3$, $\epsilon = 0.001$, $\alpha = 0.05$ for the IQAE part.

the parameter δ has been set to $\delta = 0.21\sqrt{s}$, to enhance the smoothness of the function around the threshold singularity. This choice helps the QNN in its task of fitting the function more effectively. The uncertainties are calculated as in the previous section. It is worth mentioning that uncertainties in the region $m/p_0 < 0.3$, appear to be larger compared to the rest. This is explained by the Fourier coefficients for low frequencies being larger than in the region $m/p_0 \geq 0.3$. Hence the statistical uncertainties of the integrals of the trigonometric functions for low frequencies are intrinsically larger since the integrals we are estimating are also larger, i.e have a larger weight, which is the Fourier coefficient. Another interesting point about these results is that for the mentioned region the QNN fits the target function with a slight drop in performance, hence there seems to be a correlation between the QNN struggling to fit a function and the coefficients of the lower frequencies terms being larger.

Despite this, it is remarkable that the quantum integration values are in agreement with the analytical values within uncertainties. This constitutes another significant achievement

since we have successfully circumvented the threshold singularity while applying a quantum algorithm to obtain an estimation of the bubble integral.

5.4.3.1 Bubble tensor loop integral

We now consider the following tensor bubble integral:

$$\mathcal{A}_{2,T}^{(1)}(p, m_1, m_2) = \int_{\ell} (\ell \cdot p)^2 \prod_{i=1}^2 G_F(q_i) = \int_{\ell} \frac{(\ell \cdot p)^2}{(\ell^2 - m_1^2 + i0) ((\ell + p)^2 - m_2^2 + i0)} . \quad (5.53)$$

Proceeding in a similar way that with the scalar bubble integral we obtain the results presented in Fig. 5.11

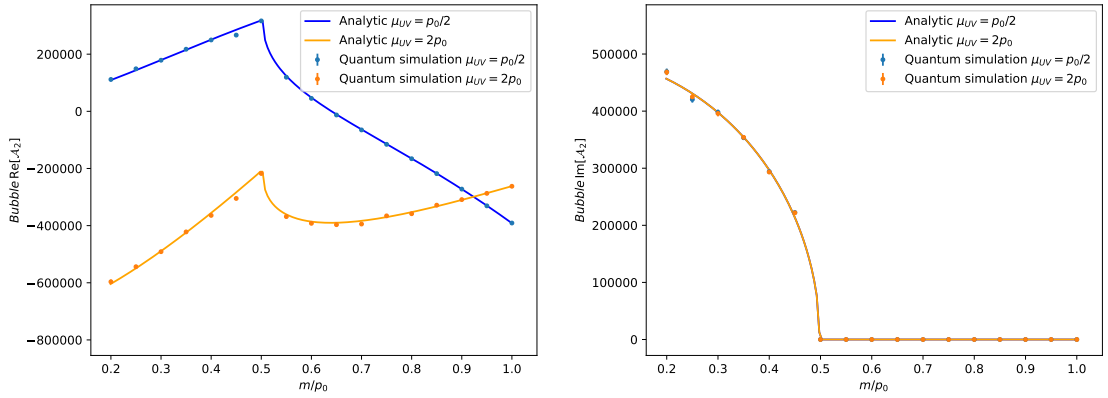


Figure 5.11: Quantum integration of the real (left) and imaginary (right) part of the renormalized tensor bubble integral $\mathcal{A}_{2,T}^{(1,R)}(p, m, m; \mu_{UV})$ as a function of the ratio of the mass m to the energy component of the external momentum set at $p_0 = 100$ GeV, and the renormalization scale μ_{UV} . The parameters used in the quantum implementation are: $max_steps = 5000$, $step_size = 0.0001$, $layers = n_{Fourier} = 8$, $n_{qubits} = 2$ for the quantum Fourier part and $n_{qubits} = 5$, $n_{shots} = 10^3$, $\epsilon = 0.001$, $\alpha = 0.05$ for the IQAE part.

5.4.4 Triangle loop integral

We now consider the one-loop three-point function, which corresponds to a triangle loop topology, see Fig. 5.12,

$$\mathcal{A}_3^{(1)}(p_1, p_2, m_1, m_2, m_3) = \int_{\ell} \prod_{i=1}^3 G_F(q_i) , \quad (5.54)$$

where $q_1 = \ell + p_1$, $q_2 = \ell + p_{12}$ and $q_3 = \ell$, with $p_{12} = p_1 + p_2$, and $p_{12}^2 = s$. This integral is UV finite and does not require any renormalization. The analytic expression for the integrated result reads

$$\mathcal{A}_3^{(1)}(p_1, p_2, m, m, m) = -\frac{1}{16\pi^2 s} \left[\text{Li}_2 \left(\frac{2}{1-\beta} \right) + \text{Li}_2 \left(\frac{2}{1+\beta} \right) \right] , \quad \beta = \sqrt{1 - \frac{4m^2}{s + i0}} , \quad (5.55)$$

The causal LTD representation is given by [248]

$$\mathcal{A}_3^{(1)}(p_1, p_2, m_1, m_2, m_3) = -\int_{\ell} \frac{1}{x_3} \left(\frac{1}{\lambda_{12}^- \lambda_{23}^+} + \frac{1}{\lambda_{23}^+ \lambda_{31}^-} + \frac{1}{\lambda_{31}^- \lambda_{12}^+} + (\lambda_{ij}^+ \leftrightarrow \lambda_{ij}^-) \right) , \quad (5.56)$$

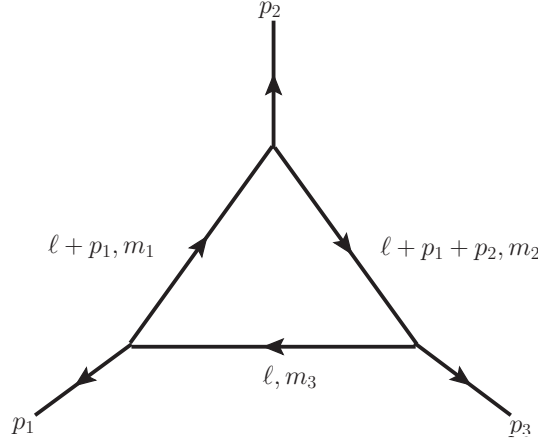


Figure 5.12: Triangle Feynman loop diagram. The external particles, with momentum $\{p_i\}_{i=1,2,3}$, are considered massless. The internal loop momenta are $q_1 = \ell + p_1$, $q_2 = \ell + p_1 + p_2$, and $q_3 = \ell$ with masses m_1, m_2, m_3 respectively.

with $x_3 = \prod_{i=1}^3 2q_{i,0}^{(+)}$, where now $q_{1,0}^{(+)} = \sqrt{(\ell + \mathbf{p}_1)^2 + m_1^2 - i0}$, and $q_{2,0}^{(+)}$ and $q_{3,0}^{(+)}$ are still described by Eq. (5.50) if we work in the center of mass (cms) frame where $\mathbf{p}_{12} = 0$ and p_1 and p_2 are back-to-back along the z axis, $\mathbf{p}_1 = p_{1,0}(0, 0, 1)$ and $\mathbf{p}_2 = p_{2,0}(0, 0, -1)$. The causal denominators are

$$\lambda_{12}^{\pm} = q_{1,0}^{(+)} + q_{2,0}^{(+)} \pm p_{2,0}, \quad \lambda_{23}^{\pm} = q_{2,0}^{(+)} + q_{3,0}^{(+)} \mp p_{12,0}, \quad \lambda_{31}^{\pm} = q_{3,0}^{(+)} + q_{1,0}^{(+)} \pm p_{1,0}. \quad (5.57)$$

The integration variables in Eq. (8.33) are the modulus of the loop three-momentum and its polar angle with respect to p_1 , considering that the azimuthal integration is trivial. This means that the Fourier decomposition is a function of two variables and we have to integrate each of them separately. We apply the change of variable in Eq. (5.42) on the modulus of the loop three-momentum, and the contour deformation in Eq. (5.51) to deal with the unitary threshold singularity at $\lambda_{23}^+ \rightarrow 0$, when $m < \sqrt{s}/2$. The contour deformation smoothes the behavior of the integrand over the threshold singularity and therefore significantly improves the quality of the Fourier decomposition. In this case we set $\delta = 0.1\sqrt{s}$.

The obtained results are shown in Figure 5.13 (left) and (right), illustrating the real and imaginary components, respectively. The uncertainties are also calculated as in the previous sections but are expected to be higher since we are performing a double integral by the IQAE method, and each integration introduces an error. However, the estimated uncertainties align with the deviations observed in the real and imaginary components of the integral. All in all, this represents another noteworthy accomplishment as we have successfully extended for the first time the QFIAE method to a two-dimensional function with a threshold singularity, and utilized a quantum algorithm to approximate the triangle integral with sufficient accuracy.

5.4.4.1 Triangle tensor loop integral

We now consider the following tensor triangle integral:

$$\mathcal{A}_{3,T}^{(1)}(p_1, p_2, m_1, m_2, m_3) = \int_{\ell} (\ell \cdot p) \prod_{i=1}^3 G_F(q_i), \quad (5.58)$$

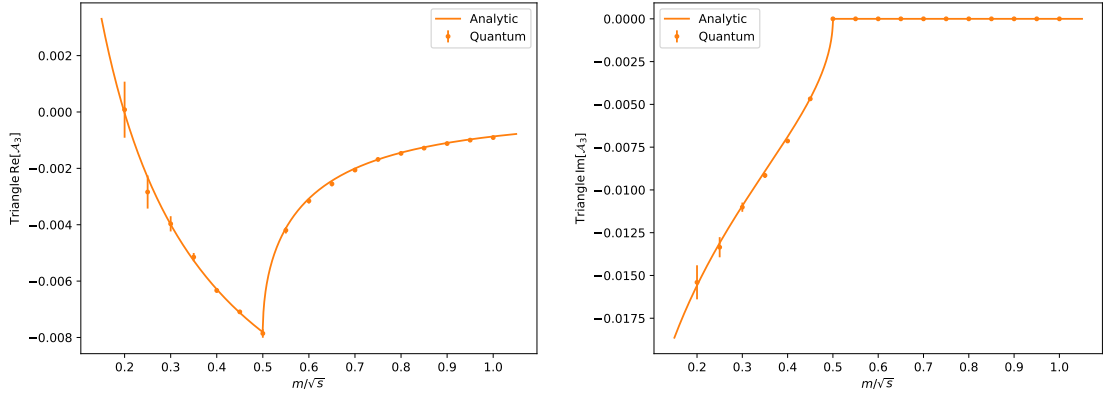


Figure 5.13: Quantum integration of the real (left) and imaginary (right) part of the triangle integral $\mathcal{A}_3^{(1)}(p_1, p_2, m, m, m)$ as a function of the ratio of the mass m to the cms energy set at $\sqrt{s} = 2$ GeV and comparison with the analytical result. For the values $m/\sqrt{s} \geq 0.5$, the parameters used in the quantum implementation are: $max_steps = 400$, $step_size = 0.05$, whereas for $m/\sqrt{s} \leq 0.5$ more precision was needed, because of the threshold singularity, so the parameters used are: $max_steps = 450$, $step_size = 0.065$. For both cases, some parameters are equal: $layers = n_{Fourier} = 10$, $n_{qubits} = 1$ for the QNN and $n_{qubits} = 5$, $n_{shots} = 10^4$, $\epsilon = 0.001$, $\alpha = 0.05$ for the IQAE part.

Proceeding in a similar way that with the scalar bubble integral we obtain the results presented in Fig. 5.14

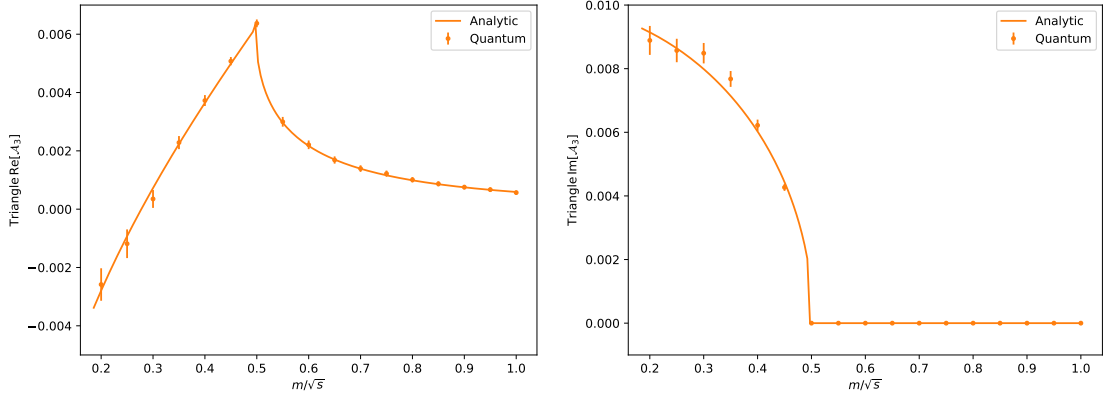


Figure 5.14: Quantum integration of the real (left) and imaginary (right) part of the renormalized tensor triangle integral $\mathcal{A}_{3,T}^{(1,R)}(p, m, m; \mu_{UV})$ as a function of the ratio of the mass m to the energy component of the external momentum set at $p_0 = 2$ GeV. The parameters used in the quantum implementation are: $max_steps = 15000$, $step_size = 0.001$, $layers = n_{Fourier} = 20$, $n_{qubits} = 6$ for the quantum Fourier part and $n_{qubits} = 5$, $n_{shots} = 10^3$, $\epsilon = 0.01$, $\alpha = 0.05$ for the IQAE part.

5.4.5 Pentagon loop integral

Five-point integrals are important because they represent a special case that is not trivial to solve (even in the fully massless case). Scalar pentagon integrals have been studied intensively in recent years [286, 287, 288, 289, 290]. Also, cases where the pentagon is free of divergences have been studied many years ago [291, 292, 293, 294]. For some applications to next to leading order (NLO) quantum chromodynamics (QCD) phenomenology, it is

possible to reduce the five-point scalar integral to a set of boxes. But beyond NLO, the reduction of pentagon integrals also requires five-point integrals in higher dimensions.

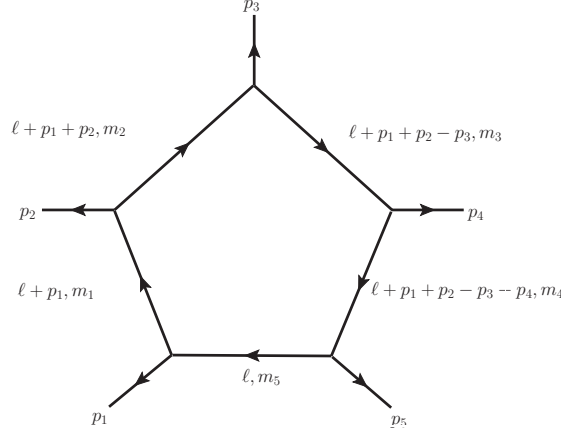


Figure 5.15: One-loop pentagon Feynman diagram with massless external momenta $\{p_i\}_{i=1}^5$ particle with momentum p_1, p_2, p_3, p_4, p_5 and massive internal loop momenta with masses $\{m_i\}_{i=1}^5$. $q_i = \{\ell + p_1, \ell + p_1 + p_2, \ell + p_1 + p_2 - p_3, \ell + p_1 + p_2 - p_3 - p_4, \ell\}$

If the number of scales (internal and external masses) is not too large, the resulting integral can be obtained in terms of hypergeometric functions [286]. However, in cases with several scales, this could not be solved trivially in terms of known functions. For this reason, in the most general case, the use of universal numerical approaches is of interest. In particular, we consider the following one-loop scalar pentagon integral,

$$\mathcal{A}_5^{(1)}(\{p_i, m_i\}_{i=1,\dots,5}) = \int_{\ell} \prod_{i=1}^5 G_F(q_i) , \quad (5.59)$$

which is depicted in Fig. 5.15, where

$$q_j = \ell + \sum_{i=1}^j p_i , \quad (5.60)$$

with $q_5 = \ell$ by momentum conservation. The LTD representation is obtained with Lotty [295],

$$\mathcal{A}_5^{(1)}(\{p_i, m_i\}_{i=1,\dots,5}) = \int_{\ell} \frac{1}{x_5} \sum \left(L_{ij}^+ L_{kl}^- + L_{ij}^- L_{kl}^+ \right) , \quad (5.61)$$

where $x_5 = \prod_{i=1}^5 2q_{i,0}^{(+)}$ with $q_{i,0}^{(+)} = \sqrt{\mathbf{q}_i^2 + m_i^2} - i0$, and

$$L_{ij}^{\pm} = \left(\frac{1}{\lambda_i^{\pm}} + \frac{1}{\lambda_j^{\pm}} \right) \frac{1}{\lambda_{ij}^{\pm}} , \quad (5.62)$$

where the causal denominators are defined as

$$\begin{aligned} \lambda_i^{\pm} &= q_{i,0}^{(+)} + q_{i+1,0}^{(+)} \pm p_{i,0} , \\ \lambda_{ij}^{\pm} &= q_{i,0}^{(+)} + q_{j+1,0}^{(+)} \pm (p_i + p_j)_0 , \quad j = i + 1 , \\ \lambda_{ij}^{\pm} &= \lambda_i^{\pm} + \lambda_j^{\pm} , \quad j = i + 2 . \end{aligned} \quad (5.63)$$

It is understood that the indices of the on-shell energies are defined cyclically, namely $i = n + 1$ with $n = 5$ is equivalent to $i = 1$. The loop momentum is parametrized in terms

of the polar and azimuthal angles, $\ell = |\ell|(\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$. The integrand in Eq. (5.61) depends on three independent integration variables.

We consider the specific kinematic configuration P11 defined in [255] and compare our results with the numerical values obtained therein. This specific configuration corresponds to:

$$\begin{aligned} p_1 &= (33.74515, 45.72730, 31.15254, -7.47943) , \\ p_2 &= (31.36435, -41.50734, 46.47897, 2.04203) , \\ p_3 &= (4.59005, 17.07010, 32.65403, 41.93628) , \\ p_4 &= (29.51054, -28.25963, 46.17333, -35.08918) , \\ m_1 &= m_2 = m_3 = m_4 = m_5 = 5.01213 . \end{aligned} \tag{5.64}$$

The results presented in Table 5.3 demonstrate the successful acquisition of the integral result using the quantum method. Although the quantum uncertainty may seem much bigger than the uncertainty estimated by classical methods it could be reduced by adjusting the ϵ parameter in the IQAE algorithm. This accomplishment is significant as it involves a three-dimensional function integrated through a quantum Monte Carlo approach, potentially paving the way for tackling two-loop or higher-loop integrals in a broader sense, which involve functions with four or more variables.

	Method	Result
P11	FQMCI	$-1.24(4) \cdot 10^{-13}$
	LTD _{class}	$-1.24027(16) \cdot 10^{-13}$

Table 5.3: Quantum integration of a pentagon configuration using the FQMCI method, and comparison with a classical numerical integration [255]. With $n_{\text{Fourier}} = 10$, $n_{\text{qubits}} = 5$, $n_{\text{shots}} = 10^4$, $\epsilon = 0.001$, $\alpha = 0.05$. The uncertainty for the quantum integration has been calculated as in previous sections.

Although the three-dimensional pentagon integral could also be computed using QFIAE, the linear Ansätze considered in Sec. 5.4.1 were not expressive enough to accurately capture the behavior of the pentagon functions. Exploring more expressive Ansätze that ensure efficient trainability for multivariate functions remains an avenue for future work.

5.5 Quantum decay rates at second order in perturbation theory

In this section, we make use of a recently proposed novel approach based on LTD to efficiently recast perturbative theoretical predictions at high-energy colliders, the LTD causal unitary [296, 4], where differential observables, cross sections and decay rates are assembled from the LTD representation of vacuum amplitudes, i.e. scattering amplitudes without external particles. We then combine this new approach of rewriting physical observables in terms of vacuum amplitudes with the QFIAE to perform numerical integration. Therefore, the aim of this section is going one step further and test the performance of QFIAE with physical decay rates at second order in perturbation theory or next-to-leading order (NLO). This involves combining one-loop and tree-level contributions, each individually singular and numerically challenging, though their sum remains finite. The LTD causal unitary approach provides a unified treatment of loop and tree-level terms, leading to relatively flat integrands that are well-suited for numerical integration, particularly through Fourier decomposition.

5.5.1 Causal unitary and decay rates at NLO

A vacuum amplitude in the LTD framework, $\mathcal{A}_D^{(\Lambda)}$, with Λ denoting the number of independent loop four-momenta, is obtained from its Feynman representation by applying the Cauchy residue theorem to integrate out one component of each loop momentum [229, 230]. This is typically done for the energy components, leading to a reformulation where Feynman propagators are replaced by causal propagators of the form [228]

$$\frac{1}{\lambda_{i_1 \dots i_m}} = \left(\sum_{s=1}^m q_{i_s,0}^{(+)} \right)^{-1}, \quad (5.65)$$

where $q_{i_s,0}^{(+)} = \sqrt{\mathbf{q}_{i_s}^2 + m_{i_s}^2} - i0$ represents the on-shell energy of each internal propagator, with $\mathbf{q}_{i_s}$ being the spatial part of the four-momentum and m_{i_s} its corresponding mass. The numerator of the vacuum amplitude in LTD depends on these on-shell energies as well as the internal masses. The infinitesimal imaginary term in the on-shell energy originates from the prescription used in Feynman propagators. While loop vacuum amplitudes in conventional Feynman representations are defined over a Minkowski space in terms of four-momenta, their LTD counterparts are formulated in an Euclidean space, involving loop three-momenta.

Each causal propagator in Eq. (5.65) corresponds to a subset of internal particles that partitions the vacuum amplitude into two subamplitudes, ensuring that the momenta of all particles within the subset flow in the same direction. Consequently, each term in the vacuum amplitude is a product of causal propagators, maintaining this alignment of momentum flow for shared particles [248, 249, 250, 297, 251, 252]. This structure is analogous to the selection of acyclic configurations in a directed graph, as encountered in graph theory [126, 127, 128, 129]. When a causal propagator becomes singular, all associated particles go on shell. This observation provides a systematic way to generate interference terms in scattering amplitudes with different numbers of final-state particles and loops by taking residues on causal propagators. This is the core idea of LTD causal unitarity [296, 4]. The vacuum amplitude in LTD acts as a kernel amplitude, which generates all the final states that contribute to a scattering or decay process from all possible residues on causal propagators.

As benchmark decay rates at NLO, in this section we analyze the decay of a heavy scalar into lighter scalars, as well as the decay of a Higgs boson or an off-shell photon into a pair of massive quarks and antiquarks. These processes have been implemented as a proof of concept for LTD causal unitarity in Ref. [4], where classical integration techniques were used to compute the total decay rates. A detailed discussion of the expressions utilized in the numerical implementation can be found in Ref. [4]. The vacuum diagrams relevant to the decay process $\gamma^* \rightarrow q\bar{q}(g)$ are depicted in Fig. 5.16. Analogous vacuum diagrams describe the other two decay channels under consideration.

The vacuum diagrams in Fig. 5.16 contribute to a vacuum amplitude that, in LTD, depends on the three loop three-momenta $\{\ell_1, \ell_2, \ell_3\}$. The momenta of the internal propagators are given by

$$\begin{aligned} \mathbf{q}_1 &= \ell_1 + \ell_2, & \mathbf{q}_2 &= \ell_1 + \ell_3, & \mathbf{q}_3 &= \ell_1, \\ \mathbf{q}_4 &= \ell_2, & \mathbf{q}_5 &= \ell_2 - \ell_3, & \mathbf{q}_6 &= \ell_3, \end{aligned}$$

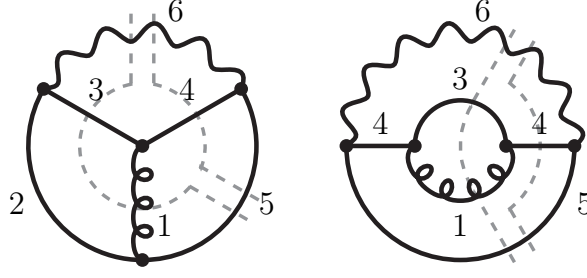


Figure 5.16: Three-loop vacuum diagrams contributing to the decay $\gamma^* \rightarrow q\bar{q}(g)$ at NLO. The gray dashed lines indicate phase-space residues, corresponding to different final states. Similar diagrams contribute to the decays $H \rightarrow q\bar{q}(g)$ and $\Phi \rightarrow \phi\phi(\phi)$, where the photon labeled 6 is replaced by a Higgs boson or a heavy scalar Φ . In the case of heavy scalar decay, particles 1 to 5 are substituted by light scalars.

with on-shell energies $q_{i,0}^{(+)} = \sqrt{\mathbf{q}_i^2 + m_i^2} - i0$. Working in the rest frame of the decaying particle, we set $\ell_3 = \mathbf{0}$, making the unintegrated decay rate dependent only on ℓ_1 and ℓ_2 through the on-shell energies.

At NLO, the differential decay rate of a particle a is expressed as

$$d\Gamma_a^{(1)} = \frac{d\Phi_{\ell_1\ell_2}}{2\sqrt{s}} \left[\left(\mathcal{A}_D^{(3,a)}(456) \tilde{\Delta}_{45\bar{6}} + \mathcal{A}_D^{(3,a)}(1356) \tilde{\Delta}_{135\bar{6}} \right) + (5 \leftrightarrow 2, 4 \leftrightarrow 3) \right], \quad (5.66)$$

where $\mathcal{A}_D^{(3,a)}(456)$ and $\mathcal{A}_D^{(3,a)}(1356)$ are phase-space residues of the vacuum amplitude in LTD. These are obtained from the residues of the corresponding causal propagators at $\lambda_{456} \rightarrow 0$ and $\lambda_{1356} \rightarrow 0$, representing quantum fluctuations at one-loop with two final-state particles and at tree-level with three final-state particles, respectively.

For the decay of a heavy scalar Φ into lighter scalars, these phase-space residues take the form

$$\begin{aligned} \mathcal{A}_D^{(3,\Phi)}(456) &= \frac{g_\Phi^{(1)} m_\Phi^2}{x_{12345}} \left(L_{2345,125}^{13\bar{4}} + L_{2345,134}^{12\bar{5}} + L_{134,125}^{2345} \right), \\ \mathcal{A}_D^{(3,\Phi)}(1356) &= \frac{g_\Phi^{(1)} m_\Phi^2}{x_{135}} \left(\frac{1}{\lambda_{13\bar{4}} \lambda_{134} \lambda_{1\bar{2}5} \lambda_{125}} \right), \end{aligned}$$

where $g_\Phi^{(1)}$ collects the interaction couplings, $x_{i_1 \dots i_n} = \prod_{s=1}^n 2q_{i_s,0}^{(+)}$ is the product of on-shell energies, and $L_{j,k}^i = \lambda_i^{-1} (\lambda_j^{-1} + \lambda_k^{-1})$. The quantity

$$\lambda_{i_1 \dots i_r \bar{i}_{r+1} \dots \bar{i}_n} = \lambda_{i_1 \dots i_r} - \lambda_{i_{r+1} \dots i_n} \quad (5.67)$$

encodes the structure of the propagator residues.

The integration measure is expressed in terms of two loop three-momenta:

$$d\Phi_{\ell_1\ell_2} = \prod_{j=1}^2 \frac{d^3\ell_j}{(2\pi)^3}, \quad (5.68)$$

involving six integration variables. However, each term in Eq. (5.66) must satisfy energy conservation, encoded as

$$\tilde{\Delta}_{i_1 \dots i_n \bar{a}} = 2\pi \delta(\lambda_{i_1 \dots i_n \bar{a}}), \quad (5.69)$$

and the decay is isotropic in the rest frame of the decaying particle. As a consequence, the decay rate depends on two independent integration variables, constrained by the Dirac delta functions in Eq. (5.69). These are a polar angle, representing the angle between the two loop three-momenta, typically parametrized as $\cos \theta = 1 - 2v$, with $v \in [0, 1]$, and the modulus of one of the loop three-momenta, mapped from $[0, \infty)$ to the finite interval $[0, 1]$ in numerical implementations. Explicitly,

$$d\Phi_{\ell_1 \ell_2} \rightarrow \frac{1}{4\pi^4} \int_0^\infty \ell_1^2 d|\ell_1| \int_0^\infty \ell_2^2 d|\ell_2| \int_0^1 dv, \quad (5.70)$$

where

$$\tilde{\Delta}_{45\bar{6}} = 2\pi \delta\left(\sqrt{\ell_2^2 + m^2} - \sqrt{s}\right), \quad (5.71)$$

and

$$\tilde{\Delta}_{135\bar{6}} = 2\pi \delta\left(|\ell_1 + \ell_2| + \sqrt{\ell_1^2 + m^2} + \sqrt{\ell_2^2 + m^2} - \sqrt{s}\right), \quad (5.72)$$

with $|\ell_1 + \ell_2| = \sqrt{\ell_1^2 + \ell_2^2 + 2(1 - 2v)|\ell_1||\ell_2|}$ and $|\ell_i| = \sqrt{\ell_i^2}$. The two Dirac delta functions in Eq. (5.71) and Eq. (5.72) allow one of the integration variables in Eq. (5.70) to be expressed in terms of the remaining two, which is solved analytically in numerical implementations.

A key feature of Eq. (5.66) is that loop and tree-level contributions—corresponding to different numbers of final-state particles—are treated under the same integration measure. This ensures local cancellation of singularities encountered in state-of-the-art methods, avoiding the need for intermediate calculations in arbitrary spacetime dimensions [263, 264]. Additionally, the resulting integrand is flatter than in other approaches, allowing for a much faster and more efficient numerical implementation, making it particularly suitable for QFIAE integration.

5.5.2 Quantum integration of NLO decay rates

In this section, we apply the quantum integration algorithm QFIAE to estimate the total decay rate at NLO for the decay processes discussed in Sec. 5.5.1. The main challenge in this quantum implementation lies in designing a Quantum Neural Network (QNN) that can effectively model the differential decay rate function. To address this, we introduce a QNN with a general-purpose Ansatz, as shown in Fig. 5.17. This Ansatz is designed with sufficient entanglement and free parameters, offering high expressibility while ensuring efficient trainability, thus enabling the accurate solution of the regression problem.

We use PennyLane [61] and JAX [298] to build and train the QNN. The architecture, shown in Fig. 5.17, consists of a 6-qubit quantum circuit where a specific Ansatz is repeated n_{layers} times. This Ansatz includes two main components: a variational layer with trainable parameters, implemented using `qml.StronglyEntanglingLayers`, and an encoding block for two input variables. Each variable is encoded three times in parallel across the 6 qubits using `qml.AngleEmbedding`.

Following the results from Sections 5.3 and 5.4 on regression with variational quantum circuits, encoding is performed using R_x rotations, and measurements are taken in the Pauli-Z basis, as illustrated in Fig. 5.17. In terms of complexity, each layer has a quantum depth of 7, consisting of one encoding step, one variational gate step, and five entangling two-qubit gate steps. For the integrated decay rates presented in Fig 5.18, we employ 20

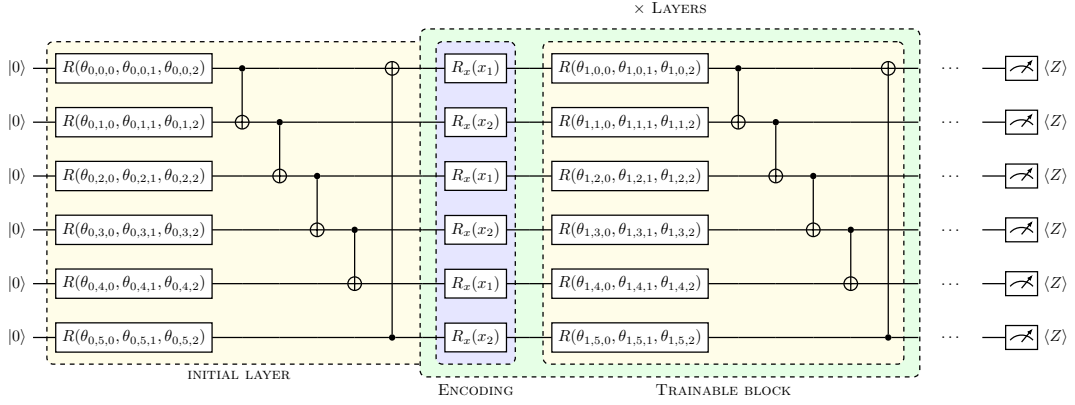


Figure 5.17: Architecture of the QNN employed to fit a 2-dimensional function.

layers, resulting in a total quantum depth of 140. To assess the feasibility of such variational circuit to be executed in current devices, we refer to two recent IonQ studies [276, 277], mentioned in Sec. 5.3, where various quantum algorithms were evaluated using QED-C benchmarks. The findings show that our algorithm, which requires a quantum depth of 140 and a low qubit count (≤ 6), would achieve a high success probability on IonQ and Quantinuum devices.

Once the QNN successfully approximates the target function, we extract its Fourier series and use it as input for the IQAE subroutine. The IQAE module is designed to operate with low quantum depth and a minimal number of qubits, making it feasible for execution on current quantum hardware. It is implemented with **Qibo**[274] for quantum simulations (Fig. 5.18 left) and with **Qiskit**[84] on real hardware (Fig. 5.18 right). Specifically, we run the IQAE module on the 27-qubit IBM Quantum superconducting processor *ibmq_mumbai*, using only 5 qubits to sequentially integrate the Fourier terms.

To mitigate quantum noise, we apply the same techniques that in Sec. 5.4. First, pulse-efficient transpilation [283], which reduces the number of two-qubit gates by leveraging hardware-native cross-resonance interactions. Additionally, we use Dynamical Decoupling (DD) during circuit execution and apply Zero Noise Extrapolation (ZNE) to the output via the **Qiskit** Runtime Estimator primitive [284].

Regarding the hyperparameters employed for this application, for training the QNN we use *max_steps*, which sets the number of iterations for the ADAM optimizer [299], and *step_size*, which represents the learning rate. The parameter *layers* specifies the number of circuit layers, while *n_{Fourier}* determines the number of Fourier coefficients used in the truncated representation of the circuit. The number of qubits in the variational circuit is defined by *n_{qubits}*. For the IQAE module, *n_{qubits}* specifies the number of qubits used, while *n_{shots}* sets the number of measurement samples per circuit run. The error tolerance for each integral is controlled by ϵ , and α defines the confidence interval for the integral results.

The results in Fig. 5.18 show a relatively small deviation from their corresponding analytical values in standard dimensional regularization (DREG). Notably, the left panel presents a systematic deviation compared to the right panel, attributed to hardware noise that remains despite the applied error mitigation techniques. Table 5.4 presents the explicit numerical results and uncertainties corresponding to these figures.

Table 5.4 shows that IQAE executions on quantum hardware present uncertainties about an order of magnitude larger than those on a quantum simulator. This difference is expected,

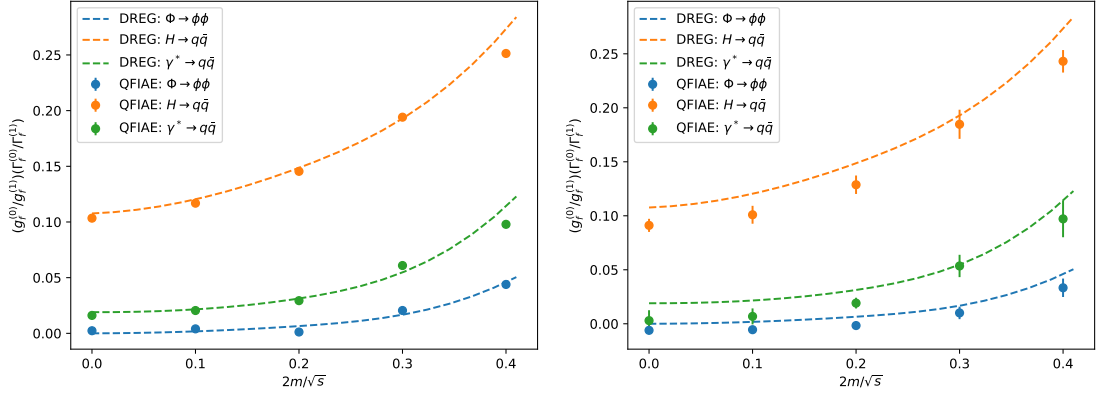


Figure 5.18: Quantum-integrated decay rates for the three processes $H \rightarrow q\bar{q}(g)$, $\gamma^* \rightarrow q\bar{q}(g)$, and $\Phi \rightarrow \phi\phi(\phi)$ at NLO as a function of the final state mass. The left panel shows results from a quantum simulator, while the right panel presents results where the IQAE has been executed on quantum hardware. The calculations use QFIAE with the LTD causal unitary approach, and the dashed lines represent theoretical predictions in dimensional regularization. Parameters for the quantum implementation: $max_steps = 15000$, $step_size = 0.001$, $layers = n_{Fourier} = 20$, $n_{qubits} = 6$ for the QNN, and $n_{qubits} = 5$, $n_{shots} = 10^3$, $\epsilon = 0.01$, $\alpha = 0.05$ for the IQAE module.

Decay	$2m/\sqrt{s}$	Hardware	Simulator	DREG
$\Phi \rightarrow \phi\phi(\phi)$	0.0	-0.0061(28)	0.0023(5)	0.0000
	0.1	-0.0055(31)	0.0040(6)	0.0018
	0.2	-0.0016(30)	0.0011(6)	0.0065
	0.3	0.0101(56)	0.0205(11)	0.0167
	0.4	0.0333(85)	0.0439(15)	0.0459
$H \rightarrow q\bar{q}(g)$	0.0	0.0911(61)	0.1034(13)	0.1077
	0.1	0.1009(83)	0.1169(14)	0.1204
	0.2	0.1288(85)	0.1455(14)	0.1486
	0.3	0.1847(135)	0.1941(20)	0.1928
	0.4	0.2431(104)	0.2513(30)	0.2730
$\gamma^* \rightarrow q\bar{q}(g)$	0.0	0.0029(96)	0.0161(14)	0.0190
	0.1	0.0068(74)	0.0205(13)	0.0215
	0.2	0.0191(50)	0.0293(13)	0.0313
	0.3	0.0535(103)	0.0609(20)	0.0547
	0.4	0.0971(171)	0.0979(30)	0.1140

Table 5.4: Quantum-integrated decay rates at NLO for the processes $H \rightarrow q\bar{q}(g)$, $\gamma^* \rightarrow q\bar{q}(g)$, and $\Phi \rightarrow \phi\phi(\phi)$ as a function of the final state mass, computed using QFIAE and LTD causal unitary. The ‘Hardware’ column reports results from the QNN on a quantum simulator and the IQAE on quantum hardware, while the ‘Simulator’ column shows results where both the QNN and IQAE run on quantum simulators. The ‘DREG’ column provides the exact analytic results at NLO accuracy.

as the inherent quantum noise on real hardware adds to the statistical uncertainty of the IQAE method. Despite this, most of the results are in agreement within the uncertainty bands with the expected values, demonstrating satisfactory agreement given the current limitations of quantum hardware.

5.6 Conclusions

In this Chapter we have discussed a new application of QML and QAE to build a novel Quantum Monte Carlo integrator, called QFIAE [2, 3, 5]. This method goes beyond previously defined quantum integrators such as FQMCI, since it exploits QNN to approximate the Fourier series. Furthermore, QFIAE eliminates the need for numerical integration to compute Fourier coefficients for a general multivariate function $f(\vec{x})$, potentially preserving the quadratic quantum speedup achieved through amplitude amplification (a generalization of Grover’s algorithm), where other methods fall short. This constitutes, to the best of our knowledge, the very first end-to-end Quantum Monte Carlo integrator with the potential to maintain this speedup.

As an initial testbed, we have successfully applied the quantum integrator to a one-dimensional function that corresponds to the elementary scattering process $e^+e^- \rightarrow q\bar{q}$. We obtained a very precise estimation of the integral with a relative error around 1%, while maintaining a relatively low quantum depth. In particular, a study of the performance and the quantum depth of both circuits of the algorithm has been conducted and have been compared with the depth and width of the algorithms that can be effectively implemented on NISQ devices. The results showed that our method lies in the region where a quantum algorithm can be executed with a high probability of success.

Then, we considered more complicated integrals and we have applied the QFIAE to estimate loop Feynman integrals for the first time. Our primary objective was to explore the efficiency of quantum algorithms in this context and evaluate their potential for a quantum speedup.

Specifically, we employed the QFIAE method to integrate the one-dimensional tadpole, and bubble Feynman loop integrals, within the LTD formalism, across various kinematical regions, circumventing different threshold singularities using a contour deformation. For the one-dimensional tadpole integral we successfully executed the full quantum pipeline on quantum hardware of Qibo and IBM Quantum obtaining satisfactory results with deviations of around 5% of the analytical value.

Furthermore, we extended QFIAE to multiple dimensions and particularly utilized it to integrate different configurations of the two-dimensional scalar and tensor triangle Feynman loop integrals. The results obtained demonstrate the remarkable accuracy of this quantum algorithm in estimating these integrals.

Moreover, we also used the FQMCI method, which relies on numerical integration to calculate the Fourier coefficients, as opposed to the QFIAE method that employs a QNN for that endeavor. With the FQMCI method, we successfully integrated a particular kinematical configuration of a pentagon integral (three-dimensional function). This integral involves a highly intricate three-dimensional integrand, making its integration a significant challenge. Therefore, the successful application of a quantum algorithm to handle such complex functions is a notable achievement. Nevertheless, designing an efficient Ansatz that enables a QNN to accurately learn the function and allow its integration via QFIAE remains an open challenge for future work.

In a follow-up study, we went one step further to perform the first quantum computation of a total decay rate at second order in perturbative quantum field theory. Leveraging the LTD framework, we have successfully combined loop and tree-level Feynman diagrams with the quantum algorithm QFIAE on a quantum computer. This methodological advancement

is significant from the high-energy physics perspective, as it allows us to integrate a real process with potential for quantum speedup. While we do not claim to have achieved quantum advantage in this work, our results lay the groundwork for future explorations in this direction. From the perspective of quantum computing, our study marks a noteworthy achievement. By solving a relatively complicated regression problem using a QNN on a realistic dataset, we found a good compromise between trainability and expressibility, a common challenge in QNNs. Most of the results presented are in agreement with the expected values within the uncertainty bands, even when the IQAE part of the algorithm is executed on hardware. This demonstrates the potential of quantum computing to address complex, real-world problems, corresponding to real physical processes, and highlights the importance of continuing to push the boundaries of what quantum technology can achieve. Our findings underscore the critical importance of integrating quantum computing methodologies with high-energy physics applications. By tackling practical challenges and pushing the limits of current quantum computing capabilities, we can better understand the potential and limitations of quantum technologies and design better algorithms for future applications.

6 Quantum generative models for Fragmentation Functions

6.1 Introduction

Fragmentation Functions (FFs) are an essential component in particle physics, specifically in the context of Quantum Chromodynamics (QCD), the Quantum Field Theory that describes the interactions between quarks and gluons [300, 301, 302]. FFs describe the hadronization process of partons after a hard-scattering event. Specifically, they quantify the probability that a given parton produces a particular hadron.

In leading-twist collinear QCD [303], collider processes are described using hard scattering partonic cross-sections, initial-state parton distribution functions (PDFs) for the initial hadronic structure, and FFs for the hadronization process [304]. These functions are the core components of the theoretical description of hard interactions and are crucial for their success. Future experiments, like those at the Electron-Ion Collider (EIC), aim to improve FF parametrizations and facilitate combined analyses of PDFs and FFs with higher-precision data [305].

In a typical collider process, partonic cross-sections are calculated perturbatively. However, unlike partonic cross-sections, FFs cannot be calculated directly and must be obtained from experimental data via global analysis across diverse processes [306, 307]. Nevertheless, their energy scale evolution is predicted by QCD.

Several analyses have been proposed to extract FFs from experimental data using statistical tools and heuristic approaches. Initial studies introduced a polynomial-based Ansatz in the parton's momentum fraction [308, 309, 310, 311]. Later approaches used Euler Beta function distributions, along with additional parameters to refine the fit [302, 300, 301]. Despite the progress achieved so far, the determinations of FFs might still be affected by sources of procedural bias, including limited functional form flexibility and uncertainty estimation challenges. Machine Learning techniques, particularly neural networks, have emerged to overcome these limitations [312, 313, 314, 315].

Although the supervised neural network approach has proven effective in determining FFs, and parton densities (PDF) in hadronic collisions [316], some challenges remain. A primary issue lies in the energy-scale evolution. The models are typically trained for a specific range of momentum fraction z and a fix energy scale Q . However, evolving a given FF to different energy scales requires solving the DGLAP evolution equations [317, 318, 319, 320], which might be computationally expensive. Additionally, once the FFs are determined, their true value are only known for specific points, necessitating interpolation to estimate the function at other regions, increasing the complexity of the method.

In the same way that the previous advancements in Machine Learning introduced powerful tools to tackle complex challenges, we are now at the dawn of a new era powered by Quantum Computing (QC) and Quantum Machine Learning (QML). These new techniques, leveraging the properties of quantum mechanics: superposition, entanglement and interference, offer a novel approach to analyzing high-energy-physics (HEP) data [7, 103, 321]. Notable applications include jet clustering [217, 218, 322, 8], elementary particle process integration [5, 226], anomaly detection [223, 224], data classification [323, 324], studying the causal structure [126, 127, 128, 129] and integration of multi-loop Feynman diagrams [2, 3, 9]. In the context of QML [221, 325] and Tensor Networks [326] different methods are being explored for determination and integration of PDFs.

These applications pave the way for analyzing HEP data from collider experiments through the lens of QC. In this context, quantum generative models (QGMs) have shown promising applications in HEP [7, 327, 328, 225, 329, 330].

QGMs usually refer to a class of QML models that leverage Parameterized Quantum Circuits (PQCs) and classical optimization subroutines to generate new data samples resembling a given dataset. QGMs may offer potential advantages over classical generative models in terms of computational power, efficiency, and expressiveness. Several QGMs adapt the classical generative frameworks, such as Quantum Boltzmann Machines (QBM) [331, 332, 333] and Quantum Generative Adversarial Networks (QGANs) [334, 335, 336, 337]. In QBM, a PQC as a Hamiltonian operator is trained such that the resulting Gibbs state matches the target data distribution. Likewise, QGANs are iteratively trained with the generator and discriminator competing against each other to create quantum states and distinguish between generated and real data. Distinct from other QGMs, Quantum Circuit Born Machines (QCBMs) are quantum-inspired generative models that leverage the inherent probabilistic nature of quantum measurement to implicitly treat data represented by output bitstrings as an observable [338, 339, 340].

In contrast to implicit QCBMs based on measurement results, the Quantum Chebyshev Probabilistic Model (QCPM) is a type of generative model that belongs to explicit models that treat data as input variable [341, 342, 343]. More specifically, QCPM allows for separation of the training/learning and sampling/generating stages to make QC differentiable in the latent Chebyshev space and to enable dense sampling in the bit-basis via incorporation of extended registers. Furthermore, QCPM allows us to investigate the association between variables and to scale up systemically for modeling of high-dimensional complex distributions owing to its modular circuit architecture.

In this Chapter, we propose a generalized QCPM framework to study and predict the behavior of FFs [1]. We also show how to study the correlation between the variables z and Q using this model, and its potential impact on QML.

The code to reproduce the results of this chapter is available in [344].

6.2 Fragmentation Functions

Fragmentation functions (FFs) are an essential tool for producing HEP predictions. In particle colliders, such as the Large Hadron Collider (LHC) at CERN, over 1 billion collisions occur every second, generating vast amounts of data that need to be processed and analyzed efficiently [42]. This highlights the importance of fully understanding each component in the process of producing predictions to streamline calculations and manage the data efficiently.

FFs provide the probability that a parton (quark or gluon) will fragment into a particular hadron. This information is key for computing integrals that define observables, such as the cross-section, which are essential for interpreting the results of particle collisions and understanding the underlying physics. To understand the role of FFs in computing exclusive observables, it is convenient to define the differential cross-section for the single-inclusive production of an hadron h in electron-positron annihilation:

$$\frac{d\sigma^h}{dz}(z, Q^2) = \frac{4\pi\alpha^2(Q)}{Q^2} F^h(z, Q^2), \quad (6.1)$$

where F^h is the fragmentation structure function, and $\alpha(Q)$ represents the Quantum Electrodynamics (QED) running coupling. While in the literature F^h is often called fragmentation function, we will denote it as fragmentation structure function to avoid any confusion with the partonic FFs. Now, following the standard collinear factorization [345] the QCD cross-section in a hadronic collider is expressed as a convolution of perturbatively calculable partonic cross-sections and non-perturbative distribution functions.

The structure function is defined as a convolution between coefficient functions and FFs:

$$\begin{aligned} F^h(z, Q^2) = \frac{1}{n_f} \sum_q \hat{e}_q^2 & \left[D_S^h(z, Q^2) \otimes C_{S2,q}(z, \alpha_s(Q)) \right. \\ & + D_g^h(z, Q^2) \otimes C_{S2,g}(z, \alpha_s(Q)) \\ & \left. + D_{NS}^h(z, Q^2) \otimes C_{NS2,q}(z, \alpha_s(Q)) \right], \end{aligned} \quad (6.2)$$

where $\hat{e}_q$ are scale-dependent quark electroweak charge factors, defined in [346] and $\alpha_s(Q)$ is the QCD running coupling. The sum is performed over the n_f active flavours at the scale Q , and $C_{S2,q}, C_{S2,g}, C_{NS2,q}$ are the coefficient functions corresponding respectively to the singlet and nonsinglet combinations of FFs,

$$\begin{aligned} D_S^h(z, Q^2) &= \sum_q D_{q+}^h(z, Q^2), \\ D_{NS}^h(z, Q^2) &= \sum_q \frac{\hat{e}_q^2}{\langle e^2 \rangle} \left[D_{q+}^h(z, Q^2) - D_S^h(z, Q^2) \right], \end{aligned} \quad (6.3)$$

and to the gluon FF, $D_g^h(z, Q^2)$. Note that in Eq. 6.3 the notation $D_{q+}^h \equiv D_q^h + D_{\bar{q}}^h$ and $\langle e^2 \rangle \equiv \frac{1}{n_f} \sum_q \hat{e}_q^2$ has been used. The usual convolution integral with respect to z is denoted by $\otimes$ and reads

$$f(z) \otimes g(z) = \int_z^1 \frac{dy}{y} f(y) g\left(\frac{z}{y}\right). \quad (6.4)$$

Then, to solve Eq. 6.1 it is needed to perform the evolution of the FFs with the energy scale Q , which follows the DGLAP evolution equations [317, 318, 319, 320]. The singlet component $D_S^h(z, Q^2)$ mixes with the gluon FF, and is given by:

$$\begin{aligned} \frac{\partial}{\partial \log(Q)^2} \begin{pmatrix} D_S^h \\ D_g^h \end{pmatrix} (z, Q^2) &= \begin{pmatrix} P_{qq} & 2n_f P_{gq} \\ P_{qg} & P_{gg} \end{pmatrix} (z, \alpha_s) \\ &\otimes \begin{pmatrix} D_S^h \\ D_g^h \end{pmatrix} (z, Q^2), \end{aligned} \quad (6.5)$$

while the non-singlet component $D_{\text{NS}}^h(z, Q^2)$ evolves as:

$$\frac{\partial}{\partial \log(Q^2)} D_{\text{NS}}^h(z, Q^2) = P_+(z, \alpha_s) \otimes D_{\text{NS}}^h(z, Q^2), \quad (6.6)$$

with $P_+ = P_{qq} + P_{q\bar{q}}$ and the splitting functions P_{ji} having a perturbative expansion in the strong coupling α_s :

$$P_{ji}(z, \alpha_s) = \sum_{l=0} a_s^{l+1} P_{ji}^{(l)}(z), \quad (6.7)$$

where $j, i = g, q$, and $a_s = \alpha_s/4\pi$. The time-like splitting functions are known up to $\mathcal{O}(a^3)$ in the modified minimal subtraction, or $\overline{\text{MS}}$, scheme [347, 348, 349].

Now that we understand the contribution that FFs have in computing observables, such as the cross-section in Eq. 6.1, it becomes evident the necessity of an effective way to determine them. One of the most effective methods to date is the NNFF1.0 neural network-based analysis developed by the NNPDF collaboration [314]. Nevertheless, it also presents some limitations, such as determining FFs at fixed energy scale Q , requiring the DGLAP evolution to obtain values at other scales. Furthermore, to explore a wider range of momentum fractions z and energy scales Q , interpolation is required. In the next section, we introduce a quantum generative model that may provide a solution to these challenges, offering a flexible approach for FF scaling and determination.

6.3 Quantum Chebyshev Probabilistic Model

The capability to sample from probability distributions is central in scientific data. In this regard, quantum generative models have the potential to be useful in this context. They combine the potential of classical generative models with one of the main strengths of quantum computing, sampling.

In this section, we introduce a quantum generative model based on Chebyshev polynomials. The main goal of most quantum generative models is to enable sampling in the computational basis, which consists of the orthonormal states $\{|x_j\rangle\}_{j=0}^{2^N-1}$ satisfying the Kronecker delta condition $\langle x_j | x_{j'} \rangle = \delta_{jj'}$. However, training a model directly in this space is often difficult, as it requires a very fine control over the different probabilities. To address this, one can build and train the model in a different orthonormal basis and then map it to the computational basis for sampling. With this in mind, we employ the orthonormal Chebyshev basis and its corresponding transform to build our generative model [341].

First, we need to build the adequate feature map to encode the data into the Chebyshev space. Hence, we build an (unnormalized) quantum state $|\tau(x)\rangle$ whose amplitudes are determined by Chebyshev polynomials of the first kind, defined as $T_k(x) = \cos(k \arccos(x))$, where k denotes the degree. This state takes the form

$$|\tau(x)\rangle = \frac{1}{2^{N/2}} T_0(x) |\emptyset\rangle + \frac{1}{2^{(N-1)/2}} \sum_{k=1}^{2^N-1} T_k(x) |k\rangle, \quad (6.8)$$

where the amplitude of the zero state is weighted by $T_0(x) = 1$. These polynomials satisfy a discrete orthogonality relation at specific sampling points, known as Chebyshev nodes:

$$\sum_{j=0}^{2^N-1} T_k(x_j^{\text{Ch}}) T_\ell(x_j^{\text{Ch}}) = \begin{cases} 0 & \text{if } k \neq \ell, \\ 2^N & \text{if } k = \ell = 0, \\ 2^{N-1} & \text{if } k = \ell \neq 0. \end{cases} \quad (6.9)$$

Here, the Chebyshev nodes are given by $x_j^{\text{Ch}} = \cos\left(\pi(2j+1)/2^{N+1}\right)$, corresponding to the roots of Chebyshev polynomials. The resulting states $\{|\tau(x_j^{\text{Ch}})\rangle\}_{j=0}^{2^N-1}$ form an orthonormal set, with $\langle\tau(x_j^{\text{Ch}})|\tau(x_{j'}^{\text{Ch}})\rangle = \delta_{jj'}$.

It is worth noting that, unlike the standard computational or Fourier bases, which rely on uniform grids, the Chebyshev nodes form a non-uniform mesh within the interval $(-1, 1)$. Outside these node points, the states $|\tau(x)\rangle$ are no longer orthogonal. An analytical expression for their squared overlap can be derived when one of the arguments is fixed to a Chebyshev node:

$$\left|\langle\tau(x')|\tau(x)\rangle\right|^2 = \frac{[T_{2^{N+1}-1}(x')T_{2^N}(x) - T_{2^N}(x')T_{2^{N+1}-1}(x)]^2}{2^{2N}(x' - x)^2}. \quad (6.10)$$

This identity follows from the Christoffel-Darboux formula for Chebyshev polynomials [350]. Since these states are not normalized in general, generating them with a quantum circuit involves using an ancillary qubit and normalizing the state post-preparation. We denote the normalized version as

$$|\tilde{\tau}(x)\rangle = \frac{|\tau(x)\rangle}{\sqrt{\langle\tau(x)|\tau(x)\rangle}}, \quad (6.11)$$

which equals the unnormalized version exactly at the Chebyshev nodes and approaches it as N grows large.

The next step is to design a quantum circuit that prepares the normalized state $|\tilde{\tau}(x)\rangle$, which we will call the feature map. We begin by noting that Chebyshev polynomials can be expressed in terms of cosine functions evaluated on a specific non-uniform grid. Since $\cos(x) = \frac{1}{2}(e^{ix} + e^{-ix})$, the amplitudes can be embedded through a combination of exponential functions, each of which can be implemented using a phase feature map [351]. These components can be combined using the Linear Combination of Unitaries (LCU) framework [352], where the unitary terms are controlled by the state of an ancillary qubit. The desired interference is achieved by post-selecting on the ancilla being measured in the $|0\rangle$ state. Notably, the identity $e^{-ix} = e^{ix} \cdot e^{-i2x}$ allows us to control only one of the exponential components, simplifying the implementation.

This approach enables the construction of equal-weighted combinations of exponentials, scaled appropriately to reproduce the amplitudes associated with Chebyshev polynomials $T_k(x)$. To correctly set the amplitude of the constant term $T_0(x)$, an additional adjustment is required. This is achieved by applying a single iteration of a Grover-like rotation circuit [51], which rotates the state around $|\emptyset\rangle$ by a fixed angle. Since the ancilla measurement commutes with this rotation, we put the measurement to the end of the circuit, completing the preparation of the desired feature map. The quantum circuit of this feature map is shown in Fig. 6.1.

Once we have established how to encode data in the Chebyshev basis we need a transformation between that and the computational basis, and vice versa. Once an orthonormal basis is prepared, there exists a one-to-one correspondence (bijection) and a unitary transformation that maps it to any other orthonormal basis. In this context, we define the Chebyshev transform as

$$\hat{U}_{\text{QChT}} = \sum_{j=0}^{2^N-1} |\tau(x_j^{\text{Ch}})\rangle\langle x_j|, \quad (6.12)$$

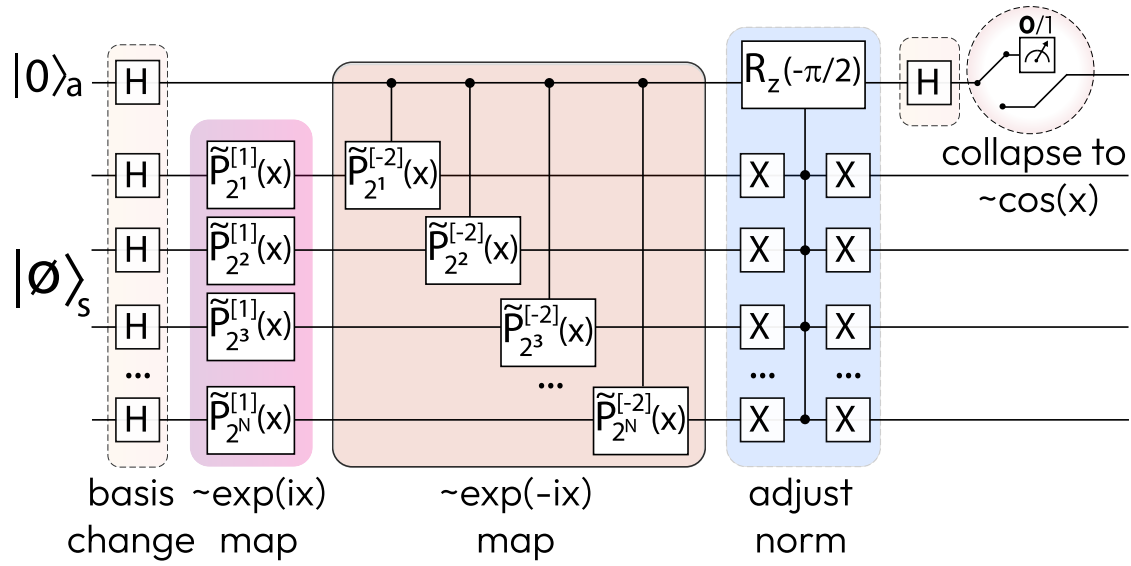


Figure 6.1: Quantum Chebyshev feature map that prepares a Chebyshev state via a sequence of phase feature maps that embed complex exponentials, a controlled rotation to adjust the normalization of the zero-frequency term, and post-selection on the ancilla measured in the $|0\rangle$ state. The scaled single-qubit phase shift gates in the feature map circuit are defined as $\tilde{P}_l^{[s]}(x) = \text{diag}\{1, \exp(is2^N \arccos(x)/l)\}$, where l increases exponentially as 2^j , with j indexing the qubit number, and $s \in \{1, -2\}$. Here, H and X denote the Hadamard and Pauli X gates, respectively. Picture adapted from [341].

which maps computational basis states $|x_j\rangle$ to Chebyshev states $|\tau(x_j^{\text{Ch}})\rangle$. The corresponding circuit to produce this map is presented in Fig. 6.2.

The Chebyshev transform can be viewed as a variant of the cosine transform [353]. Specifically, the amplitude vector of the state $|\tau(x_j^{\text{Ch}})\rangle$ corresponds to the $(j+1)$ th column of the type-II discrete cosine transform matrix, DCT_N^{II} , defined as

$$\text{DCT}_N^{\text{II}} \equiv 2^{-(N-1)/2} \left\{ c_k \cos \left[\frac{k(j+1/2)\pi}{2^N} \right] \right\}_{k,j=0}^{2^N-1}, \quad (6.13)$$

where $c_0 = 1/\sqrt{2}$ and $c_k = 1$ for $k \neq 0$. This matrix is closely related to the Fourier transform, but requires mixing and interference of its elements, motivating the design of the extended QFT-based circuit of Fig. 6.2.

The circuit begins with a Hadamard gate applied to the ancilla qubit (the most significant bit), followed by a ladder of CNOT gates, which collectively prepare a cat-like entangled state. A quantum Fourier transform is then applied to all $N+1$ qubits. This is followed by a sequence of unitary operations that separate and align the real and imaginary parts of the state. In particular, local gates U_1 and R_Z are used to adjust relative phases of the components $|0\rangle_a|\Phi\rangle$ and $|1\rangle_a|\Phi\rangle$, where $|\Phi\rangle$ represents an arbitrary N -qubit intermediate state.

A permutation subcircuit is then applied to reorder the amplitudes appropriately, followed by an additional CNOT ladder. The circuit concludes with a global phase adjustment via U_2 and a set of multi-controlled R_X gates to scale the output amplitudes such that $|0\rangle_a|\Phi\rangle$ and $|1\rangle_a|\Phi\rangle$ are purely real and imaginary, respectively. We note that the ancilla starts and ends in the $|0\rangle$ state, ensuring a "clean" execution. Finally, the complete feature map is obtained by composing the embedding with the transform as $\hat{U}_f(x) = \hat{U}_{\text{QChT}}^\dagger \hat{U}_\tau(x)$.

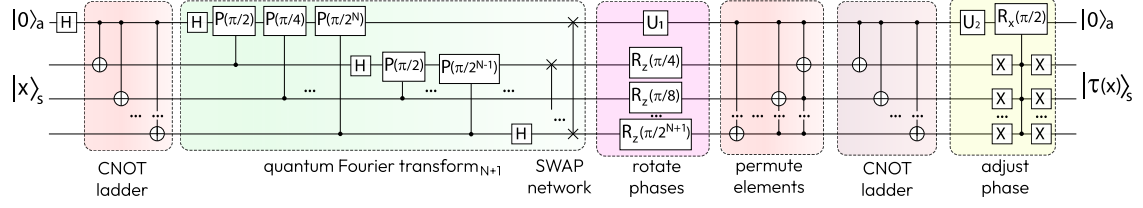


Figure 6.2: Quantum Chebyshev transform ($\hat{U}_{\text{QChT}}$) circuit, which maps computational basis states $\{|x_j\rangle\}_{j=0}^{2^N-1}$ into Chebyshev states $\{|\tau(x_j^{\text{Ch}})\rangle\}_{j=0}^{2^N-1}$. The transform consists of a quantum Fourier transform (QFT) applied to $N + 1$ qubits (an N -qubit system plus one ancillary qubit), followed by a sequence of phase-adjusting and permutation operations. We employ a standard phase shift gate defined as $P(\phi) = \text{diag}\{1, \exp(i\phi)\}$. Additionally, we use local phase rotations acting on the ancilla, given by $U_1 = P(-\pi/2^{N+1})R_Z(-\pi(2^N - 1)/2^{N+1})$ and $U_2 = P(-\pi/2)R_Y(-\pi/2)$, which can be combined into a single gate for implementation. Picture adapted from [341].

After we have defined a protocol to encode data into Chebyshev basis states and a map between this basis and the computational one, we can build a probabilistic model that aims to learn in Chebyshev basis and sample in the computational basis. The workflow of this Quantum Chebyshev Probabilistic Model (QCPM) is depicted in Fig. 6.3.

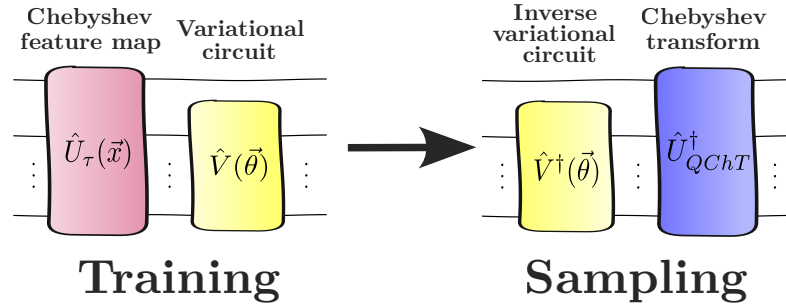


Figure 6.3: Workflow of the QCPM. *Left:* Training circuit encodes x using a Chebyshev feature map, and includes a variational circuit whose parameters will be tuned in order to reproduce the behavior of the desired distribution. *Right:* Sampling circuit performs the inverse operations of the trained variational circuit, and the quantum Chebyshev transform, producing data that follows the distribution employed for training.

One of the most interesting features of this QCPM of Fig. 6.3 is that it allows you to train the data over the target distribution with a limited amount of qubits, which is usually associated to a limited sampling density, and then perform an extended sampling using a QChT with additional qubits. This enables a natural “quantum interpolation” that increases the resolution up to the desired level. This improvement comes at the cost of reducing the maximum probability per point, as the output is distributed over a larger number of datapoints. In Section 6.4 we will see a practical example of how this feature can be leveraged.

6.4 Quantum Chebyshev Probabilistic Models for Fragmentation Functions

6.4.1 Model

We introduce a quantum generative model that provides a flexible approach for FF scaling and determination. The workflow of the QCPM algorithm is shown in Fig. 6.4.

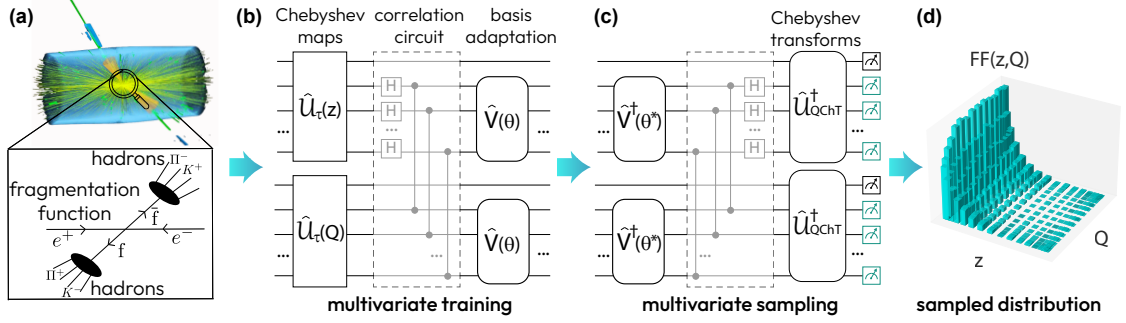


Figure 6.4: Workflow for describing fragmentation functions with quantum probabilistic models based on Chebyshev polynomials. (a) The input data for $D_i^h(z, Q)$ is produced for a grid of z and Q . (b) Quantum probabilistic model is composed of two Chebyshev feature maps for encoding z and Q , a correlation circuit that entangles both registers, and basis adaptation circuits to be trained on $D_i^h(z, Q)$. (c) For sampling we perform the inverse of basis adaptation, the correlation circuit, followed by parallel inverse quantum Chebyshev transforms for mapping the model into the bit basis. (d) Sampling results assembled in a 2-dimensional plot that represents $D_i^h(z, Q)$.

The training circuit is composed of two Chebyshev feature map circuits $\hat{U}_\tau(u)$ and $\hat{U}_\tau(v)$ for encoding two independent variables, u and v in parallel registers, followed by a correlation circuit $\hat{C}$ and two separate variational Ansatzes $\hat{V}_\theta$ and $\hat{V}_\vartheta$, as illustrated in Fig. 6.4(b). The correlation circuit generates Bell-like entangled states to make two otherwise independent latent variables correlated for efficient training. The quantum model $p_Q(u, v) = \alpha p_{\theta, \vartheta}(u, v) + \beta$ is trained to search for optimal angles (θ_{opt} and ϑ_{opt}) and classical weighting parameters (α_{opt} and β_{opt}), so that a mean squared error loss function is minimized. On the other hand, the sampling circuit consists of the inverse versions of the trained variational Ansatzes, followed by two identical sets of extended inverse quantum Chebyshev transform $\hat{U}_{\text{QChT}}^\dagger$ circuits, a procedure accounting for the basis transformation from the Chebyshev to bitbasis spaces, as illustrated in Fig. 6.4(c).

Fig. 6.5 depicts the proposed scheme to make QCPMs more general and applicable to most of probability distributions. Given a probability density function $p(x, y)$ distributed within a bounding box defined by two points, a and b (Fig. 6.5(a), top), we build a quantum model $p_Q(x, y)$ with finite $2N$ qubits to approximately represent $p(x, y)$ and express the quantum model as a two-dimensional Chebyshev expansion, $p_Q(x, y) = \sum_{k,l=0}^{2N-1} c_{k,l} T_k(u) T_l(v)$, where $u = u(x) = \frac{2x-(x_a+x_b)}{x_b-x_a}$ and $v = v(y) = \frac{2y-(y_a+y_b)}{y_b-y_a}$, which is a linear transformation that maps the problem domain $\Omega_P = [x_a, x_b] \times [y_a, y_b]$ with coordinates (x, y) to the Chebyshev domain $\Omega_C = [-1, 1] \times [-1, 1]$ with coordinates (u, v) , as shown in the top panels of Fig. 6.5(a,b). The reason why this shifted transformation is required is that $T_k(x)$ ($T_l(y)$), the Chebyshev polynomials of the first kind, only satisfies a discrete orthogonality condition and forms a complete orthogonal basis over the range $x \in [-1, 1]$ ($y \in [-1, 1]$). In Ω_C , the problem becomes a matter of determining the coefficients $c_{k,l}$, which can be found after a successful training of the parameterized quantum circuits. The model $p_Q(u, v)$ is

trained on a *training grid* consisting of 2D Chebyshev nodes $\{u_i^{\text{Ch}}\}_{i=0}^{2^N-1} \times \{v_j^{\text{Ch}}\}_{j=0}^{2^N-1}$ plus additional half-index points $\{u_{i+1/2}^{\text{Ch}}\}_{i=0}^{2^N-2} \times \{v_{j+1/2}^{\text{Ch}}\}_{j=0}^{2^N-2}$. The *training grid* is schematically illustrated as a black-dotted grid within Ω_C with the total number of training points denoted by $G_T = (2^{N+1} - 1)^2$. Once the model got trained, sampling is carried out through projective measurements in the same domain (Fig. 6.5(b), bottom).

Finally, the sampled $p_Q(u, v)$ is mapped back to the problem domain to obtain the sampled histogram $p_Q(x, y)$ (Fig. 6.5(a), bottom) by $x = x(u) = \frac{(x_b - x_a)u + (x_a + x_b)}{2}$ and $y = y(v) = \frac{(y_b - y_a)v + (y_a + y_b)}{2}$. Specifically, we perform classical post-processing tasks on a batch of measured binary datasets after projective measurements in the computational basis. This involves periodically dropping out those bits with zero probability, concatenating rest of the bits in a sequential way, and then reshaping the resulting bitstring into a 2D array with the size compatible to the Hilbert space of $2(N + S)$ -qubit states, where S is the number of extended registers. Therefore, the sampled $p_Q(u, v)$ is represented by the processed samples spatially assigned to a *sampling grid* composed of 2D Chebyshev nodes $\{u_i^{\text{Ch}}\}_{i=0}^{2^{(N+S)}-1} \times \{v_j^{\text{Ch}}\}_{j=0}^{2^{(N+S)}-1}$, schematically illustrated as a color-dotted grid within Ω_C with the total number of sampling points denoted by $G_S = (2^{N+S})^2$. For the purpose of visualization, we set $N = 2$ and superimpose both the training and sampling grids for different values of S , as shown in Fig. 6.5(c). The ratios of G_S to G_T are 0.33, 1.31 and 5.23, respectively, and they converge to a constant value of 0.25, 1 and 4 when N gets larger. Considering that the overall probability sum for all possible outcomes stays at one, this fact implies that the maximum probability will be decreased by four folds as the number of the extended registers increases.

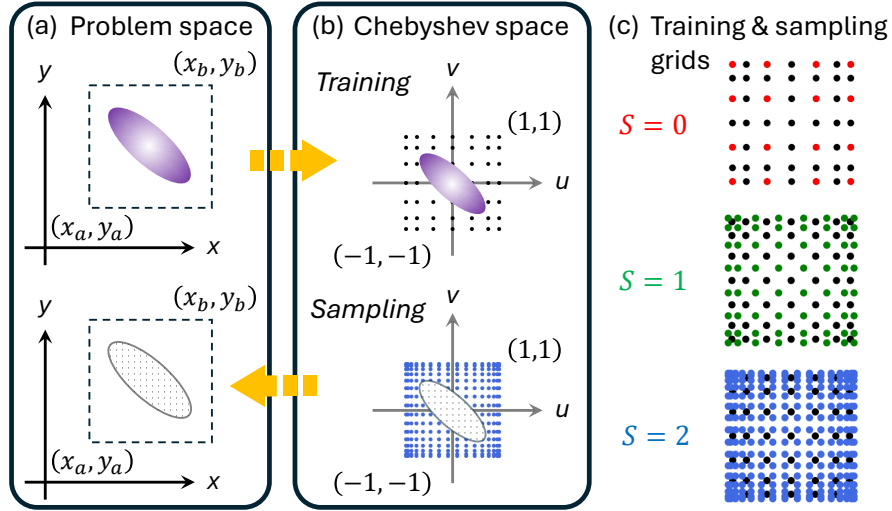


Figure 6.5: Generalized Quantum Chebyshev Probabilistic modeling. Application of shifted Chebyshev polynomials to the problem (a) and Chebyshev (b) spaces. (c) Comparisons of training and sampling grids for different extended registers.

Most importantly, the quantum model is trained on a sparse *training grid* given insufficient training dataset, whereas during sampling stage for $S > 0$, the trained model provides predictions about unseen data at a controllable dense *sampling grid* not spatially overlapped with the *training grid*, i.e., the learned distribution of the training data has enough information to mimic the actual distribution of the data, which enables robust representations of generated images through so-called quantum generalization/interpolation. In the following, we assume that the problem domain of

interest $\Omega_P = [10^{-2}, 1] \times [1, 10000]$ is expressed in terms of the momentum fraction and energy scale (z, Q) , as shown in Fig. 6.4(d).

The training and sampling quantum circuits that form the QCPM employed here are presented in Fig. 6.6.

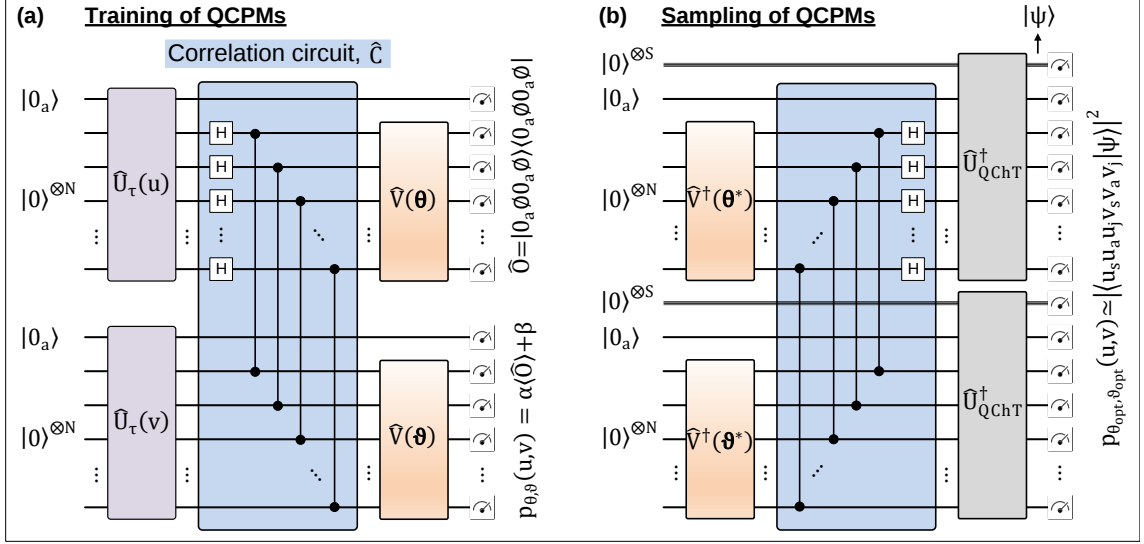


Figure 6.6: (a) Quantum circuit used to train the multivariate distribution in the QCPM latent space, where a correlation circuit $\hat{C}$ is sandwiched between two identical sets of feature map circuits and variational Ansätze. Measured observable is defined as $\hat{O} = |0_a 0_a\rangle\langle 0_a 0_a|$, where $|\rangle \equiv |0\rangle^{\otimes N}$. Here, α and β are trainable scaling and bias parameters. (b) Quantum circuit used to sample the multivariate distribution from the trained model, where the inverse versions of the same parameterized circuits are applied with θ^* and ϑ^* being retrieved after the optimization procedure, followed by the inverse versions of the same correlation and two identical sets of quantum Chebyshev transform circuits associated with extended registers of S qubits ($|0\rangle^{\otimes S}$) in parallel, for fine sampling in the computational basis $|u_s u_a u_j v_s v_a v_j\rangle$. The quantum state prior to measurement is denoted as $|\psi\rangle$.

The training circuit (Fig. 6.6(a)) is composed of two Chebyshev feature map circuits $\hat{U}_\tau(u)$ and $\hat{U}_\tau(v)$ for encoding two independent variables, u and $v \in \Omega_C$, in parallel registers, followed by a correlation circuit $\hat{C}$ and two separate variational Ansätze $\hat{V}(\theta)$ and $\hat{V}(\vartheta)$. The correlation circuit generates Bell-like entangled states to make two otherwise independent latent variables correlated for efficient training of fragmentation functions. The quantum model is defined as:

$$p_Q(u, v) = p_{\theta, \vartheta}(u, v) = \alpha |\langle 0_a 0_a | (\hat{I} \otimes \hat{V}(\theta) \otimes \hat{I} \otimes \hat{V}(\vartheta)) \hat{C} (\hat{U}_\tau(u) \otimes \hat{U}_\tau(v)) | 0_a 0_a \rangle|^2 + \beta, \quad (6.14)$$

and it is trained to minimize a mean squared error loss by optimizing the angles θ^* , ϑ^* and classical weighting parameters α_{opt} and β_{opt} . Since

$$p_{\theta^*, \vartheta^*}(u, v) \simeq |\langle u_s u_a u_j v_s v_a v_j | (\hat{U}_{\text{QChT}}^\dagger \otimes \hat{U}_{\text{QChT}}^\dagger) \hat{C}_{\text{ext}}^\dagger \times (\hat{I}^{\otimes (S+a)} \otimes \hat{V}^\dagger(\theta^*) \otimes \hat{I}^{\otimes (S+a)} \otimes \hat{V}^\dagger(\vartheta^*)) | 0_s 0_a 0_s 0_a \rangle|^2 \quad (6.15)$$

the sampling circuit (Fig. 6.6(b)) consists of the inverse operations of the trained variational Ansätze, followed by an extended inverse correlation circuit $\hat{C}_{\text{ext}}^\dagger$ and two identical sets of extended inverse quantum Chebyshev transform $\hat{U}_{\text{QChT}}^\dagger$ circuits. This procedure allows the

basis transformation from the Chebyshev to computational basis spaces. In this context, we assume that the problem domain of interest is expressed in terms of the momentum fraction and energy scale $(z, Q) \in \Omega_P$.

The training setup for the QCPM is defined as follows. We fix the epoch count of the ADAM [299] optimizer to 10^4 , the number of Ansatz layers to 3, and the number of qubits per variable to 4. Hence, both circuits have the same number of free parameters (16 for each variable) and require the same training time. To optimize performance, we perform a sweep over the learning rate of the ADAM optimizer in the range $[0.1, 1.0]$ and select the model that achieves the highest accuracy in learning the functions $D_i^h(z, Q)$. All quantum simulations are performed using PennyLane [354], and the training process is accelerated with JAX [298]. The structure of the variational Ansatz used for the variational quantum circuit is depicted in Fig. 6.7.

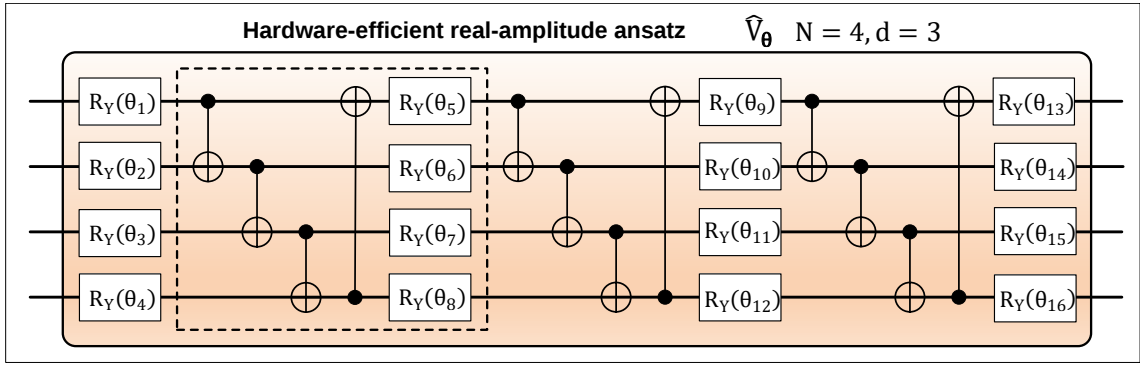


Figure 6.7: Hardware-Efficient Real-Amplitude (HERA) quantum circuit for $N = 4$ qubits with a circuit quantum depth of $d = 3$. The first block ($d = 1$) is framed by a dashed box. The HERA is composed of $N(d + 1)$ tunable single-qubit R_Y gates and Nd entangling (CNOT) gates. The training parameters are $\{\theta_i\}_{i=1}^{N(d+1)}$.

In contrast to GAN architectures, which involve the simultaneous training of two neural networks, the QCPM framework separates the training and sampling stages. Because the probability distribution is encoded in the orthogonal Chebyshev basis states provided by the quantum Chebyshev feature map circuit, the model effectively represents a wide range of functions with sufficient training data and learns patterns with a decent number of variational parameters. The number of variational parameters used for training in this work is 32 ($= 2N(d + 1)$) with $N = 4$ and $d = 3$. The trained model generates new data that fall within the trained distribution. Beyond its generative capabilities, the QCPM efficiently encodes probability distributions into quantum states, making it a valuable tool for a broader range of quantum algorithms and applications.

6.4.2 Results

In this section, we apply the quantum generative Chebyshev model to Fragmentation Functions (FFs), using data, accessible via the LHAPDF6 interface [355, 356]. These FF sets are derived from data on hadron production in electron-positron Single-Inclusive Annihilation (SIA), one of the cleanest processes for studying hadron production, as it does not require simultaneous knowledge of PDFs. The process of obtaining FFs for different hadrons incorporates symmetry assumptions and empirical fragmentation preferences. Isospin symmetry is employed to relate FFs for oppositely charged pions and kaons, with the assumption that certain quarks preferentially fragment into specific hadrons, i.e. they are favoured, unfavoured towards a particular hadron. Charge conjugation symmetry is

also utilized to simplify the analysis by linking FFs for positive and negative hadrons, allowing the functions for opposite charges to be expressed in terms of each other.

We use the NNFF10_PISum_nnlo set for pions ($\pi^\pm = \pi^+ + \pi^-$) and the NNFF10_KASum_nnlo set for kaons ($K^\pm = K^+ + K^-$), both at next-to-next-to-leading order (NNLO) in perturbation theory. Regarding the specific FFs, D_i^h with i being the parton and h the hadron, that will be analyzed, the inclusive SIA data involves a total of five independent combinations of FF:

$$\{D_g^h, D_{b^+}^h, D_{c^+}^h, D_{d^++s^+}^h, D_{u^+}^h\}, \quad (6.16)$$

where h represents the hadron the parton fragments into, $D_{q^+}^h \equiv D_q^h + D_{\bar{q}}^h$, and $D_{d^++s^+}^h \equiv D_{d^+}^h + D_{s^+}^h$.

In Fig. 6.8 we present the results of one particular FF as an example of the validity of the method, $D_g^{K^\pm}$, which corresponds to the sum of FFs of the gluon g fragmenting into the kaon K^+ and its antiparticle K^- . Note that the visualization of the sampling for all the other FFs in Eq. 8.43 for $h = K^\pm, \pi^\pm$ are available in the Appendix B.

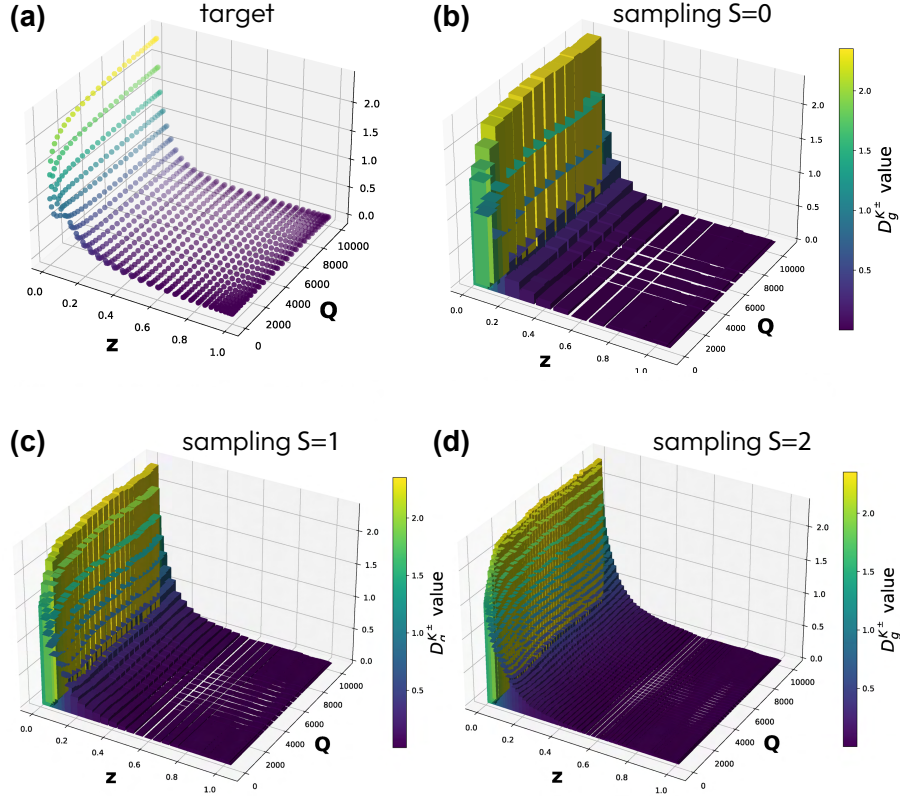


Figure 6.8: Sampling of $D_g^{K^\pm}(z, Q)$, the FF of a gluon fragmenting into kaons, with $S = 0, 1, 2$ additional qubits for each variable to interpolate in untrained regions. (a) Target distribution $D_g^{K^\pm}(z, Q)$. (b) Samples from trained QCPM of $D_g^{K^\pm}(z, Q)$ with the same number of qubits ($S = 0$). (c, d) Samples from $D_g^{K^\pm}(z, Q)$ with extended register ($S = 1, 2$).

The results of Fig. 6.8 show that the model correctly captures the behavior of the two-dimensional function $D_{c^+}^{K^\pm}(z, Q)$ in the region of interest. Fig. 6.8 (a) depicts the target function that the model is trained to learn, represented by the datapoints in the *training grid*. Fig. 6.8 (b) shows the ability of the model to generate samples using the

same quantum registers employed during training. Fig. 6.8 (c) and (d) display the sampling performance when the model utilizes one ($S = 1$) or two ($S = 2$) additional qubits per variable. In the case of $S = 2$ the model generalizes the behavior of the target function and makes predictions of the values of $D_{c^+}^{K^\pm}(z, Q)$ in untrained regions. Since one can increase the number of extra qubits for sampling, this approach shows significant potential as an effective method for achieving natural quantum interpolation with the desired accuracy by adding more qubits.

At this point, it is of particular interest to analyze how correlations between the variables affect the training process of the QCPMs. In this study, we explicitly introduce correlations between the variables z and Q in the training circuit and analyze their impact on the performance of the models. The introduction of correlations between variables z and Q in the training circuit is motivated by their analytical relationship through the DGLAP evolution equations. In this work, we use a heuristic-based approach, where we explicitly introduce a correlation circuit $\hat{C}$, as illustrated in Fig. 6.6, to entangle the registers that load z and Q , which we refer to as $\mathcal{Z}$ and $\mathcal{Q}$. This circuit combines Hadamard gates (H) applied to the first variable with controlled- Z gates (CZ). Our aim is to evaluate the impact of these correlations on QCPM performance and infer correlations.

Specifically, we compare the accuracy of the models with (w/ CC) and without (w/o CC) correlations between the variables and under identical conditions (same number of qubits, trainable parameters and optimizer iterations). To do so, we consider the same setup as in Fig. 6.8. Since we are solving a regression problem, we use the coefficient of determination R^2 as the accuracy to quantify the goodness of the fit.

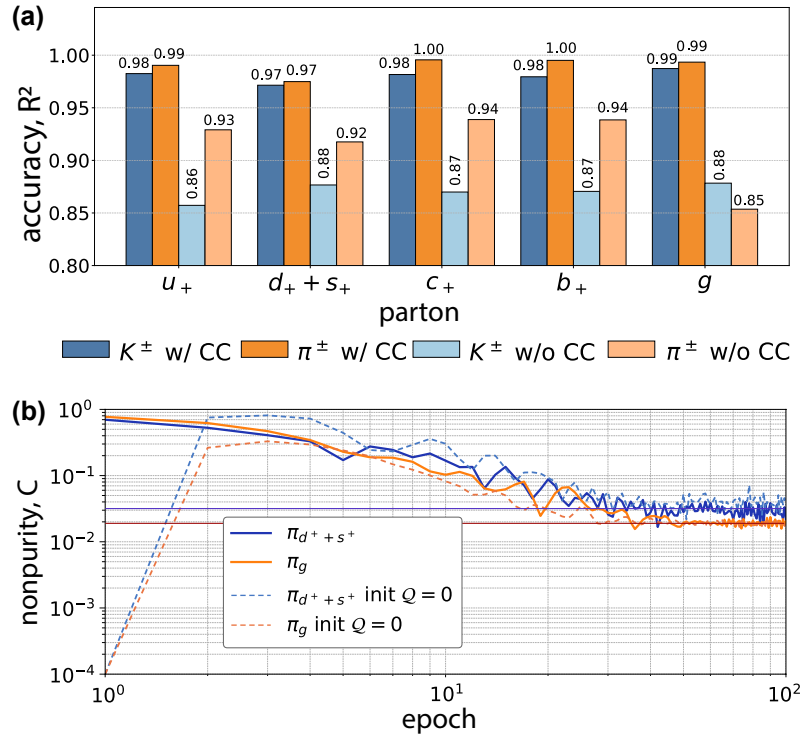


Figure 6.9: (a) Accuracy (R^2) comparison of QCPMs for learning FFs $D_i^h(z, Q)$ with (w/ CC) and without (w/o CC) correlations between z and Q . (b) Nonpurity coefficient $C = |1 - \gamma_{\mathcal{Z}}|$ of system $\mathcal{Z}$ for 100 training epochs with the fixed correlation circuit (solid curves, log-log curves). Dashed curves represent C when the system starts in a product state without entanglement between registers (no correlations). Solid curves highlight the values of nonpurity after training.

The results presented in Fig. 6.9(a) demonstrate that incorporating the correlation circuit $\hat{C}$ in the QCPM, leads to an improvement in performance across all the FFs studied. While models without correlations could reach similar accuracy with more optimizer iterations or additional trainable parameters, under the same conditions, the models with correlations consistently outperform those without. These findings highlight an important insight offered by the developed quantum probabilistic models: it is possible to infer, indirectly, the degree of correlation between physical variables by analyzing the performance of the quantum model and the role of entanglement between $\mathcal{Z}$ and $\mathcal{Q}$. While correlations are typically estimated from sample statistics and require large datasets, we propose assessing them through the entanglement structure within QCPMs. This can be done using various measures, such as entanglement entropy, purity, or mutual information, this is further explored in Appendix C. In Fig. 6.9(b) we focus on purity [357], as it can be efficiently evaluated via the SWAP test [358].

In particular, to quantify the correlations we define the nonpurity as $C = |1 - \gamma_{\mathcal{Z}}|$, where $\gamma_{\mathcal{Z}}(\rho) \equiv \text{Tr} \mathcal{Q}(\rho \mathcal{Z}^2)$ denotes the purity of subsystem $\mathcal{Z}$ after tracing out $\mathcal{Q}$. We study the evolution of C during the sampling stage of the QCPM, which includes a fixed correlation circuit, and present the behavior of C over 100 training epochs in Fig. 6.9(b) (solid lines for two representative processes). Initially, the nonpurity is relatively high, around $C_0 \approx 0.7$, but gradually decreases to a small yet non-zero value, $C_{100} \approx 2 \times 10^{-2}$ (see red and blue curves in Fig. 6.9(b)). Note that similar final values of C are observed even when starting from uncorrelated register initializations (dashed curves in Fig. 6.9(b)). Although modest, this residual nonpurity indicates the presence of non-negligible $\mathcal{Z}$ - $\mathcal{Q}$ correlations necessary to achieve the model's final accuracy, as shown in Fig. 6.9(a). Overall, the analysis of QCPM trained on multidimensional datasets leads to quantifiable measures of cross-correlations to high accuracy.

6.5 Conclusions

In this Chapter we have studied how Quantum Chebyshev Probabilistic Models (QCPMs) learns multivariate HEP distributions, inferring relevant properties, and performing generative modeling by sampling of quantum circuits. In particular, we applied QCPMs to describe fragmentation functions (FFs) of charged pions and kaons [1], which are essential for understanding hadron production in high-energy collisions. This approach aligns well with previous works, which have established that FFs present a polynomial dependence on the momentum fraction z . Hence, the Chebyshev basis proves to be particularly suitable for capturing this behavior.

Moreover, the Quantum Chebyshev Probabilistic Model presents some interesting features. First, we have demonstrated that the generalized QCPM enables the extension of the Chebyshev space for modeling the largest class of probability distributions. Moreover, once the model is trained, it allows for a $2^{2(S-1)}$ -fold increase in sampling grids by incorporating extended registers, facilitating a natural “quantum interpolation” in regions where the model lacks direct training data. This stands in contrast to previous classical models, which often require the computationally expensive DGLAP evolution for interpolation in untrained regions. On the other hand, the Chebyshev quantum model's structure enables us to study the correlation between the momentum fraction z and the energy scale Q . We have performed an analysis to examine how correlations between quantum registers encoding z and Q affect the model's training performance. We found that introducing entanglement between these registers leads to improved model accuracy compared to non-entangled models under identical conditions. This finding extends what was found

in previous studies suggesting that entanglement (“quantumness”) can enhance model training. These results underscore the potential of QML in analyzing quantum physical processes, such as HEP ones, and motivate the need for further research into how quantum computing techniques can be leveraged to study quantum systems.

7 Outlook and final remarks

The Standard Model (SM) of particle physics is one of the most successful theories to date, with the discovery of the Higgs boson in 2012, 13 years ago, at the CERN's Large Hadron Collider (LHC) as a major breakthrough in high-energy physics (HEP). Further sophisticated HEP experiments are being planned to dig deeper into the origin of the universe and the fundamental content of matter. In this regard, these more complex experiments, demand advanced computational tools, both to analyze the vast amounts of data produced in high-energy collisions and to generate more accurate and precise theoretical predictions.

In this context, quantum computing has emerged over the past few decades as a promising technology with the potential to surpass the capabilities of classical devices in data analysis. Nevertheless, significant progress is still required. Both in hardware, where better quantum processors must be built, and in software, where more efficient and resilient algorithms are needed to achieve quantum advantage. One particularly interesting application lies in HEP. Inspired by Richard Feynman's pioneering idea of using quantum computers to simulate quantum systems, it seems natural to apply quantum computing to analyze HEP data, which is essentially governed by Quantum Field Theory. In particular in this Thesis we have explored different avenues of research where quantum algorithms can be used for HEP usecases.

In Chapter 4 we introduced two quantum circuits to speed up clustering algorithms: a quantum subroutine for computing Minkowski distances and a quantum routine for tracking the maximum value in unsorted data. Employing these quantum subroutines, we designed quantum versions of three widely used clustering algorithms: K-means, Affinity Propagation, and k_T -jet clustering. For K-means, we implemented a quantum version that follows the classical structure but replaces Minkowski distance computations and minimum distance calculations with quantum circuits. This approach maintains clustering accuracy while offering exponential speedups in vector dimensionality d and number of clusters K when executed on quantum hardware with qRAM. The quantum Affinity Propagation algorithm follows a similar design, using quantum Minkowski computations to evaluate similarities, again achieving exponential speedup obtaining the same performance. We also introduced a quantum approach to k_T -jet clustering, where the minimum distance is computed with our quantum algorithm, that improves the runtime from $\mathcal{O}(N^3)$ classically to $\mathcal{O}(N^2 \log N)$ quantumly, with further potential to reach $\mathcal{O}(N \log N)$ by incorporating geometric nearest-neighbor optimizations, analogous to those used in the classical FastJet package. Across all three cases, our quantum simulations demonstrated clustering efficiencies comparable to classical methods, while offering theoretical advantages in scalability and runtime when executed on quantum computers.

Then, in Chapter 5 we introduced a novel Quantum Monte Carlo integrator, dubbed Quantum Fourier Iterative Amplitude Estimation (QFIAE), which leverages Quantum

Machine Learning (QML) and Quantum Amplitude Estimation (QAE) to overcome limitations of earlier quantum integration methods like Fourier Quantum Monte Carlo Integration (FQMCI). The key feature of QFIAE is that it uses a Quantum Neural Network (QNN) to learn the target function that we want to integrate and extracts the Fourier series that this circuit represents. Then, the Fourier trigonometric components are easily integrated using QAE surpassing the problem of encoding arbitrarily complicated target functions. Unlike FQMCI, it provides with a full quantum pipeline and eliminates the need for classical numerical integration when estimating Fourier coefficients. This makes QFIAE, to the best of our knowledge, the first end-to-end quantum integrator capable of preserving the full quadratic speedup from amplitude amplification. We began by validating the method on a one-dimensional scattering process, $e^+e^- \rightarrow q\bar{q}$, where QFIAE achieved a relative error of about 1% with circuits that remain within the depth and width limits of current NISQ devices. Encouraged by these results, we applied the algorithm to more complex tasks: evaluating loop Feynman integrals using the Loop-Tree Duality (LTD). In one dimension, we successfully ran the full algorithm in the quantum computers of Qibo and IBM Quantum for tadpole integrals, reaching errors around 5%. We then extended QFIAE to higher dimensions, applying it to scalar and tensor triangle loop integrals in two dimensions, and demonstrating accurate and reliable estimates. Our most advanced application combined tree and loop-level Feynman diagrams to compute a physical observable for the first time on a quantum compute. The decay rate at next-to leading order (NLO) in perturbative Quantum Field Theory. This milestone, achieved within the LTD framework using QFIAE, demonstrates the feasibility of applying quantum algorithms to realistic HEP processes. While quantum advantage is not claimed, our results represent a significant step forward, as the results are in agreement with analytical values. From a quantum computing point of view, we found that the QNNs' Ansatz we chose can handle realistic regression problems effectively, with a good balance between trainability and expressibility.

Finally, in Chapter 6 we employed Quantum Chebyshev Probabilistic Models (QCPMs) as a new approach for learning and generating multivariate distributions in HEP. Specifically, we applied these models to fragmentation functions (FFs), which describe how quarks and gluons transform into hadrons during high-energy collisions. One of the strengths of this method lies in its flexibility, since QCPMs can be pretrained to match known marginal distributions and then fine-tuned to capture inter-variable correlations more accurately. Our results showed that introducing entanglement between the quantum registers representing the variables z and Q led to a noticeable improvement in training performance, supporting the growing evidence that quantum correlations can help in the learning process in quantum models. On the generative side, QCPMs demonstrated strong generalization capabilities, accurately interpolating between training points. Thanks to the Chebyshev basis, our models could sample over finely spaced grids, and the use of additional S qubits in the quantum registers exponentially increased the resolution of the sampling space by a factor of $2^{2(S-1)}$. These results highlight the potential of QCPMs as powerful QML tools for modeling complex, structured distributions in particle physics. By enabling both precise inference and high-resolution generative modeling, this approach opens new directions for using quantum computing to gain deeper insight into fundamental processes in high-energy collisions.

Overall, the work presented in this thesis paves the way for developing more specialized and powerful quantum methods that make full use of core quantum features like superposition and entanglement to improve computational methods to analyze HEP data and Quantum Field Theories. A key direction for future research is to scale these techniques to larger,

fault-tolerant quantum systems, where real quantum advantage over classical machines can be fulfilled.

8 Resumen de la tesis

A continuación, se presenta una descripción resumida del contenido de esta tesis. En ella se incluye una introducción a los temas principales abordados, un resumen de los objetivos planteados, una explicación de la metodología empleada, una síntesis de los principales resultados obtenidos y, finalmente, se exponen las conclusiones generales del trabajo.

Introducción

El Modelo Estándar es la teoría más exitosa en el ámbito de la física de partículas. Una de sus mayores fortalezas es la capacidad que tiene de producir una descripción altamente precisa de tres de las cuatro interacciones fundamentales de la naturaleza: la fuerza electromagnética, la interacción nuclear débil y la interacción nuclear fuerte. En el Capítulo 2 se presenta una descripción detallada de sus fundamentos y de sus características más relevantes.

No obstante, a pesar de sus numerosos éxitos, existe un consenso en la comunidad científica de que el Modelo Estándar no representa una teoría completa de la naturaleza a nivel fundamental. Estas limitaciones continúan siendo un campo activo de investigación y no se ha encontrado una solución hasta la fecha. La falta de contradicciones internas o refutaciones directas deja sin una guía clara para extender el Modelo Estándar, lo que obliga a explorar nuevas posibilidades sin una dirección definida. En esta línea, se plantean nuevos experimentos en colisionadores de alta energía como el Large Hadron Collider (LHC) para llegar a regímenes de energía no explorados que puedan arrojar algo de luz sobre cuales son las piezas que faltan en el rompecabezas que conforma el Modelo Estándar de la física de partículas.

Paralelamente, se está produciendo lo que ha sido bautizado como la “segunda revolución cuántica”. Esta hace referencia a la capacidad de controlar sistemas cuánticos de forma coherente aprovechando sus propiedades inherentes para almanecinar, manipular y leer información. Este campo recibe el nombre de información cuántica. Dentro del mismo, una de las aplicaciones más inmediatas es la capacidad de realizar cálculos y operaciones de interés, lo cual recibe el nombre de computación cuántica. La computación cuántica y los ordenadores cuánticos, que son los dispositivos capaces de realizar estas operaciones, han emergido en los últimos años como una tecnología incipiente con mucho potencial para revolucionar la forma en que resolvemos ciertos problemas y analizamos los datos. Sin embargo, la computación cuántica es un campo de investigación que, con apenas un par de décadas de historia, aún se encuentra en sus inicios. Hacen falta muchos avances tanto a nivel de desarrollo de algoritmos más eficientes y con capacidad de aprovechar las diferentes propiedades cuánticas, como a nivel de desarrollo de dispositivos capaces de ejecutarlos. Actualmente se dice que nos encontramos en lo que se llama la era NISQ (Noise Intermediate Scale Quantum), en la que existen ordenadores cuánticos de un número

limitado de bits cuánticos (qubits) que además son propensos a errores. En este paradigma es difícil pensar que podamos obtener una ventaja cuántica en el futuro inmediato. No obstante, el campo está evolucionando a una velocidad vertiginosa y no es para nada descabellado pensar en obtener resultados útiles de aplicación industrial y científica en las próximas décadas.

Con todo esto en mente e inspirados por la célebre frase de Richard Feynman: “La naturaleza no es clásica, maldita sea, y si quieres hacer una simulación de la naturaleza, más te vale que sea cuántica”, resulta natural considerar la computación cuántica como una herramienta prometedora para analizar datos de física de altas energías, ya que estos están descritos por la Teoría Cuántica de Campos. En esta tesis, exploramos distintas líneas de investigación donde los algoritmos cuánticos pueden aplicarse a problemas relevantes en la física de partículas.

Objetivos

El objetivo principal de esta tesis es identificar problemas en física de partículas con un coste computacional significativo y explorar su resolución mediante algoritmos cuánticos que puedan ofrecer una ventaja, ya sea teórica o heurística. Asimismo, se aborda el diseño y la aplicación de dichos algoritmos a estos problemas concretos.

En particular, podemos diferenciar los algoritmos estudiados en dos tipos. Por un lado se estudian algoritmos cuánticos que ofrecen una ventaja teórica, los cuales garantizan una mejora al ser ejecutados en ordenadores cuánticos suficiente grandes. Por otro lado, también se diseñan algoritmos cuánticos variacionales, los cuales funcionan de forma parecida a métodos ampliamente empleados en el aprendizaje automático (Machine Learning), como las redes neuronales, y por este motivo se asocian habitualmente a lo que se conoce como el aprendizaje automático cuántico (Quantum Machine Learning). Estos algoritmos variacionales, al igual que sus análogos clásicos, no garantizan una ventaja en el rendimiento al ejecutarse en ordenadores cuánticos de tamaño suficientemente grande. En su lugar, ofrecen soluciones alternativas de carácter “heurístico”, que, al escalar el problema, suelen mostrar ventajas computacionales significativas frente a otros métodos. Aunque la ausencia de una prueba matemática que asegure una mejora en la escalabilidad podría parecer una debilidad, lo cierto es que esta situación también se da en el aprendizaje automático clásico, donde, a pesar de la falta de garantías teóricas, las soluciones que producen han demostrado funcionar en una amplia gama de problemas computacionales.

Respecto a los casos de uso que se estudian en esta tesis, nos centraremos en los siguientes tres. Clusterización de jets de partículas, en el Capítulo 4, integración de diagramas de Feynman y tasas de desintegración a next-to-leading order (NLO), en el Capítulo 5, y generación de funciones de fragmentación, en el Capítulo 6.

La clusterización de jets es un proceso que consiste en agrupar partículas que comparten ciertas propiedades para analizarlas como un solo objeto, conocido como jet. La mayoría de algoritmos empleados para resolver este problema se basan en calcular las distancias entre las partículas y agrupar aquellas que se encuentran más próximas. En este contexto, el objetivo es desarrollar y aplicar dos algoritmos cuánticos: uno para calcular eficientemente las distancias entre partículas, y otro que identifique el mínimo de esas distancias para determinar qué partículas deben formar parte del mismo jet.

En cuanto al Capítulo 5, el objetivo es el diseño de algoritmo cuántico de integración tipo Monte Carlo para poderlo aplicar a integrales de Feynmann de loop y al cálculo de observables físicos como las tasas de desintegración. Este algoritmo cuántico combina una

red neuronal cuántica (algoritmo variacional) con una variante del algoritmo de Grover (algoritmo de ventaja teórica).

Finalmente, el objetivo del Capítulo 6 es desarrollar un algoritmo variacional capaz de aprender las funciones de fragmentación asociadas a distintos partones al fragmentarse en hadrones, y de reproducir estos resultados con una resolución superior a la utilizada durante el entrenamiento. Esto permitiría realizar una interpolación “cuántica” de los valores que toman dichas funciones en las regiones de interés.

Metodología

Para obtener los resultados presentados en esta tesis, se ha empleado una metodología precisa y específica que consta de algoritmos deterministas, de aprendizaje automático y de algoritmos cuánticos. Podemos dividir los métodos empleados por secciones.

Algoritmos para jet clustering

Los algoritmos para clusterización de jets empleados en el capítulo 4 de esta Tesis son modificaciones de: K-means, Affinity Propagation y k_T -jet. En particular las modificaciones de estos algoritmos consiste en sustituir las rutinas clásicas que calculan distancias y mínimos por algoritmos cuánticos integrados en la “pipeline” de cada método.

En primer lugar, en esta tesis presentamos un algoritmo cuántico que permite el cálculo de una distancia de Minkowski entre dos puntos. La motivación detrás de usar este tipo de distancia es que en física de altas energías (HEP), los vectores suelen definirse en un espacio-tiempo de cuatro dimensiones con métrica de Minkowski. Estos vectores se expresan como $x_i = (x_{i,0}, \mathbf{x}_i)$, donde $x_{i,0}$ es la componente temporal y $\mathbf{x}_i$ representa las tres componentes espaciales. En general, supondremos d dimensiones espacio-temporales, con $d - 1$ indicando el número de componentes espaciales. El análogo de la distancia Euclídea clásica en el espacio de Minkowski es la suma invariante al cuadrado $s_{ij}^{(C)}$, también conocida como masa invariante al cuadrado cuando se trabaja con cuádrimomentos:

$$s_{ij}^{(C)} = (x_{0,i} + x_{0,j})^2 - |\mathbf{x}_i + \mathbf{x}_j|^2 . \quad (8.1)$$

Esta cantidad invariante de Lorentz se utiliza como métrica de similitud entre momentos de partículas. Además, corresponde a la distancia empleada en ciertos algoritmos tradicionales de clusterización de jets en colisionadores e^+e^- [165, 166, 167]. Calcular esta distancia tipo Minkowski mediante un algoritmo cuántico requiere dos aplicaciones del circuito cuántico *SwapTest* (detallado en el Apéndice A), una para las componentes espaciales y otra para las temporales. La distancia espacial se obtiene a partir de los estados:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}} (|0, x_i\rangle + |1, x_j\rangle) , \quad |\psi_2\rangle = \frac{1}{\sqrt{Z_{ij}}} (|\mathbf{x}_i||0\rangle - |\mathbf{x}_j||1\rangle) , \quad (8.2)$$

donde $Z_{ij} = |\mathbf{x}_i|^2 + |\mathbf{x}_j|^2$. Mientras que la distancia temporal se obtiene a partir de:

$$|\varphi_1\rangle = H|0\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) , \quad |\varphi_2\rangle = \frac{1}{\sqrt{Z_0}} (x_{0,i}|0\rangle + x_{0,j}|1\rangle) , \quad (8.3)$$

donde $Z_0 = x_{0,i}^2 + x_{0,j}^2$. Aplicando el *SwapTest* a estos estados se obtiene:

$$\begin{aligned} P(|0\rangle|_{\text{time}}) &= \frac{1}{2} + \frac{1}{2} |\langle\varphi_1|\varphi_2\rangle|^2 , \\ P(|0\rangle|_{\text{spatial}}) &= \frac{1}{2} + \frac{1}{2} |\langle\psi_1|\psi_2\rangle|^2 , \end{aligned} \quad (8.4)$$

donde los overlap están dados por

$$|\langle \varphi_1 | \varphi_2 \rangle|^2 = \frac{1}{2Z_0} (x_{0,i} + x_{0,j})^2. \quad |\langle \psi_1 | \psi_2 \rangle|^2 = \frac{1}{2Z_{ij}} |\mathbf{x}_i - \mathbf{x}_j|^2. \quad (8.5)$$

Por lo tanto la versión cuántica de la suma invariante al cuadrado resulta:

$$s_{ij}^{(Q)} = 2(Z_0(2P_{\Psi_3}(|0\rangle)_{\text{time}}) - 1) - Z_{ij}(2P_{\Psi_3}(|0\rangle)_{\text{spatial}} - 1). \quad (8.6)$$

Para los $\log_2(d-1)$ primeros qubits, se aplica el *SwapTest* a las componentes espaciales, asumiendo que los estados ψ_1 y ψ_2 se han cargado desde una qRAM en un tiempo de $\mathcal{O}(\log(d-1))$, ya que ψ_1 se codifica en $\log_2(d-1)$ qubits. Para el último qubit, el *SwapTest* se usa para las componentes temporales, y en este caso el coste es $\mathcal{O}(1)$, ya que solo se usan estados de una dimensión.

En segundo lugar, en esta tesis presentamos un algoritmo cuántico sencillo para identificar el valor máximo en una lista no ordenada $L[0, \dots, N-1]$ de N elementos. El procedimiento consta de dos pasos:

1. Los datos se codifican en un estado de $\log(N)$ qubits mediante codificación por amplitudes:

$$|\Psi\rangle = \frac{1}{\sqrt{L_{\text{sum}}}} \sum_{j=0}^{N-1} L[j] |j\rangle, \quad (8.7)$$

donde $L_{\text{sum}} = \sum_{j=0}^{N-1} L[j]^2$ es una constante de normalización

2. El estado resultante se mide múltiples veces, y el índice observado con mayor frecuencia se asocia con el valor máximo.

El cuello de botella del procedimiento radica en la codificación de los datos en el estado cuántico. Asumiendo que los datos están almacenados en una qRAM, como sería el caso en un ordenador cuántico universal real, la codificación requiere $\mathcal{O}(\log(N))$ pasos [169, 170, 171, 172, 173, 174, 175, 176]. Dado que el mejor algoritmo clásico requiere $\mathcal{O}(N)$ pasos, la mejora cuántica es exponencial bajo estas suposiciones.

Este método es probabilístico y puede fallar en identificar correctamente el valor máximo absoluto cuando la desviación estándar de los datos es baja o los extremos son similares. Sin embargo, como se muestra en la Sección 4.4, estas limitaciones no afectan significativamente la aplicación en clusterización de jets. Más allá de HEP, este algoritmo puede ser útil en áreas como la Teoría de Valores Extremos (EVT) [178, 179], relevante en predicción de eventos raros a partir de grandes volúmenes de datos. Esto incluye aplicaciones en finanzas, meteorología, ingeniería, geología, tráfico, entre otras [180, 181, 182].

Algoritmos para integración de funciones multidimensionales

En el Capítulo 5 se describe el método de integración cuántica QFIAE introducido en [5], y que se utiliza para calcular integrales relevantes en física de partículas. Este algoritmo cuántico combina de forma coherente el poder de los circuitos cuánticos parametrizados (PQC), el análisis de Fourier y la estimación de amplitud cuántica para aproximar integrales con alta precisión. En particular, se aplica a integrales Monte Carlo de la forma:

$$I = \int_{x_{\min}}^{x_{\max}} p(x) f(x) dx. \quad (8.8)$$

La estimación de esta integral se realiza evaluando el valor esperado de una función $0 \leq f(x) \leq 1$ sobre entradas de n bits $x \in \{0, 1\}^n$, con probabilidad $p(x)$:

$$\mathbb{E}[f(x)] = \sum_{x=0}^{2^n-1} p(x)f(x). \quad (8.9)$$

El método ofrece una aceleración cuadrática en el número de muestras necesarias, gracias al uso de la estimación de amplitud cuántica basada en el algoritmo de Grover [168]. Para ello, se definen los operadores:

$$\mathcal{P}|0\rangle_n = \sum_{x=0}^{2^n-1} \sqrt{p(x)}|x\rangle_n, \quad (8.10)$$

$$\mathcal{R}|x\rangle_n|0\rangle = |x\rangle_n \left(\sqrt{f(x)}|1\rangle + \sqrt{1-f(x)}|0\rangle \right), \quad (8.11)$$

que permiten construir el operador $\mathcal{A}$, el cual actúa como:

$$\begin{aligned} |\psi\rangle &= \mathcal{A}|0\rangle_{n+1} = \mathcal{R}(\mathcal{P} \otimes \mathbb{I})|0\rangle_{n+1} \\ &= \sum_{x=0}^{2^n-1} \sqrt{p(x)}|x\rangle_n \left(\sqrt{f(x)}|1\rangle + \sqrt{1-f(x)}|0\rangle \right). \end{aligned} \quad (8.12)$$

Definimos:

$$a = \sum_{x=0}^{2^n-1} p(x)f(x) = \mathbb{E}[f(x)], \quad (8.13)$$

$$|\tilde{\psi}_1\rangle = \frac{1}{\sqrt{a}} \sum_{x=0}^{2^n-1} \sqrt{p(x)}\sqrt{f(x)}|x\rangle_n, \quad (8.14)$$

$$|\tilde{\psi}_0\rangle = \frac{1}{\sqrt{1-a}} \sum_{x=0}^{2^n-1} \sqrt{p(x)}\sqrt{1-f(x)}|x\rangle_n, \quad (8.15)$$

y el estado:

$$|\psi\rangle = \sqrt{a}|\tilde{\psi}_1\rangle|1\rangle + \sqrt{1-a}|\tilde{\psi}_0\rangle|0\rangle. \quad (8.16)$$

Ahora se define el operador de Grover:

$$\mathcal{Q} = U_\psi U_{\psi_0}, \quad (8.17)$$

donde:

$$U_{\psi_0} = \mathbb{I}_{n+1} - 2\mathbb{I}_n|0\rangle\langle 0|, \quad U_\psi = \mathbb{I}_{n+1} - 2|\psi\rangle\langle\psi|. \quad (8.18)$$

Usando $a = \sin^2 \theta_a$, se aplica el algoritmo IQAE. Este consta de tres pasos: determinar el número óptimo de aplicaciones de $\mathcal{Q}$, aplicar $\mathcal{A}$ y $\mathcal{Q}^k$, y finalmente extraer el valor de a :

$$\mathcal{Q}^k \mathcal{A}|0\rangle_{n+1} = \sin((2k+1)\theta_a)|\tilde{\psi}_1\rangle|1\rangle + \cos((2k+1)\theta_a)|\tilde{\psi}_0\rangle|0\rangle. \quad (8.19)$$

Una estrategia eficiente para codificar funciones consiste en descomponer $f(x)$ como serie de Fourier:

$$f(x) = c + \sum_{i=1}^{\infty} a_n \cos(n\omega x) + b_n \sin(n\omega x). \quad (8.20)$$

Esto permite usar QAE para estimar los componentes trigonométricos y combinarlos.

Para evitar el costo de calcular los coeficientes de Fourier por métodos clásicos, en [5] se propone usar una red neuronal cuántica (QNN) para obtener la serie de Fourier. Se entrena un circuito siguiendo la arquitectura de “data-reuploading” [85], que alterna bloques de codificación $\mathcal{S}(\vec{x})$ y bloques entrenables $\mathcal{A}(\vec{\theta}_l)$:

$$U_0 \equiv \mathcal{A}(\vec{\theta}_0) , \quad U_l \equiv \mathcal{A}(\vec{\theta}_l) \mathcal{S}(\vec{x}) . \quad (8.21)$$

El circuito consta de dos componentes principales: un operador inicial, U_0 , que prepara un estado en superposición, y un operador U_l , aplicado de forma repetida en capas con distintos parámetros entrenables $\vec{\theta}_l$. Este operador se encarga de mapear los datos de entrada y se optimiza según una métrica definida. La función de coste es el error cuadrático entre el valor esperado del circuito y el valor real de la función. Como muestra [271], este valor esperado se puede expresar como:

$$\langle M \rangle(\vec{x}, \vec{\theta}) = \sum_{\vec{w} \in \Omega} c_{\vec{w}} e^{i\vec{x}\vec{w}} , \quad (8.22)$$

donde los elementos de la serie de Fourier se obtienen directamente del circuito cuántico.

En resumen, el método QFIAE consiste en descomponer la función objetivo en su serie de Fourier mediante el uso de una QNN. Posteriormente, cada componente trigonométrico se integra utilizando IQAE. El funcionamiento de este algoritmo se ilustra en la Fig. 8.1, que ofrece una representación visual de su operativa.

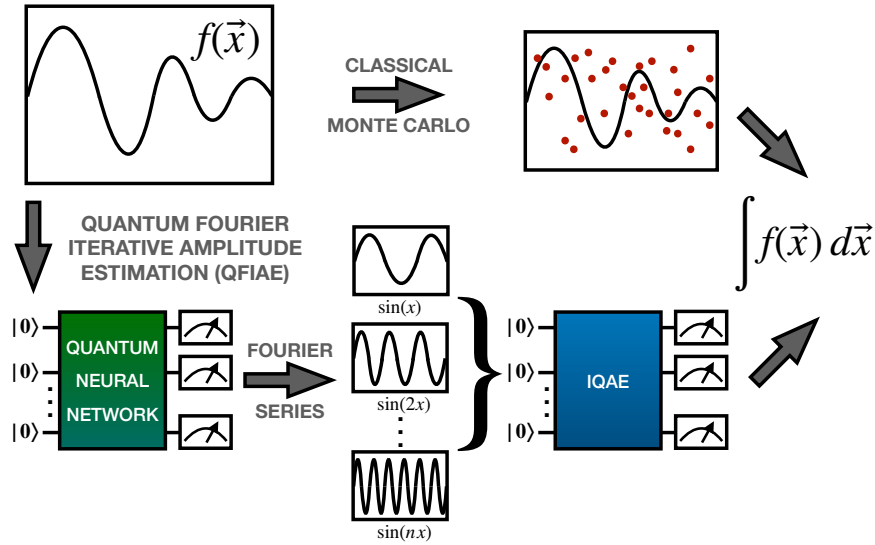


Figure 8.1: Comparación entre los flujos de trabajo de la integración Monte Carlo clásica y el algoritmo cuántico QFIAE. En QFIAE, se introducen como entrada la función objetivo, la distribución de probabilidad y el intervalo de integración. A continuación, una QNN ajusta la función y permite extraer su serie de Fourier. Posteriormente, se utilizan IQAE para calcular los integrales de cada componente trigonométrico en el intervalo especificado. Finalmente, se combinan estos resultados ponderados por sus coeficientes para obtener el valor final de la integral.

Algoritmos para generación de funciones de fragmentación

En el Capítulo 6 introducimos un modelo generativo cuántico basado en polinomios de Chebyshev. El objetivo principal de la mayoría de los modelos generativos cuánticos es

permitir el muestreo en la base computacional, que consiste en los estados ortonormales $\{|x_j\rangle\}_{j=0}^{2^N-1}$ que satisfacen la condición delta de Kronecker $\langle x_j | x_{j'} \rangle = \delta_{jj'}$. Sin embargo, entrenar un modelo directamente en este espacio suele ser difícil, ya que requiere un control muy fino sobre las distintas probabilidades. Para abordar esto, se puede construir y entrenar el modelo en una base ortonormal diferente y luego mapearlo a la base computacional para realizar el muestreo. Con esto en mente, empleamos la base ortonormal de Chebyshev y su correspondiente transformada para construir nuestro modelo generativo [341].

Primero, construimos el mapa de características adecuado para codificar los datos en el espacio de Chebyshev. Así, codificamos un estado cuántico (no normalizado) $|\tau(x)\rangle$ cuyas amplitudes estén determinadas por polinomios de Chebyshev de primer tipo, definidos como $T_k(x) = \cos(k \arccos(x))$, donde k denota el grado. Este estado toma la forma

$$|\tau(x)\rangle = \frac{1}{2^{N/2}} T_0(x) |\emptyset\rangle + \frac{1}{2^{(N-1)/2}} \sum_{k=1}^{2^N-1} T_k(x) |k\rangle, \quad (8.23)$$

donde la amplitud del estado cero está ponderada por $T_0(x) = 1$. Estos polinomios satisfacen una relación de ortogonalidad discreta en puntos de muestreo específicos, conocidos como nodos de Chebyshev:

$$\sum_{j=0}^{2^N-1} T_k(x_j^{\text{Ch}}) T_\ell(x_j^{\text{Ch}}) = \begin{cases} 0 & \text{si } k \neq \ell, \\ 2^N & \text{si } k = \ell = 0, \\ 2^{N-1} & \text{si } k = \ell \neq 0. \end{cases} \quad (8.24)$$

Aquí, los nodos de Chebyshev están dados por $x_j^{\text{Ch}} = \cos(\pi(2j+1)/2^{N+1})$, correspondientes a las raíces de los polinomios de Chebyshev. Los estados resultantes $\{|\tau(x_j^{\text{Ch}})\rangle\}_{j=0}^{2^N-1}$ forman un conjunto ortonormal, con $\langle \tau(x_j^{\text{Ch}}) | \tau(x_{j'}^{\text{Ch}}) \rangle = \delta_{jj'}$.

Cabe señalar que, a diferencia de las bases computacional o de Fourier, que se basan en mallas uniformes, los nodos de Chebyshev forman una malla no uniforme dentro del intervalo $(-1, 1)$. Fuera de estos puntos nodales, los estados $|\tau(x)\rangle$ ya no son ortogonales. Se puede derivar una expresión analítica para su superposición al cuadrado cuando uno de los argumentos se fija en un nodo de Chebyshev:

$$\left| \langle \tau(x') | \tau(x) \rangle \right|^2 = \frac{[T_{2^{N+1}-1}(x') T_{2^N}(x) - T_{2^N}(x') T_{2^{N+1}-1}(x)]^2}{2^{2N} (x' - x)^2}. \quad (8.25)$$

Esta identidad se deduce de la fórmula de Christoffel-Darboux para los polinomios de Chebyshev [350]. Dado que estos estados no están normalizados en general, generarlos con un circuito cuántico implica usar un qubit auxiliar y normalizar el estado después de su preparación. Denotamos la versión normalizada como

$$|\tilde{\tau}(x)\rangle = \frac{|\tau(x)\rangle}{\sqrt{\langle \tau(x) | \tau(x) \rangle}}, \quad (8.26)$$

que coincide con la versión no normalizada exactamente en los nodos de Chebyshev y se le aproxima a medida que N crece.

El siguiente paso es diseñar un circuito cuántico que prepare el estado normalizado $|\tilde{\tau}(x)\rangle$, al que llamaremos mapa de características. Las amplitudes pueden ser incorporadas mediante

una combinación de funciones exponenciales, cada una de las cuales puede implementarse usando un mapa de fase [351]. Estos componentes pueden combinarse utilizando el marco de Combinación Lineal de Unitarios (LCU) [352], donde los términos unitarios son controlados por el estado de un qubit auxiliar. La interferencia deseada se logra seleccionando la medición del auxiliar en el estado $|0\rangle$. En particular, la identidad $e^{-ix} = e^{ix} \cdot e^{-i2x}$ permite controlar solo uno de los componentes exponenciales, simplificando la implementación. Este enfoque permite construir combinaciones de exponenciales con pesos iguales, escaladas apropiadamente para reproducir las amplitudes asociadas con los polinomios de Chebyshev $T_k(x)$. Para ajustar correctamente la amplitud del término constante $T_0(x)$, se requiere una corrección adicional. Esto se logra aplicando una única iteración de un circuito de rotación tipo Grover [51], que rota el estado alrededor de $|\emptyset\rangle$ por un ángulo fijo. Dado que la medición del qubit auxiliar conmuta con esta rotación, colocamos la medición al final del circuito, completando la preparación del mapa de características deseado.

Una vez que hemos establecido cómo codificar los datos en la base de Chebyshev, necesitamos una transformación entre esta y la base computacional, y viceversa. Una vez que una base ortonormal está preparada, existe una correspondencia biunívoca (biyección) y una transformación unitaria que la mapea a cualquier otra base ortonormal. En este contexto, definimos la transformada de Chebyshev como

$$\hat{U}_{\text{QChT}} = \sum_{j=0}^{2^N-1} |\tau(x_j^{\text{Ch}})\rangle \langle x_j|, \quad (8.27)$$

que mapea los estados de la base computacional $|x_j\rangle$ a los estados de Chebyshev $|\tau(x_j^{\text{Ch}})\rangle$. La transformada de Chebyshev puede verse como una variante de la transformada del coseno [353]. Específicamente, el vector de amplitudes del estado $|\tau(x_j^{\text{Ch}})\rangle$ corresponde a la columna $(j+1)$ -ésima de la matriz de la transformada discreta del coseno de tipo II, DCT_N^{II} , definida como

$$\text{DCT}_N^{\text{II}} \equiv 2^{-(N-1)/2} \left\{ c_k \cos \left[\frac{k(j+1/2)\pi}{2^N} \right] \right\}_{k,j=0}^{2^N-1}, \quad (8.28)$$

donde $c_0 = 1/\sqrt{2}$ y $c_k = 1$ para $k \neq 0$. Esta matriz está estrechamente relacionada con la transformada de Fourier, pero requiere mezclar e interferir sus elementos, lo que motiva el diseño del circuito extendido basado en la QFT.

El circuito comienza con una compuerta de Hadamard aplicada al qubit auxiliar (el bit más significativo), seguida por una escalera de compuertas CNOT, que en conjunto preparan un estado entrelazado tipo gato. A continuación, se aplica una transformada cuántica de Fourier a todos los $N+1$ qubits. Esto es seguido por una secuencia de operaciones unitarias que separan y alinean las partes real e imaginaria del estado. En particular, compuertas locales U_1 y R_Z se utilizan para ajustar las fases relativas de los componentes $|0\rangle_a|\Phi\rangle$ y $|1\rangle_a|\Phi\rangle$, donde $|\Phi\rangle$ representa un estado intermedio arbitrario de N qubits.

A continuación, se aplica un subcircuito de permutación para reordenar las amplitudes de manera apropiada, seguido por otra escalera de CNOTs. El circuito concluye con un ajuste global de fase mediante U_2 y un conjunto de compuertas R_X controladas por múltiples qubits, con el fin de escalar las amplitudes de salida de modo que $|0\rangle_a|\Phi\rangle$ y $|1\rangle_a|\Phi\rangle$ sean puramente reales e imaginarias, respectivamente. Observamos que el qubit auxiliar comienza y termina en el estado $|0\rangle$, asegurando una ejecución “limpia”.

Finalmente, el mapa de características completo se obtiene componiendo el embedding con la transformada como $\hat{U}_f(x) = \hat{U}_{\text{QChT}}^\dagger \hat{U}_\tau(x)$.

Una vez que hemos definido un protocolo para codificar datos en los estados base de Chebyshev y un mapeo entre esta base y la base computacional, podemos construir un modelo probabilístico que busque aprender en la base de Chebyshev y muestrear en la base computacional. El funcionamiento de este Modelo Probabilístico Cuántico de Chebyshev (QCPM, por sus siglas en inglés) se muestra en la Fig. 8.2.

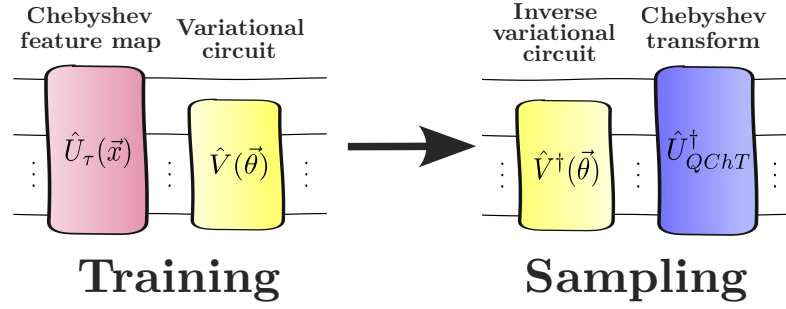


Figure 8.2: Funcionamiento del QCPM. *Izquierda:* El circuito de entrenamiento codifica x utilizando un mapa de características de Chebyshev, e incluye un circuito variacional cuyos parámetros se ajustarán para reproducir el comportamiento de la distribución objetivo. *Derecha:* El circuito de muestreo realiza las operaciones inversas del circuito variacional entrenado y de la transformada cuántica de Chebyshev, generando datos que siguen la distribución utilizada durante el entrenamiento.

Una de las características más interesantes de este QCPM es que permite entrenar los datos sobre la distribución objetivo usando una cantidad limitada de qubits, lo cual normalmente implica una densidad de muestreo limitada, y luego realizar un muestreo extendido utilizando una QChT con qubits adicionales. Esto permite una “interpolación cuántica” natural que incrementa la resolución hasta el nivel deseado.

Resultados

Los resultados más relevantes de esta tesis pueden dividirse en tres secciones que se corresponden con la aplicación de las tres metodologías introducidas anteriormente.

Clusterización de jets

En el Capítulo 4 aplicamos los algoritmos cuánticos descritos en la metodología de cálculo de distancia y cálculo de mínimos a problemas reales de clusterización de jets con datos de simulaciones de eventos que ocurren en las colisiones de partículas del Gran Colisionador de Hadrones (LHC) del CERN.

En particular, simulamos un evento de n partículas en el LHC utilizando un generador de eventos de espacio de fases personalizado, implementado en C++ y basado en ROOT [185]. Este generador puede manejar eventos con hasta decenas de miles de partículas, permitiendo una elección flexible de configuraciones en el estado final, incluyendo partones de QCD sin masa, partones de QCD con fotones, bosones vectoriales masivos o quarks top. Concretamente, estudiamos la producción de estados finales sin masa de n partículas en colisiones protón-protón, aunque nuestro estudio también es aplicable a colisionadores e^+e^- , a una energía en el centro de masas de $\sqrt{s} = 14$ TeV. La selección del estado final sigue los siguientes criterios: los jets se identifican utilizando el algoritmo de jets k_T con

un momento transversal mínimo de $p_{T,\min} \geq 10$ GeV y un radio de $R = 1$. Nos centramos en eventos con $n = 128$ partículas sin masa en el estado final.

Los resultados de aplicar los algoritmos K-means, Affinity Propagation y anti k_T -jet a estos datos aparecen reflejados en la Fig 8.3.

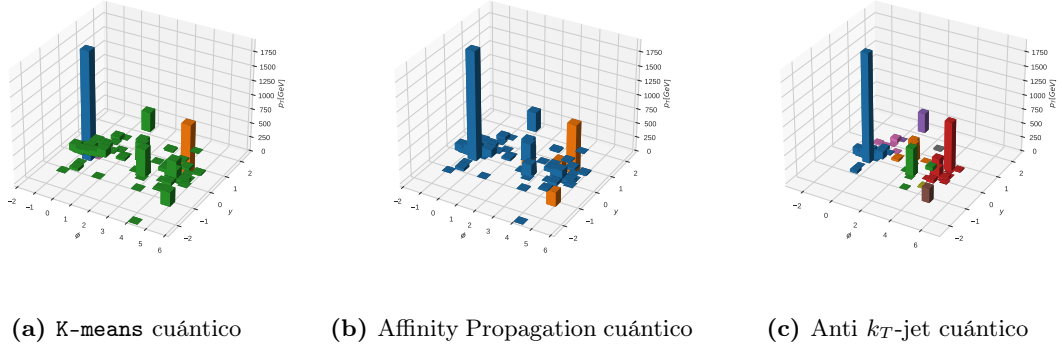


Figure 8.3: Proceso de clusterización de jets de distintos algoritmos cuánticos.

En Fig. 8.3 podemos observar como cada algoritmo clasifica los datos de forma distinta y en un número de jets diferente. En concreto podría parecer que la Fig. 8.3 (c) realiza una agrupación más adecuada con un mayor número de jets. Esto es un efecto visual, ya que la agrupación de jets se representa gráficamente en tres dimensiones, lo cual coincide con la dimensionalidad de la métrica k_T , mientras que K-means y Affinity Propagation utilizan una distancia de Minkowski en cuatro dimensiones, lo que dificulta una representación visualmente atractiva en tres dimensiones.

Por otra parte, la comparación más adecuada es la de estos algoritmos cuánticos con sus análogos clásicos, ya que se trata de métodos probabilísticos cuyo mejor resultado posible es replicar el comportamiento de los algoritmos clásicos, que son deterministas. Estos resultados pueden verse en la Tabla 8.1.

	K-means cuántico	Affinity Propagation cuántico	k_T jet cuántico	Anti- k_T jet cuántico	Cam/Aachen jet cuántico
ε_c	0.94	1.00	0.98	0.99	0.98

Table 8.1: Eficiencias de los algoritmos de clusterización de jet al compararse con los resultados que obtienen sus análogos clásicos.

En la Tabla 8.1 se recogen las eficiencias de los algoritmos clásicos, incluyendo las versiones de k_T -jet que no se habían incluido gráficamente en la Fig. 8.3. Los resultados muestran que en todos los casos las eficiencias son muy altas, cercanas a 1, por lo que podemos concluir que los métodos cuánticos, pese a ser probabilísticos no conducen a una clusterización significativamente diferente de los métodos clásicos. Finalmente, la complejidad temporal de estos algoritmos es comparada con la de sus análogos clásicos en función de que subrutinas cuánticas se utilizan como se muestra en la Tabla 8.2.

Algoritmo de clusterización de jets	Subrutina cuántica	Versión clásica	Versión cuántica
K-means	Ambos	$\mathcal{O}(NKd)$	$\mathcal{O}(N \log K \log(d-1))$

AP	Distancia	$\mathcal{O}(N^2 T d)$	$\mathcal{O}(N^2 T \log(d-1))$
Anti- k_T Jet	Máximo	$\mathcal{O}(N^2)$	$\mathcal{O}(N \log N)$
Anti- k_T FastJet	Máximo	$\mathcal{O}(N \log N)$	$\mathcal{O}(N \log N)$

Table 8.2: Complejidad temporal de los algoritmos clásicos y cuánticos de clusterización de jets cuando se aplica una o ambas subrutinas cuánticas.

Integración de diagramas de Feynmann y tasas de desintegración

En el Capítulo 5 empleamos el algoritmo de integración cuántica QFIAE para integrar secciones eficaces de procesos elementales de física de partículas, integrales de Feynman y tasas de desintegración de partículas elementales a segundo orden en teoría de perturbaciones.

La primers integral de Feynman que abordamos es el integral de un lazo (loop) del tipo renacuajo (tadpole):

$$\mathcal{A}_1^{(1)}(m) = \int_{\ell} \frac{1}{\ell^2 - m^2 + i0} , \quad (8.29)$$

con masa interna m . El factor $i0$ corresponde a la prescripción compleja habitual de Feynman para la continuación analítica en distintas regiones cinemáticas. El momento de lazo a integrar es ℓ . En LTD (Loop-Tree Duality), la expresión matemática correspondiente tiene soporte sobre el momento de lazo tridimensional, ya que la componente energética se integra [229]. Si se introduce un contratérmino ultravioleta local, entonces su representación en LTD es

$$\mathcal{A}_1^{(1,R)}(m; \mu_{UV}) = -\frac{1}{2} \int_{\ell} \left[\frac{1}{q_{1,0}^{(+)} } - \frac{1}{q_{UV,0}^{(+)} } \left(1 + \frac{\mu_{UV}^2 - m^2}{2(q_{UV,0}^{(+)})^2} \right) \right] , \quad (8.30)$$

la cual está bien definida en cuatro dimensiones físicas, donde la medida de integración es $\int_{\ell} = \int d^3\ell / (2\pi)^3$. En la Ecuación (8.30), definimos las energías en la capa de masa como $q_{1,0}^{(+)} = \sqrt{\ell^2 + m^2 - i0}$ y $q_{UV,0}^{(+)} = \sqrt{\ell^2 + \mu_{UV}^2 - i0}$, donde μ_{UV} es la escala de renormalización. La integral renacuajo es unidimensional porque el integrando es independiente del ángulo sólido, y solo depende del módulo del momento tridimensional. El cambio de variable

$$|\ell| = \frac{mz}{1-z} , \quad z \in [0, 1) , \quad (8.31)$$

remapea la variable de integración a un rango finito donde se define la serie de Fourier de la función objetivo.

Para implementar QFIAE en una computadora cuántica real, debemos considerar que los dispositivos de la era NISQ se enfrentan a diversas fuentes de ruido, desde efectos cuánticos como la decoherencia hasta errores específicos del hardware, incluyendo errores en las compuertas, lectura y calibración. Trabajando con la misma tecnología (qubits superconductores), abogamos por una estrategia de implementación independiente del hardware. Específicamente, proponemos una aproximación donde los dos módulos de QFIAE se implementan en dos computadores cuánticos distintos suministrados por diferentes proveedores. Esto permite minimizar el impacto del ruido específico del hardware en el desempeño algorítmico total.

Una QNN de un qubit es entrenada usando una versión actualizada del método de descenso por gradiente Adam. El algoritmo se ha escrito usando Qibo, mientras que Qibolab [280]

y Qibocal [281] se usan respectivamente para controlar y calibrar el dispositivo cuántico superconductor de 5 qubits alojado en el Quantum Research Centre (QRC) del Technology Innovation Institute (TII). Por otro lado, un algoritmo IQAE de 5 qubits ha sido ejecutado usando Qiskit [282] en el dispositivo superconductor IBM Quantum de 27 qubits *ibmq_mumbai*.

Fourier	IQAE	$\mathcal{A}_1^{(1,R)}(m; \mu_{UV})$ $m = 5, \mu_{UV} = m/2$	$\mathcal{A}_1^{(1,R)}(m; \mu_{UV})$ $m = 5, \mu_{UV} = 2m$
C	C	-0.106	-0.258
S	S	-0.101(3)	-0.254(9)
S	Q	-0.108(4)	-0.270(12)
Q	S	-0.105(2)	-0.252(6)
Q	Q	-0.106(3)	-0.270(9)
Analítico		-0.1007	-0.2554

Table 8.3: Integral renormalizada del tipo renacuajo $\mathcal{A}_1^{(1,R)}(m; \mu_{UV})$ evaluada en Qibo e IBM Quantum, como función del cociente μ_{UV}/m , donde se fija $m = 5$ GeV. La descomposición de Fourier y la integración mediante IQAE se realizan ya sea con un simulador cuántico (S) o en un dispositivo cuántico real (Q). En la primera fila ambas componentes se realizan con métodos clásicos (C).

Los resultados presentados en la Tabla 8.3 muestran una desviación relativamente pequeña con respecto al valor analítico tanto para $\mu_{UV} = m/2$ como para $\mu_{UV} = 2m$. En particular, el acuerdo con los valores analíticos cuando ambos módulos del algoritmo se ejecutan sobre un computador cuántico (penúltima fila) es mejor que 1.7 desviaciones estándar en todos los casos. Este resultado representa un logro significativo dado el estado actual de la tecnología de computación cuántica. Además, hay que tener en cuenta que se trata de la primera aplicación de un algoritmo cuántico extremo a extremo ejecutado en un computador cuántico para la estimación de integrales de lazos de Feynman. Parte de la desviación se debe también a la propia aproximación por serie de Fourier, como se observa en la primera fila de la Tabla 8.3, lo cual constituye una prueba de la robustez del algoritmo cuántico propuesto.

A continuación procedemos a integrar una integral de Feynmann más compleja, tipo burbuja (bubble) con singularidades de umbral (threshold). Su representación en LTD es [228]

$$\mathcal{A}_2^{(1,R)}(p, m_1, m_2) = \int_{\ell} \left[\frac{1}{x_2} \left(\frac{1}{\lambda^+} + \frac{1}{\lambda^-} \right) - \frac{1}{4(q_{UV,0}^{(+)})^3} \right], \quad (8.32)$$

donde $x_2 = \prod_{i=1,2} 2q_{i,0}^{(+)}$, $\lambda^{\pm} = \sum_{i=1,2} q_{i,0}^{(+)} \pm p_0$, y las energías on-shell están dadas por $q_{i,0}^{(+)} = \sqrt{\ell^2 + m_i^2 - i0}$ con $i \in \{1, 2\}$, asumiendo que el momento externo tiene componentes espaciales nulas, $p = (p_0, \mathbf{0})$. Si $p_0^2 < (m_1 + m_2)^2$, la integral es puramente real. En caso contrario, aparece una contribución imaginaria debida a la singularidad en el umbral unitario en $\lambda^- \rightarrow 0$, suponiendo $p_0 > 0$. Para tratar esta singularidad de umbral, es conveniente introducir una deformación de contorno δ en el plano complejo, con el fin de suavizar el comportamiento de la función en las cercanías del umbral sin alterar el resultado de la integral. Consideramos el caso $m_1 = m_2 = m$ y procedemos a estimar la integral usando dos redes neuronales cuánticas (QNNs) diferentes para ajustar por separado las partes real e imaginaria del integrando, e integrar cada serie de Fourier

utilizando IQAE. Hemos implementado QFIAE en dos simuladores cuánticos diferentes. Para las QNN usamos **PennyLane** [61], mientras que para IQAE empleamos **Qibo**. Los resultados se muestran en la Fig. 8.4, donde se representan las partes real e imaginaria en los gráficos izquierdo y derecho, respectivamente. Estos resultados muestran un acuerdo entre los valores integrados por QFIAE y los resultados analíticos.

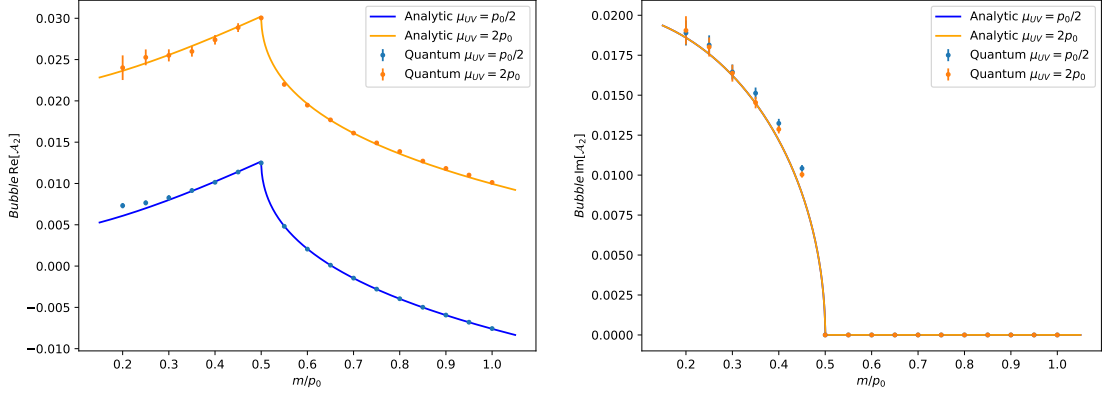


Figure 8.4: Integración cuántica de la parte real (izquierda) e imaginaria (derecha) de la integral de burbuja renormalizada $\mathcal{A}^{(1,R)}_2(p, m, m; \mu_{UV})$ como función del cociente entre la masa m y la componente de energía del momento externo, fijado en $p_0 = 100$ GeV, y de la escala de renormalización μ_{UV} .

En tercer lugar, consideramos una integral bidimensional de Feynmann de tipo triángulo (triangle). La representación LTD de esta integral está dada por [248]

$$\begin{aligned} \mathcal{A}_3^{(1)}(p_1, p_2, m_1, m_2, m_3) = & - \int_{\ell} \frac{1}{x_3} \left(\frac{1}{\lambda_{12}^- \lambda_{23}^+} \right. \\ & \left. + \frac{1}{\lambda_{23}^+ \lambda_{31}^-} + \frac{1}{\lambda_{31}^- \lambda_{12}^+} + (\lambda_{ij}^+ \leftrightarrow \lambda_{ij}^-) \right), \end{aligned} \quad (8.33)$$

con $x_3 = \prod_{i=1}^3 2q_{i,0}^{(+)}$, donde ahora las energías son $q_{1,0}^{(+)} = \sqrt{(\ell + \mathbf{p}_1)^2 + m_1^2 - i0}$ y $q_{i,0}^{(+)} = \sqrt{\ell^2 + m_i^2 - i0}$ para $i \in \{2, 3\}$. Trabajamos en el sistema del centro de masas, donde $\mathbf{p}_{12} = \mathbf{p}_1 + \mathbf{p}_2 = 0$, $p_{12,0} = p_{1,0} + p_{2,0} = \sqrt{s}$, y los momentos externos p_1 y p_2 están en direcciones opuestas a lo largo del eje z . Los denominadores causales son

$$\begin{aligned} \lambda_{31}^{\pm} &= q_{3,0}^{(+)} + q_{1,0}^{(+)} \pm p_{1,0}, \quad \lambda_{12}^{\pm} = q_{1,0}^{(+)} + q_{2,0}^{(+)} \pm p_{2,0}, \\ \lambda_{23}^{\pm} &= q_{2,0}^{(+)} + q_{3,0}^{(+)} \mp p_{12,0}. \end{aligned} \quad (8.34)$$

Las variables de integración en Eq. (8.33) son el módulo del momento de lazo tridimensional y su ángulo polar respecto a p_1 , considerando que la integración azimutal es trivial. Esto implica que la descomposición en series de Fourier depende de dos variables y debemos integrar cada una por separado. También aplicamos una deformación del contorno para tratar la singularidad de umbral unitaria en $\lambda_{23}^+ \rightarrow 0$, cuando $s > (m_2 + m_3)^2$ que suaviza el comportamiento del integrando cerca de la singularidad de umbral y mejora significativamente la calidad de la descomposición en Fourier. Las estimaciones de la integral del triángulo, obtenidas en los simuladores **PennyLane** y **Qibo**, se muestran en las Figs. 8.5(izquierda) y 8.5(derecha), ilustrando las partes real e imaginaria, respectivamente.

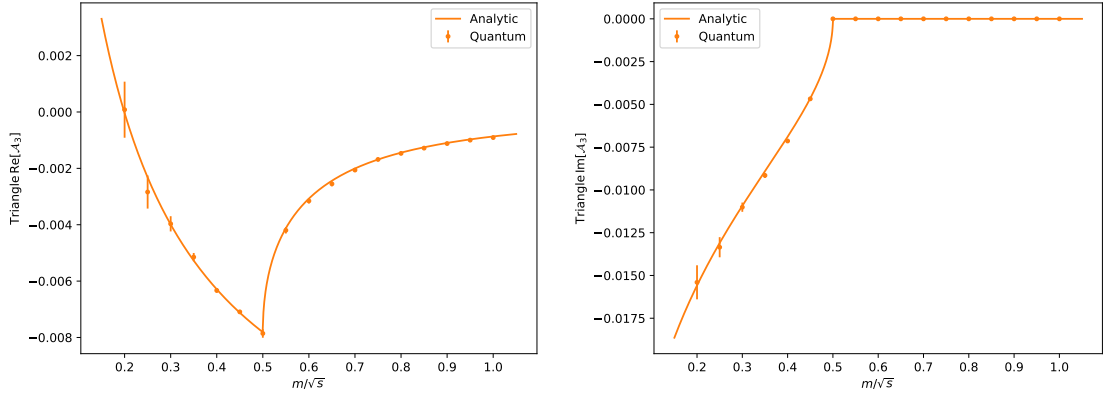


Figure 8.5: Integración cuántica de la parte real (izquierda) e imaginaria (derecha) de la integral del triángulo $\mathcal{A}_3^{(1)}(p_1, p_2, m, m, m)$ como función del cociente entre la masa m y la energía del centro de masas fijada en $\sqrt{s} = 2$ GeV.

A partir de aquí, podemos ir un paso más allá y estudiar aplicaciones del algoritmo QFIAE de integrales que se correspondan con observables físicos en las desintegraciones de partículas elementales. Para ello hacemos uso del método causal unitario de LTD que emplea amplitudes del vacío cuántico para expresar integrales de Feynmann. Con este formalismo podemos escribir la tasa de desintegración a segundo orden en teoría de perturbaciones de una partícula elemental a :

$$d\Gamma_a^{(1)} = \frac{d\Phi_{\ell_1 \ell_2}}{2\sqrt{s}} \left[\left(\mathcal{A}_D^{(3,a)}(456) \tilde{\Delta}_{45\bar{6}} + \mathcal{A}_D^{(3,a)}(1356) \tilde{\Delta}_{135\bar{6}} \right) + (5 \leftrightarrow 2, 4 \leftrightarrow 3) \right], \quad (8.35)$$

donde $\mathcal{A}_D^{(3,a)}(456)$ y $\mathcal{A}_D^{(3,a)}(1356)$ son los residuos en el espacio de fases de la amplitud de vacío en LTD, cuando $\lambda_{456} \rightarrow 0$ y $\lambda_{1356} \rightarrow 0$. Estas contribuciones representan, respectivamente, las fluctuaciones cuánticas perturbativas a un lazo con dos partículas en el estado final y a nivel árbol con tres partículas en el estado final. En una teoría escalar:

$$\begin{aligned} \mathcal{A}_D^{(3,\Phi)}(456) &= \frac{g_\Phi^{(1)} m_\Phi^2}{x_{12345}} \left(L_{2345,125}^{134} + L_{2345,134}^{125} + L_{134,125}^{2345} \right), \\ \mathcal{A}_D^{(3,\Phi)}(1356) &= \frac{g_\Phi^{(1)} m_\Phi^2}{x_{135}} \left(\frac{1}{\lambda_{134} \lambda_{134} \lambda_{125} \lambda_{125}} \right), \end{aligned} \quad (8.36)$$

donde $x_{i_1 \dots i_n} = \prod_{s=1}^n 2q_{i_s,0}^{(+)}$ y $L_{j,k}^i = \lambda_i^{-1} (\lambda_j^{-1} + \lambda_k^{-1})$, con

$$\lambda_{i_1 \dots i_r \bar{i}_{r+1} \dots \bar{i}_n} = \lambda_{i_1 \dots i_r} - \lambda_{i_{r+1} \dots i_n}. \quad (8.37)$$

La medida de integración es

$$d\Phi_{\ell_1 \ell_2} = \prod_{j=1}^2 \frac{d^3 \ell_j}{(2\pi)^3}, \quad (8.38)$$

y usando simetría rotacional y la delta de Dirac:

$$\tilde{\Delta}_{i_1 \dots i_n \bar{a}} = 2\pi \delta(\lambda_{i_1 \dots i_n \bar{a}}), \quad (8.39)$$

lo que reduce la integral a dos variables independientes:

$$d\Phi_{\ell_1\ell_2} \rightarrow \frac{1}{4\pi^4} \int_0^\infty \ell_1^2 d|\ell_1| \int_0^\infty \ell_2^2 d|\ell_2| \int_0^1 dv, \quad (8.40)$$

con

$$\tilde{\Delta}_{45\bar{6}} = 2\pi \delta \left(\sqrt{\ell_2^2 + m^2} - \sqrt{s} \right), \quad (8.41)$$

y

$$\tilde{\Delta}_{135\bar{6}} = 2\pi \delta \left(|\ell_1 + \ell_2| + \sqrt{\ell_1^2 + m^2} + \sqrt{\ell_2^2 + m^2} - \sqrt{s} \right), \quad (8.42)$$

donde $|\ell_1 + \ell_2| = \sqrt{\ell_1^2 + \ell_2^2 + 2(1-2v)|\ell_1||\ell_2|}$. Las deltas se usan para resolver una variable en términos de las otras dos.

La ventaja clave de esta formulación es que permite tratar simultáneamente contribuciones a lazo y árbol, lo que garantiza cancelaciones locales de singularidades sin necesidad de usar regularización dimensional [263, 264], y además produce integrandos más suaves y eficientes numéricamente.

Hemos integrado estas tasas de desintegración para diferentes procesos. Los resultados que se presentan en la Fig. 8.6 muestran una desviación relativamente pequeña respecto a sus valores analíticos correspondientes en Regularización Dimensional estándar (DREG). Notablemente, el panel izquierdo presenta una desviación sistemática en comparación con el panel derecho, atribuida al ruido del hardware que persiste a pesar de las técnicas de mitigación de errores aplicadas.

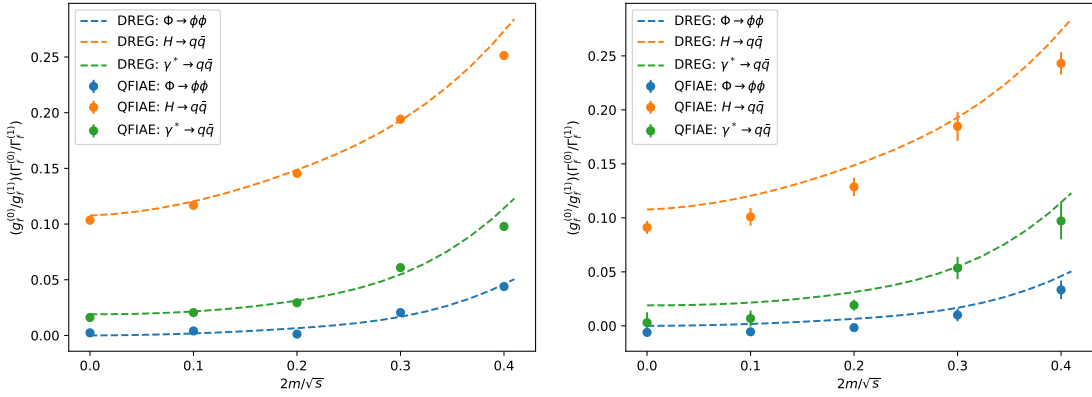


Figure 8.6: Tasas de desintegración integradas cuánticamente para los tres procesos $H \rightarrow q\bar{q}(g)$, $\gamma^* \rightarrow q\bar{q}(g)$ y $\Phi \rightarrow \phi\phi(\phi)$ a segundo orden en teoría de perturbaciones, en función de la masa del estado final. El panel izquierdo muestra resultados obtenidos con un simulador cuántico, mientras que el derecho presenta resultados en los que el algoritmo IQAE se ejecutó en hardware cuántico. Los cálculos emplean QFIAE con el enfoque causal unitario de LTD, y las líneas discontinuas representan las predicciones teóricas en regularización dimensional.

Generación de funciones de fragmentación

En el Capítulo 6 empleamos el método QCPM para aprender y generar funciones de fragmentación de diferentes partones que hadronizan en piones y kaones. Concretamente, utilizamos el conjunto de datos NNFF10_PISum_nnlo para piones ($\pi^\pm = \pi^+ + \pi^-$) y el conjunto NNFF10_KAsum_nnlo para kaones ($K^\pm = K^+ + K^-$), ambos a tercer orden

(NNLO) en teoría de perturbaciones. Las funciones de fragmentación (FFs) específicas que se analizan, extraídas de datos de producción de hadrones en aniquilación inclusiva simple electrón-positrón (SIA), involucran un total de cinco combinaciones independientes de FFs:

$$D_g^h, D_{b^+}^h, D_{c^+}^h, D_{d^++s^+}^h, D_{u^+}^h, \quad (8.43)$$

donde h representa el hadrón en el que fragmenta el partón, $D_{q^+}^h \equiv D_q^h + D_{\bar{q}}^h$, y $D_{d^++s^+}^h \equiv D_{d^+}^h + D_{s^+}^h$.

En la Fig. 8.7 presentamos los resultados de una función de fragmentación (FF) en particular como ejemplo de la validez del método, $D_g^{K^\pm}$, que corresponde a la suma de las FF del gluón g fragmentando en el kaón K^+ y su antipartícula K^- .

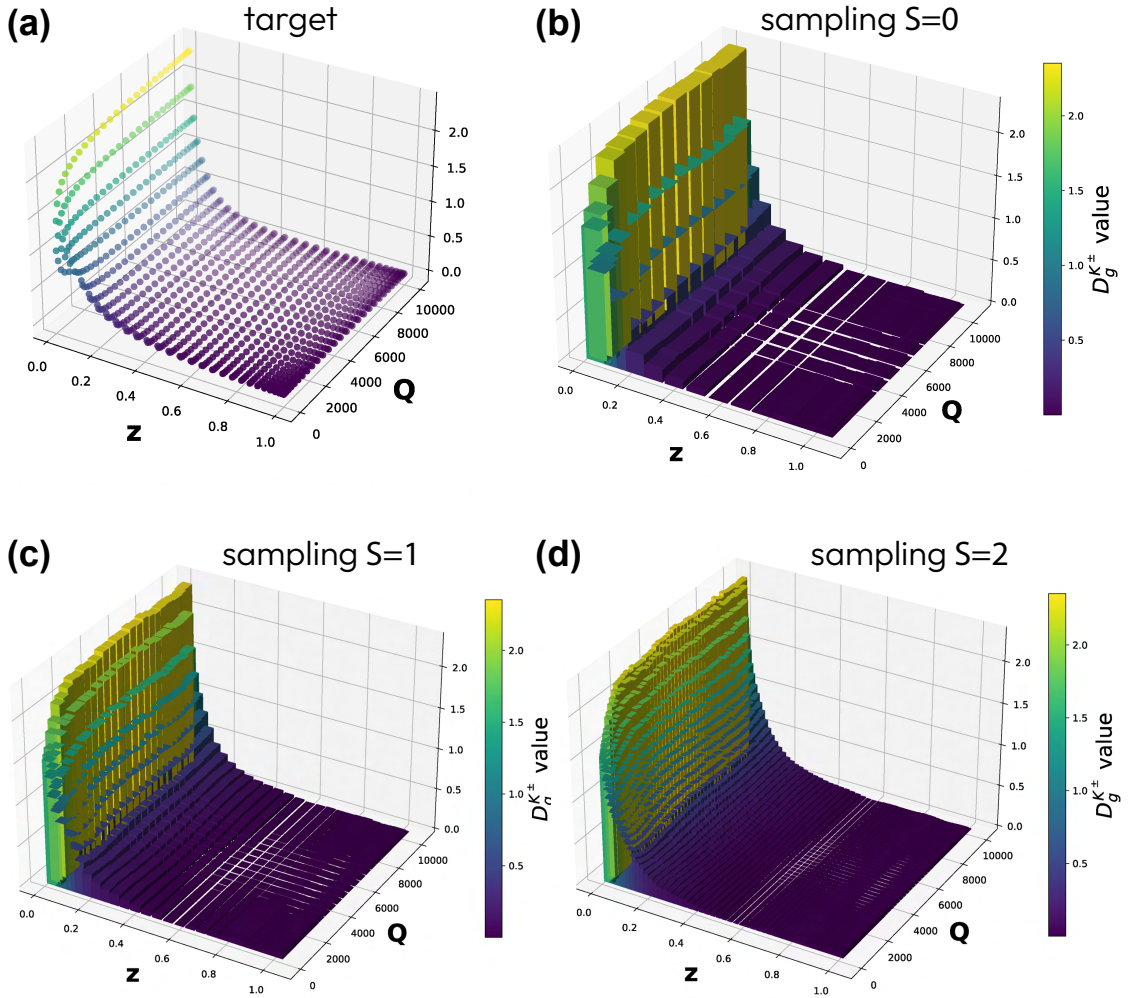


Figure 8.7: Muestreo de $D_g^{K^\pm}(z, Q)$, la FF de un gluón fragmentando en kaones, con $S = 0, 1, 2$ qubits adicionales por variable para interpolar en regiones no entrenadas. (a) Distribución objetivo $D_g^{K^\pm}(z, Q)$. (b) Muestreo del modelo QCPM entrenado de $D_g^{K^\pm}(z, Q)$ con el mismo número de qubits ($S = 0$). (c,d) Muestreo de $D_g^{K^\pm}(z, Q)$ con registro extendido ($S = 1, 2$).

Los resultados en la Fig. 8.7 muestran que QCPM captura correctamente el comportamiento de la función bidimensional $D_g^{K^\pm}(z, Q)$ en la región de interés. La Fig. 8.7(a) muestra la distribución objetivo sobre los datos de entrenamiento. La Fig. 8.7(b) demuestra la capacidad del modelo para generar muestras utilizando los mismos registros

cuánticos empleados durante el entrenamiento. Las Fig. 8.7(c) y 8.7(d) presentan el rendimiento del muestreo cuando el modelo utiliza uno ($S = 1$) y dos ($S = 2$) qubits adicionales por variable, respectivamente, donde el modelo realiza predicciones en regiones no entrenadas y, en general, muestra una excelente correspondencia con los datos de referencia.

Conclusiones

El Modelo Estándar (SM) de la física de partículas es una teoría muy exitosa, con el descubrimiento del bosón de Higgs en 2012, hace 13 años, en el Gran Colisionador de Hadrones (LHC) del CERN, como el mayor avance reciente en la física de altas energías (HEP). Se están planificando experimentos HEP aún más sofisticados para profundizar en el origen del universo y el contenido fundamental de la materia. En este sentido, estos experimentos más complejos exigen herramientas computacionales avanzadas, tanto para analizar las enormes cantidades de datos producidos en las colisiones de alta energía como para generar predicciones teóricas más precisas y exactas. En este contexto, la computación cuántica surge como una tecnología prometedora para complementar los métodos clásicos en el análisis de datos y la generación de simulaciones.

Esta tesis explora diversas aplicaciones de algoritmos cuánticos en física de altas energías. Primero, se presentan algoritmos cuánticos para clusterización de jets como **K-means**, Affinity Propagation y k_T -jet clustering, mostrando ventajas teóricas en escalabilidad y tiempos de ejecución. Posteriormente, se desarrolló un nuevo integrador cuántico basado en estimación de amplitud y redes neuronales cuánticas (QFIAE), que permite integrar funciones multidimensionales, que ha sido aplicado en ordenadores cuánticos actuales a cálculos de integrales de Feynman así como al cálculo de observables físicos como tasas de desintegración a segundo orden perturbativo en Teoría Cuántica de Campos. Finalmente, se emplearon Modelos Probabilísticos Cuánticos de Chebyshev (QCPMs) para aprender y generar funciones de fragmentación, obteniendo unos resultados satisfactorios en precisión y capacidad generativa. Además, el uso de qubits adicionales permitió aumentar de forma exponencial la resolución del muestreo.

En definitiva, este trabajo demuestra el potencial de la computación cuántica para abordar problemas complejos en física de partículas y abre nuevas direcciones hacia aplicaciones más sofisticadas, especialmente en un futuro a medio plazo cuando se disponga de dispositivos cuánticos tolerantes a fallos.

A Controlled *SwapTest*

A well-known procedure for determining the entanglement between two quantum states is the controlled *SwapTest* [164] method. The corresponding quantum circuit associated to the *SwapTest* method is shown in Fig. A.1.

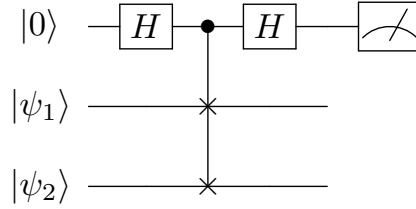


Figure A.1: Quantum circuit of the *SwapTest*.

This method allows us to quantify the overlap between $|\psi_1\rangle$ and $|\psi_2\rangle$, which are two input general quantum states of n and m qubits respectively such that $n \geq m$ (otherwise we exchange the labels 1 and 2), by measuring an ancillary qubit. The controlled *SwapTest* proceeds in three steps starting from the initial state

$$|\Psi_0\rangle = |0, \psi_1, \psi_2\rangle, \quad (\text{A.1})$$

where the ancillary qubit has been initialized to $|0\rangle$. In the first step, a Hadamard (H) gate is applied to the ancillary qubit, while the states to be probed are left unchanged, resulting in the new state

$$|\Psi_1\rangle = \left(H \otimes \mathbf{I}^{\otimes n+m}\right) |\Psi_0\rangle = \frac{1}{\sqrt{2}} (|0, \psi_1, \psi_2\rangle + |1, \psi_1, \psi_2\rangle), \quad (\text{A.2})$$

where the identity $\mathbf{I}^{\otimes n+m}$ acts over the $|\psi_1\rangle$ and $|\psi_2\rangle$ states and the tensor product $\otimes$ is omitted in the composed states (e.g $|0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle = |0, \psi_1, \psi_2\rangle$). A controlled swap gate (CSWAP) is then applied to $|\Psi_1\rangle$ where all the m qubits of $|\psi_2\rangle$ are swapped with the m first qubits of $|\psi_1\rangle$, leading to

$$|\Psi_2\rangle = \text{CSWAP} |\Psi_1\rangle = \frac{1}{\sqrt{2}} (|0, \psi_1, \psi_2\rangle + |1, \psi_2, \psi'_1\rangle), \quad (\text{A.3})$$

where ψ'_i is the swapped state of ψ_i , i.e., a state where the m first qubits of ψ_1 have been swapped with the rest $n - m$ qubits. The final step consist of applying again a Hadamard

gate to the ancillary qubit

$$|\Psi_3\rangle = \left(H \otimes \mathbf{I}^{\otimes n+m}\right) |\Psi_2\rangle = \frac{1}{2} \left(|0\rangle \otimes (|\psi_1, \psi_2\rangle + |\psi_2, \psi'_1\rangle) + |1\rangle \otimes (|\psi_1, \psi_2\rangle - |\psi_2, \psi'_1\rangle) \right). \quad (\text{A.4})$$

The resulting probability of measuring the ancillary qubit in the state $|0\rangle$ is given by

$$\begin{aligned} P_{\Psi_3}(|0\rangle) &= |\langle 0 | \Psi_3 \rangle|^2 = \frac{1}{4} \left| |\psi_1, \psi_2\rangle + |\psi_2, \psi'_1\rangle \right|^2 = \frac{1}{2} + \frac{1}{2} \text{Re} [\langle \psi_2, \psi'_1 | \psi_1, \psi_2 \rangle] \\ &= \frac{1}{2} + \frac{1}{2} \langle \psi'_1 | \psi_2 \rangle \langle \psi_2 | \psi_1 \rangle, \end{aligned} \quad (\text{A.5})$$

which turns out to be as follows if $m = n$, thus $|\psi'_1\rangle = |\psi_1\rangle$

$$P_{\Psi_3}(|0\rangle) = \frac{1}{2} + \frac{1}{2} \left| \langle \psi_1 | \psi_2 \rangle \right|^2, \quad (\text{A.6})$$

and this provides the squared inner product between the two states with an uncertainty of $\mathcal{O}(\epsilon)$ after $\mathcal{O}(\epsilon^{-2})$ shots.

B Training and sampling of fragmentation functions of kaon and pion

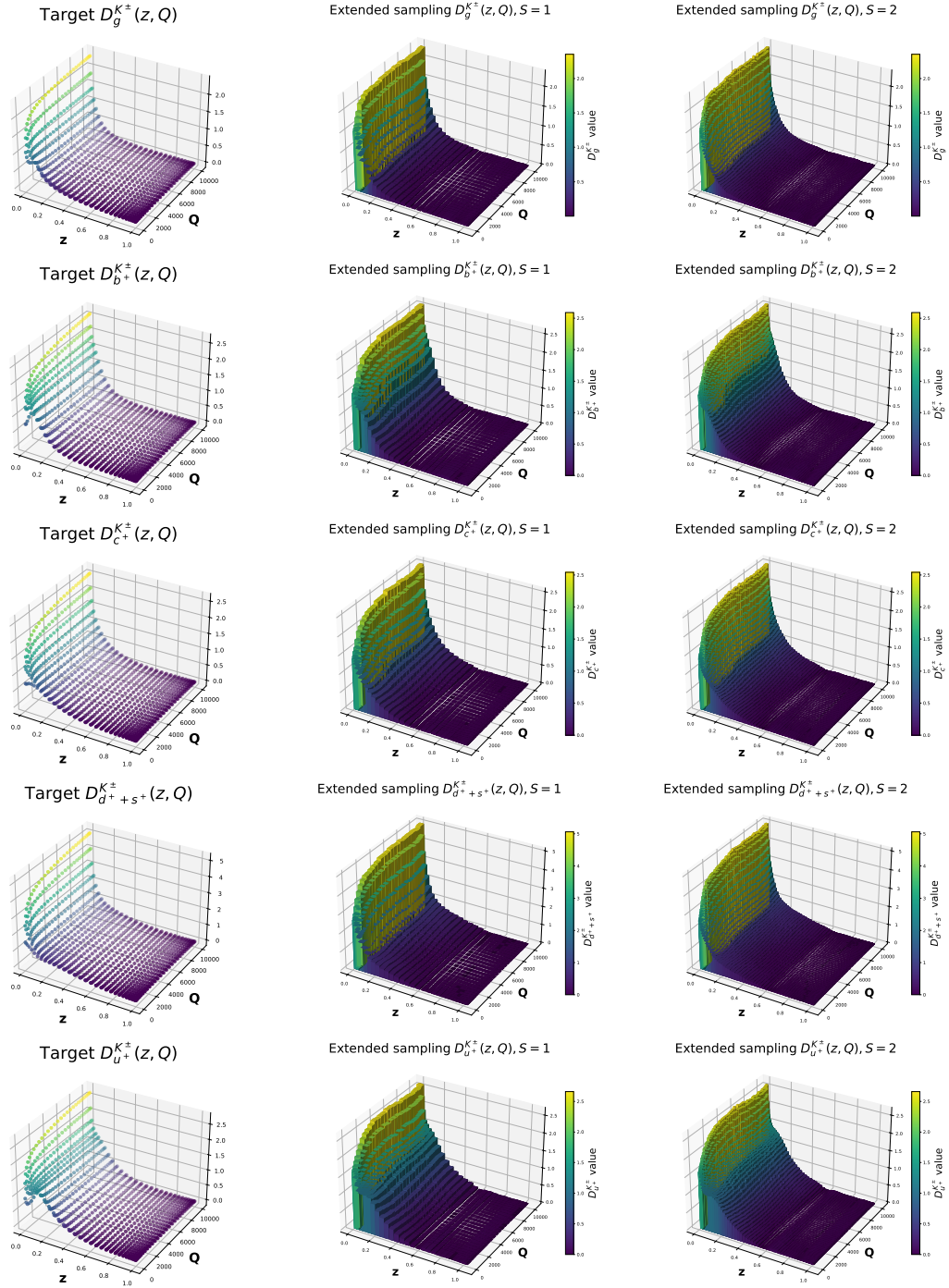
In this Appendix, we present a visualization of the training (implicitly) and sampling (explicitly) processes for all fragmentation functions (FFs) considered in Section 6.4, corresponding to partons fragmenting into kaons and pions. Specifically, the ten FF distributions under study are:

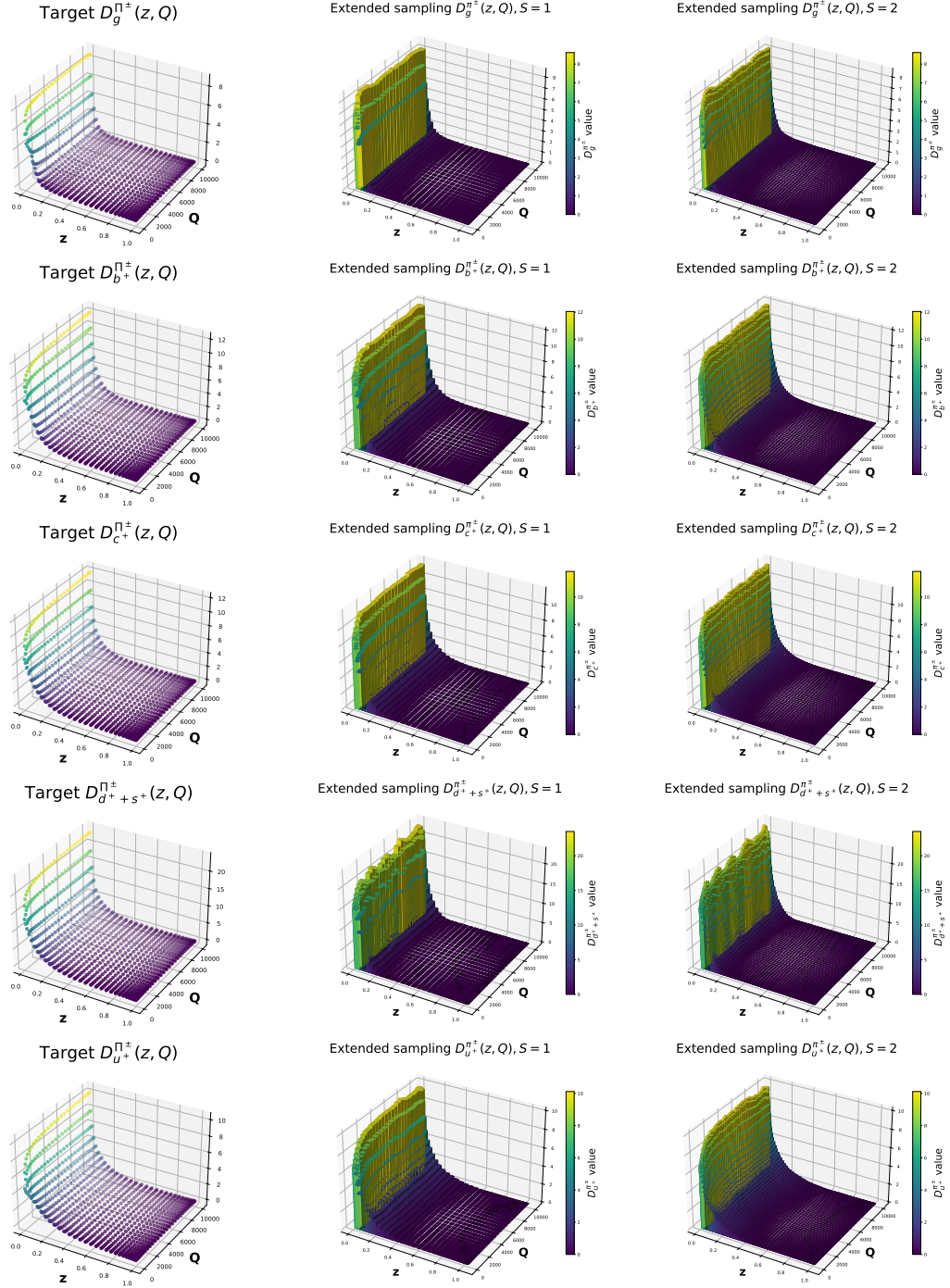
$$D_g^h, D_{b^+}^h, D_{c^+}^h, D_{d^++s^+}^h, D_{u^+}^h, \quad h = K^\pm, \pi^\pm, \quad (\text{B.1})$$

where $D_{q^+}^h \equiv D_q^h + D_{\bar{q}}^h$ and $D_{d^++s^+}^h \equiv D_{d^+}^h + D_{s^+}^h$. Additionally, we define $D_i^{h^\pm} = D_i^{h^+} + D_i^{h^-}$, with i denoting the fragmenting parton.

Figures B.1 and B.2 demonstrate that the quantum model effectively learns the full set of FF distributions and successfully generates samples from them.

Regarding the resources used for sampling, we have employed 10^6 shots to obtain enough data points to provide a smooth description of the target function that correctly mimics the FF behavior. When performing extended sampling with additional qubits for each variable the number of qubits is increased $2S$ times, expanding the Hilbert space. Specifically, for $S = 1$, the Hilbert space increases by a factor of 4, and for $S = 2$, it expands by a factor of 16. This increase in the Hilbert space translates to a need for 4 times more shots when using $S = 1$ and 16 times more shots when using $S = 2$ to maintain the same level of accuracy. Additionally, this enlargement of the Hilbert space results in smaller measurement bins, with the area of each bin becoming 4 times smaller for $S = 1$ and 16 times smaller for $S = 2$.


 Figure B.1: Sampling of fragmentation functions of partons into kaons K^+ and K^- .


 Figure B.2: Sampling of fragmentation functions of partons into pions π^+ and π^- .

C Correlations in Quantum Chebyshev Probabilistic Models

In this Appendix we study the correlations between the variables z and Q corresponding to the momentum fraction and the energy scale of the Fragmentation Functions studied in Section 6.4.

C.1 Von Neumann entropy in quantum registers

In this section, we quantify the entanglement within each of the registers encoding the variables z and Q during the training phase shown in Fig.6.6(a), using the von Neumann entropy. This quantity is a fundamental measure of entanglement for pure bipartite states in quantum information theory[359, 17]. Given a quantum state described by a density matrix ρ , the von Neumann entropy is defined as

$$S(\rho) = -\text{Tr}(\rho \log_2 \rho) = -\sum_i \lambda_i \log_2 \lambda_i, \quad (\text{C.1})$$

where λ_i are the eigenvalues of ρ . For a pure state $|\psi\rangle$, the entropy of either subsystem provides a quantitative measure of entanglement between them [360]. Specifically, in a bipartite setting, the entanglement entropy is given by

$$S(\rho_A) = S(\rho_B) = -\text{Tr}(\rho_A \log_2 \rho_A) = -\text{Tr}(\rho_B \log_2 \rho_B), \quad (\text{C.2})$$

where ρ_A and ρ_B are the reduced density matrices of the respective subsystems A and B [361].

In our setup, we consider two quantum systems, $\mathcal{Z}$ and $\mathcal{Q}$, each represented by $N = 4$ qubits and encoding the variables z and Q , respectively. To analyze entanglement within these systems, we partition each register into two subsystems of $N/2 = 2$ qubits, allowing us to assess internal correlations independently in $\mathcal{Z}$ and $\mathcal{Q}$.

It is important to clarify that we do not compute the entanglement between $\mathcal{Z}$ and $\mathcal{Q}$ as a whole. Since these systems are entangled via controlled- Z (CZ) gates, the combined system remains in a pure state, and its entropy would be zero. Instead, our analysis targets the internal entanglement within each register.

To perform these calculations, we use the `qml.math.vn_entropy` function provided by PennyLane [362]. The corresponding results are presented in Fig. C.1.

The results presented in Fig. C.1(a) show significant variations in the von Neumann entropy within the $\mathcal{Z}$ system across the different circuits that have learned the FFs. However, these variations do not appear to correlate with the inclusion or exclusion of

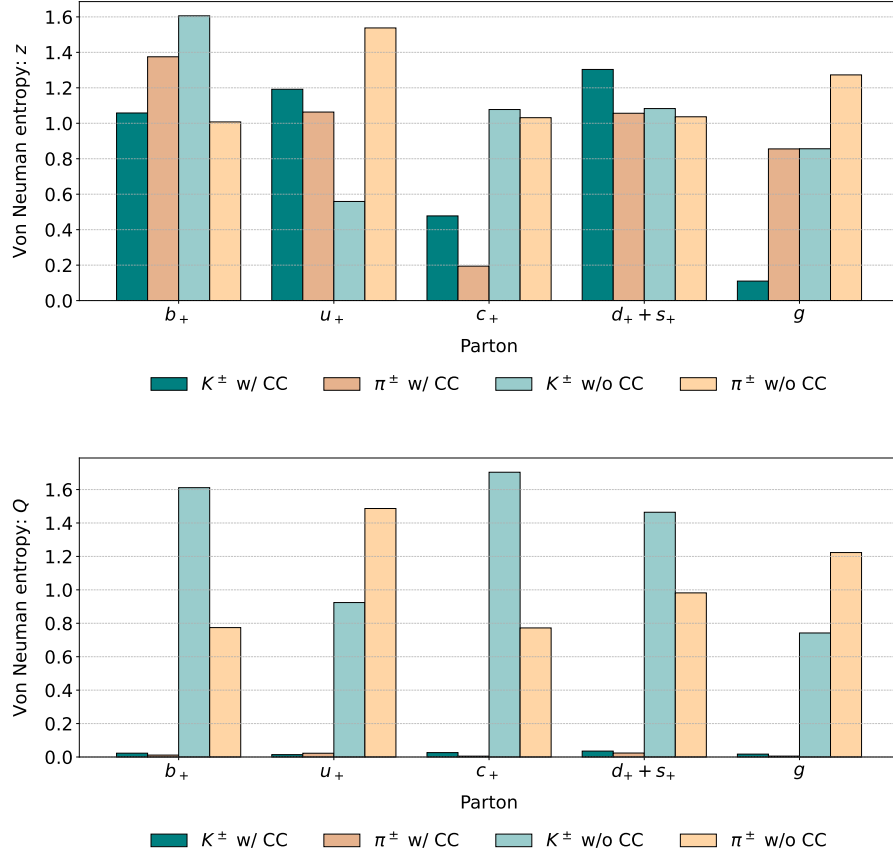


Figure C.1: Von Neumann entropies of the circuits in z registers (top) and Q registers (bottom) with (w/ CC) and without (w/o CC) correlations between z and Q .

the correlations circuit between systems $\mathcal{Z}$ and $\mathcal{Q}$. This suggests that while the internal entanglement structure of the $\mathcal{Z}$ system is dynamic, it is not driven by the inter-system correlations.

On the other hand, the results of Fig. C.1(b) reveal a strong dependence of the von Neumann entropy in the $\mathcal{Q}$ system on the inclusion of the correlations circuit. When the correlations circuit is not present, we observe a high von Neumann entropy, indicating a substantial level of internal entanglement within the $\mathcal{Q}$ system. Nonetheless, when the correlations circuit is included, the von Neumann entropy drops drastically, approaching almost zero. This significant reduction in entropy suggests a transfer or “absorption” of entanglement from the $\mathcal{Q}$ system to the $\mathcal{Z}$ system. Such behavior is consistent with the principle of entanglement monogamy, wherein the entanglement shared between two systems constrains the entanglement each system can maintain internally [363].

C.2 Purity in quantum registers

In quantum information theory, purity serves as a measure of how mixed or coherent a quantum state is. For a system described by a density matrix ρ , purity is defined as

$$\gamma(\rho) = \text{Tr}(\rho^2), \quad (\text{C.3})$$

where ρ is the state's density operator. A purity value of $\gamma(\rho) = 1$ indicates a pure state, meaning the system resides in a well-defined quantum state. Conversely, if $\gamma(\rho) < 1$, the state is mixed, reflecting a probabilistic combination of different quantum states. In the extreme case of a maximally mixed state, purity reaches its minimum value of $\gamma(\rho) = 1/d$, where d denotes the dimension of the system's Hilbert space[364]. In the context of quantum circuits, purity offers useful insight into the system's internal coherence and entanglement structure. Greater mixing generally implies higher entanglement, while lower mixing corresponds to a more separable (and hence less entangled) quantum state.

In this section, we quantify the entanglement between the registers encoding z and Q , during the training phase of Fig. 6.6(a), using their purity. In particular, we calculate the purity of the system $\mathcal{Z}$, of $N = 5$ qubits, that encodes the variable z which by construction is equivalent to the system $\mathcal{Q}$ that encodes the variable Q . To evaluate the purity of $\mathcal{Z}$, we trace out the degrees of freedom associated with $\mathcal{Q}$, effectively isolating the subsystem of interest.

$$\gamma_{\mathcal{Z}}(\rho) \equiv \text{Tr}(\rho_{\mathcal{Z}}^2) = \text{Tr}(\rho_{\mathcal{Q}}^2) \equiv \gamma_{\mathcal{Q}}(\rho). \quad (\text{C.4})$$

We now examine how the purity of the register $\mathcal{Z}$ evolves as a function of the momentum fraction z . As appears in Fig. C.2, the purity evaluated at the Chebyshev nodes (every second orange point) presents an alternating pattern of local maxima and minima. In particular, purity reaches its highest values at the boundaries $z = -1$ and $z = 1$, as well as at $z = 0$, indicating minimal entanglement. In contrast, the lowest purity values (signaling maximal entanglement) are observed around $z \approx 0.5$ and $z \approx 0.8$.

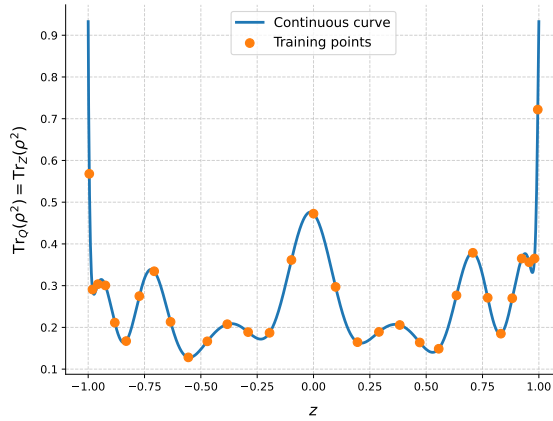


Figure C.2: Purity of the quantum register $\mathcal{Z}$ for different z values. Orange points represent the training points in Chebyshev nodes and halfpoints. Blue line represents a continuous curve in the range $[-1, 1]$.

C.3 Mutual information in quantum registers

In quantum information theory, quantum mutual information measures the total correlations, both classical and quantum, between two subsystems of a quantum state [17]. For a bipartite system represented by a density matrix $\rho_{\mathcal{Z}\mathcal{Q}}$, the quantum mutual information between subsystems $\mathcal{Z}$ and $\mathcal{Q}$ is expressed as:

$$I(\mathcal{Z} : \mathcal{Q}) = S(\rho_{\mathcal{Z}}) + S(\rho_{\mathcal{Q}}) - S(\rho_{\mathcal{Z}\mathcal{Q}}) \quad (\text{C.5})$$

where $S(\rho) = -\text{Tr}(\rho \log \rho)$ is the von Neumann entropy. A large mutual information indicates strong correlations between the two subsystems, suggesting significant entanglement. On the other hand, when $I(\mathcal{Z} : \mathcal{Q}) = 0$, it implies that the subsystems are entirely independent.

A notable simplification of Eq. C.5 occurs when the matrix $\rho_{\mathcal{Z}\mathcal{Q}}$ represents a pure state, as is the case here. In this scenario, $S(\rho_{\mathcal{Z}\mathcal{Q}}) = 0$ and $S(\rho_{\mathcal{Z}}) = S(\rho_{\mathcal{Q}})$. Therefore, Eq. C.5 is as follows:

$$I(\mathcal{Z} : \mathcal{Q}) = S(\rho_{\mathcal{Z}}) + S(\rho_{\mathcal{Q}}) = 2S(\rho_{\mathcal{Z}}) = -2\text{Tr}_{\mathcal{Z}}(\rho \log_2 \rho). \quad (\text{C.6})$$

In this section, we quantify the entanglement between the registers encoding z and Q during the training stage of Fig. 6.6(a), using their mutual information. Specifically, we calculate the quantum mutual information for the system $\mathcal{Z}$, consisting of $N = 5$ qubits, which depends on the variable z . As illustrated in Fig. C.3, the mutual information presents an alternating pattern of local maxima and minima at the Chebyshev nodes. The strongest correlations are observed at intermediate values of z , where the entanglement is most pronounced, while minimal correlations occur at the boundaries, -1 and 1 , as well as at $z = 0$.

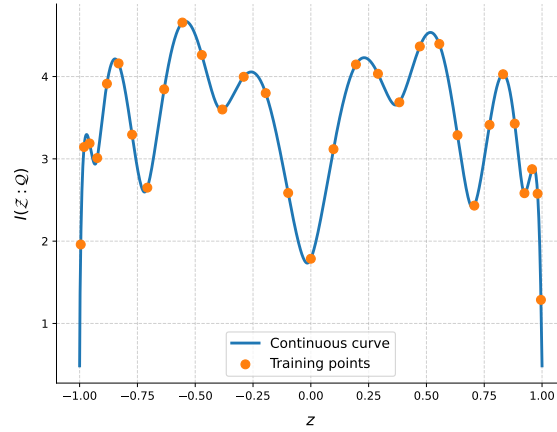


Figure C.3: Quantum mutual information between registers $\mathcal{Z}$ and $\mathcal{Q}$ for different z values. Orange points are training points in Chebyshev nodes and halfpoints. Blue line represents a continuous curve over the range $[-1, 1]$.

List of Figures

2.1	(a) Classification of the particles in the SM. Picture taken from [19]. (b) Interactions among particles in the SM. Picture taken from [20].	5
3.1	Bloch sphere representation of a qubit, showing quantum states as points on a unit sphere. Picture taken from [45].	14
3.2	Truth tables for fundamental logic gates: AND, OR, NOT, NAND, NOR, XOR, and XNOR, and Buffer illustrating input-output relationships for binary operations. Picture taken from [47].	16
3.3	Common quantum gates, including their names (with abbreviations), circuit representations, and corresponding unitary matrices. Picture taken from [50].	17
3.4	Quantum circuit that represents the maximally entangled state, the Bell state.	18
3.5	Quantum circuit <i>Swap Test</i>	20
3.6	Geometrical interpretation of the action of Grover operator G over the initial state $ \psi\rangle$. Picture taken from [17].	22
3.7	Quantum circuit representing Grover's algorithm for two qubits.	23
3.8	Quantum circuit measuring the expectation value of the observable $\hat{M}$	26
3.9	Graphical representation of Gradient Descent. The algorithm iteratively moves towards the minimum of the cost function by following the direction of the negative gradient. Picture taken from [62]	28
3.10	Venn diagram of the intersection between quantum information and Machine Learning (ML).	29
3.11	Quantum Machine Learning (QML) approaches by algorithm and data nature. Picture taken from [69].	29
3.12	Binary classification problem using a Support Vector Machine (SVM). Picture taken from [72].	32
3.13	A nonlinear embedding used to enhance the capabilities of a linear classifier. Linear classification in the embedding space translated to nonlinear decision boundaries in the original space. Picture taken from [72]	33
3.14	Quantum kernel circuit for states encoded by $S(x_i)$ and $S(x_j)$	34
3.15	Diagram of a classical Neural Network (NN) with three hidden layers. Picture taken from [80].	34
3.16	Generic Quantum Neural Network (QNN) structure. Picture taken from [84].	35
3.17	Quantum neural network following a data-reuploading scheme.	36
3.18	Quantum Circuit Born Machine (QCBM) workflow.	39
3.19	Quantum Generative Adversarial Network (QGAN) workflow.	40
4.1	Event with four reconstructed hadronic jets observed by the ATLAS detector at CERN. Picture taken from the ATLAS public event display repository [158]	43
4.2	Workflow of the K-means algorithm.	44

4.3	Workflow of the AP algorithm. Picture taken from [162]	46
4.4	Quantum circuit to obtain the invariant sum squared between two d -dimensional vectors in Minkowski space.	49
4.5	Quantum circuit for quantum maximum searching by qRAM amplitude encoding.	50
4.6	Clusters identified after five iterations by the classical and quantum versions of the K-means algorithm are shown in different colors. The dataset consists of Gaussian-distributed points generated with a random seed and a standard deviation of 2.0 from the cluster centroids. Clustering was performed using a Minkowski-type distance, assuming all particles are massless and on-shell, with both algorithms achieving an efficiency of $\varepsilon_t = 1.00$	52
4.7	Cluster efficiency of the K-means algorithm as a function of the data's standard deviation from the centroids. Both the classical and quantum versions were tested on 15 datasets with standard deviations ranging from 0.5 to 7.5.	52
4.8	A sample parton-level event generated as described in the text and clustered with the classical and the quantum version of the K-means++ algorithm, taking $K = 8$	53
4.9	A sample parton-level event, generated as described in the text, is clustered into $K = 2$ distinct clusters using both the classical and quantum versions of the AP algorithm.	55
4.10	A sample parton-level event generated as described in the text and clustered with three different k_T -jets algorithms as well as its quantum versions. . . .	56
5.1	Quantum circuit for amplitude amplification and estimation in Monte Carlo integration.	66
5.2	Quantum circuit Ansätze of the QNN. The Linear Ansatz (a) encodes each data feature in a single qubit. Thus the circuit depth grows linearly with the total number of variables M . The Parallel Ansatz (b) encodes the M variables in M qubits. In both Ansätze the circuit depth grows linearly with the number of layers L . Note that for $M = 1$, both the Linear and Parallel Ansatz converge to the same linear circuit.	68
5.3	Workflow of QFIAE. The input consists of the target function $f(\vec{x})$, the probability distribution $\rho(\vec{x})$, and the integration domain $\{\vec{x}_{min}, \vec{x}_{max}\}$. The QNN fits $f(\vec{x})$ and extracts its Fourier series from the quantum circuit. Next, IQAE estimates the integral for each trigonometric term in the Fourier series. Finally, these integrals are added with their corresponding coefficients to obtain the final integral result.	69
5.4	Quantum simulation using a Linear Ansatz in the QNN to fit the function $1 + x^2$. The setup uses $n_{\text{qubits,QF}} = 1$ and $L = 10$, resulting in a Fourier series with $n_{\text{Fourier}} = 10$ coefficients. A gradient descent method, the Adam optimizer, is employed with 200 data points in the range $[-1, 1]$. The optimizer hyperparameters are $learning_rate = 0.05$ and $nepochs = 100$. . .	71
5.5	Quantum circuits of the operators for the IQAE component of the QFIAE algorithm.	74
5.6	Tapdole Feynman diagram of an external particle with momentum p and an internal loop momentum ℓ with mass m	75

- 5.7 Quantum integration using QFIAE of the renormalized tadpole integral $\mathcal{A}_1^{(1,R)}(m; \mu_{UV})$ as a function of the mass m and the renormalization scale μ_{UV} (left) and comparison with the analytical result (right). The parameters used in the quantum implementation are: $max_steps = 300$, $step_size = 0.06$, $layers = n_{Fourier} = 5$ for the quantum Fourier part and $n_{qubits} = 5$, $n_{shots} = 10^3$, $\epsilon = 0.01$, $\alpha = 0.05$ for the IQAE part. 76
- 5.8 (Left) Quantum integration on Qibo and IBM Quantum hardware of the renormalized tadpole integral $\mathcal{A}_1^{(1,R)}(p, m; \mu_{UV})$ as a function of the mass m and the renormalization scale μ_{UV} . (Right) Deviations from the analytical value of the integral when the Fourier (F) decomposition or the IQAE integration (I) are performed in a classical (C) or a quantum (Q) computer. 77
- 5.9 Bubble Feynman diagram with external momentum p and internal loop momenta $q_1 = \ell$ and $q_2 = \ell + p$ with masses m_1 and m_2 , respectively. . . . 78
- 5.10 Quantum integration of the real (left) and imaginary (right) part of the renormalized bubble integral $\mathcal{A}_2^{(1,R)}(p, m, m; \mu_{UV})$ as a function of the ratio of the mass m to the energy component of the external momentum set at $p_0 = 100$ GeV, and the renormalization scale μ_{UV} . For the values $m/p_0 \geq 0.3$, the parameters used in the quantum implementation are: $max_steps = 200$, $step_size = 0.02$, whereas for $m/p_0 < 0.3$ more precision was needed, because of the threshold singularities, so the parameters used are: $max_steps = 450$, $step_size = 0.095$. For both cases, some parameters are equal: $layers = n_{Fourier} = 10$, $n_{qubits} = 1$ for the quantum Fourier part and $n_{qubits} = 5$, $n_{shots} = 10^3$, $\epsilon = 0.001$, $\alpha = 0.05$ for the IQAE part. 79
- 5.11 Quantum integration of the real (left) and imaginary (right) part of the renormalized tensor bubble integral $\mathcal{A}_{2,T}^{(1,R)}(p, m, m; \mu_{UV})$ as a function of the ratio of the mass m to the energy component of the external momentum set at $p_0 = 100$ GeV, and the renormalization scale μ_{UV} . The parameters used in the quantum implementation are: $max_steps = 5000$, $step_size = 0.0001$, $layers = n_{Fourier} = 8$, $n_{qubits} = 2$ for the quantum Fourier part and $n_{qubits} = 5$, $n_{shots} = 10^3$, $\epsilon = 0.001$, $\alpha = 0.05$ for the IQAE part. 80
- 5.12 Triangle Feynman loop diagram. The external particles, with momentum $\{p_i\}_{i=1,2,3}$, are considered massless. The internal loop momenta are $q_1 = \ell + p_1$, $q_2 = \ell + p_1 + p_2$, and $q_3 = \ell$ with masses m_1 , m_2 , m_3 respectively. . . 81
- 5.13 Quantum integration of the real (left) and imaginary (right) part of the triangle integral $\mathcal{A}_3^{(1)}(p_1, p_2, m, m, m)$ as a function of the ratio of the mass m to the cms energy set at $\sqrt{s} = 2$ GeV and comparison with the analytical result. For the values $m/\sqrt{s} \geq 0.5$, the parameters used in the quantum implementation are: $max_steps = 400$, $step_size = 0.05$, whereas for $m/\sqrt{s} \leq 0.5$ more precision was needed, because of the threshold singularity, so the parameters used are: $max_steps = 450$, $step_size = 0.065$. For both cases, some parameters are equal: $layers = n_{Fourier} = 10$, $n_{qubits} = 1$ for the QNN and $n_{qubits} = 5$, $n_{shots} = 10^4$, $\epsilon = 0.001$, $\alpha = 0.05$ for the IQAE part. 82

- 5.14 Quantum integration of the real (left) and imaginary (right) part of the renormalized tensor triangle integral $\mathcal{A}_{3,T}^{(1,R)}(p, m, m; \mu_{UV})$ as a function of the ratio of the mass m to the energy component of the external momentum set at $p_0 = 2$ GeV. The parameters used in the quantum implementation are: $max_steps = 15000$, $step_size = 0.001$, $layers = n_{Fourier} = 20$, $n_{qubits} = 6$ for the quantum Fourier part and $n_{qubits} = 5$, $n_{shots} = 10^3$, $\epsilon = 0.01$, $\alpha = 0.05$ for the IQAE part. 82
- 5.15 One-loop pentagon Feynman diagram with massless external momenta $\{p_i\}_{i=1}^5$ particle with momentum p_1, p_2, p_3, p_4, p_5 and massive internal loop momenta with masses $\{m_i\}_{i=1}^5$. $q_i = \{\ell + p_1, \ell + p_1 + p_2, \ell + p_1 + p_2 - p_3, \ell + p_1 + p_2 - p_3 - p_4, \ell\}$ 83
- 5.16 Three-loop vacuum diagrams contributing to the decay $\gamma^* \rightarrow q\bar{q}(g)$ at NLO. The gray dashed lines indicate phase-space residues, corresponding to different final states. Similar diagrams contribute to the decays $H \rightarrow q\bar{q}(g)$ and $\Phi \rightarrow \phi\phi(\phi)$, where the photon labeled 6 is replaced by a Higgs boson or a heavy scalar Φ . In the case of heavy scalar decay, particles 1 to 5 are substituted by light scalars. 86
- 5.17 Architecture of the QNN employed to fit a 2-dimensional function. 88
- 5.18 Quantum-integrated decay rates for the three processes $H \rightarrow q\bar{q}(g)$, $\gamma^* \rightarrow q\bar{q}(g)$, and $\Phi \rightarrow \phi\phi(\phi)$ at NLO as a function of the final state mass. The left panel shows results from a quantum simulator, while the right panel presents results where the IQAE has been executed on quantum hardware. The calculations use QFIAE with the LTD causal unitary approach, and the dashed lines represent theoretical predictions in dimensional regularization. Parameters for the quantum implementation: $max_steps = 15000$, $step_size = 0.001$, $layers = n_{Fourier} = 20$, $n_{qubits} = 6$ for the QNN, and $n_{qubits} = 5$, $n_{shots} = 10^3$, $\epsilon = 0.01$, $\alpha = 0.05$ for the IQAE module. 89
- 6.1 Quantum Chebyshev feature map that prepares a Chebyshev state via a sequence of phase feature maps that embed complex exponentials, a controlled rotation to adjust the normalization of the zero-frequency term, and post-selection on the ancilla measured in the $|0\rangle$ state. The scaled single-qubit phase shift gates in the feature map circuit are defined as $\tilde{P}_l^{[s]}(x) = \text{diag}\{1, \exp(is2^N \arccos(x)/l)\}$, where l increases exponentially as 2^j , with j indexing the qubit number, and $s \in \{1, -2\}$. Here, H and X denote the Hadamard and Pauli $\hat{X}$ gates, respectively. Picture adapted from [341]. 97
- 6.2 Quantum Chebyshev transform ($\hat{U}_{\text{QChT}}$) circuit, which maps computational basis states $\{|x_j\rangle\}_{j=0}^{2^N-1}$ into Chebyshev states $\{|\tau(x_j^{\text{Ch}})\rangle\}_{j=0}^{2^N-1}$. The transform consists of a quantum Fourier transform (QFT) applied to $N + 1$ qubits (an N -qubit system plus one ancillary qubit), followed by a sequence of phase-adjusting and permutation operations. We employ a standard phase shift gate defined as $P(\phi) = \text{diag}\{1, \exp(i\phi)\}$. Additionally, we use local phase rotations acting on the ancilla, given by $U_1 = P(-\pi/2^{N+1})R_Z(-\pi(2^N - 1)/2^{N+1})$ and $U_2 = P(-\pi/2)R_Y(-\pi/2)$, which can be combined into a single gate for implementation. Picture adapted from [341]. 98

- 6.3 Workflow of the QCPM. *Left*: Training circuit encodes x using a Chebyshev feature map, and includes a variational circuit whose parameters will be tuned in order to reproduce the behavior of the desired distribution. *Right*: Sampling circuit performs the inverse operations of the trained variational circuit, and the quantum Chebyshev transform, producing data that follows the distribution employed for training. 98
- 6.4 Workflow for describing fragmentation functions with quantum probabilistic models based on Chebyshev polynomials. (a) The input data for $D_i^h(z, Q)$ is produced for a grid of z and Q . (b) Quantum probabilistic model is composed of two Chebyshev feature maps for encoding z and Q , a correlation circuit that entangles both registers, and basis adaptation circuits to be trained on $D_i^h(z, Q)$. (c) For sampling we perform the inverse of basis adaptation, the correlation circuit, followed by parallel inverse quantum Chebyshev transforms for mapping the model into the bit basis. (d) Sampling results assembled in a 2-dimensional plot that represents $D_i^h(z, Q)$. 99
- 6.5 Generalized Quantum Chebyshev Probabilistic modeling. Application of shifted Chebyshev polynomials to the problem (a) and Chebyshev (b) spaces. (c) Comparisons of training and sampling grids for different extended registers. 100
- 6.6 (a) Quantum circuit used to train the multivariate distribution in the QCPM latent space, where a correlation circuit $\hat{C}$ is sandwiched between two identical sets of feature map circuits and variational Ansätze. Measured observable is defined as $\hat{O} = |0_a 0_a\rangle\langle 0_a 0_a|$, where $|\rangle \equiv |0\rangle^{\otimes N}$. Here, α and β are trainable scaling and bias parameters. (b) Quantum circuit used to sample the multivariate distribution from the trained model, where the inverse versions of the same parameterized circuits are applied with θ^* and ϑ^* being retrieved after the optimization procedure, followed by the inverse versions of the same correlation and two identical sets of quantum Chebyshev transform circuits associated with extended registers of S qubits ($|0\rangle^{\otimes S}$) in parallel, for fine sampling in the computational basis $|u_s u_a u_j v_s v_a v_j\rangle$. The quantum state prior to measurement is denoted as $|\psi\rangle$ 101
- 6.7 Hardware-Efficient Real-Amplitude (HERA) quantum circuit for $N = 4$ qubits with a circuit quantum depth of $d = 3$. The first block ($d = 1$) is framed by a dashed box. The HERA is composed of $N(d + 1)$ tunable single-qubit R_Y gates and Nd entangling (CNOT) gates. The training parameters are $\{\theta_i\}_{i=1}^{N(d+1)}$ 102
- 6.8 Sampling of $D_g^{K^\pm}(z, Q)$, the FF of a gluon fragmenting into kaons, with $S = 0, 1, 2$ additional qubits for each variable to interpolate in untrained regions. (a) Target distribution $D_g^{K^\pm}(z, Q)$. (b) Samples from trained QCPM of $D_g^{K^\pm}(z, Q)$ with the same number of qubits ($S = 0$). (c, d) Samples from $D_g^{K^\pm}(z, Q)$ with extended register ($S = 1, 2$). 103
- 6.9 (a) Accuracy (R^2) comparison of QCPMs for learning FFs $D_i^h(z, Q)$ with (w/ CC) and without (w/o CC) correlations between z and Q . (b) Nonpurity coefficient $C = |1 - \gamma_Z|$ of system Z for 100 training epochs with the fixed correlation circuit (solid curves, log-log curves). Dashed curves represent C when the system starts in a product state without entanglement between registers (no correlations). Solid curves highlight the values of nonpurity after training. 104

8.1	Comparación entre los flujos de trabajo de la integración Monte Carlo clásica y el algoritmo cuántico QFIAE. En QFIAE, se introducen como entrada la función objetivo, la distribución de probabilidad y el intervalo de integración. A continuación, una QNN ajusta la función y permite extraer su serie de Fourier. Posteriormente, se utilizan IQAE para calcular los integrales de cada componente trigonométrico en el intervalo especificado. Finalmente, se combinan estos resultados ponderados por sus coeficientes para obtener el valor final de la integral.	115
8.2	Funcionamiento del QCPM. <i>Izquierda:</i> El circuito de entrenamiento codifica x utilizando un mapa de características de Chebyshev, e incluye un circuito variacional cuyos parámetros se ajustarán para reproducir el comportamiento de la distribución objetivo. <i>Derecha:</i> El circuito de muestreo realiza las operaciones inversas del circuito variacional entrenado y de la transformada cuántica de Chebyshev, generando datos que siguen la distribución utilizada durante el entrenamiento.	118
8.3	Proceso de clusterización de jets de distintos algoritmos cuánticos.	119
8.4	Integración cuántica de la parte real (izquierda) e imaginaria (derecha) de la integral de burbuja renormalizada $\mathcal{A}^{(1,R)}_2(p, m, m; \mu_{UV})$ como función del cociente entre la masa m y la componente de energía del momento externo, fijado en $p_0 = 100$ GeV, y de la escala de renormalización μ_{UV}	122
8.5	Integración cuántica de la parte real (izquierda) e imaginaria (derecha) de la integral del triángulo $\mathcal{A}_3^{(1)}(p_1, p_2, m, m, m)$ como función del cociente entre la masa m y la energía del centro de masas fijada en $\sqrt{s} = 2$ GeV.	123
8.6	Tasas de desintegración integradas cuánticamente para los tres procesos $H \rightarrow q\bar{q}(g)$, $\gamma^* \rightarrow q\bar{q}(g)$ y $\Phi \rightarrow \phi\phi(\phi)$ a segundo orden en teoría de perturbaciones, en función de la masa del estado final. El panel izquierdo muestra resultados obtenidos con un simulador cuántico, mientras que el derecho presenta resultados en los que el algoritmo IQAE se ejecutó en hardware cuántico. Los cálculos emplean QFIAE con el enfoque causal unitario de LTD, y las líneas discontinuas representan las predicciones teóricas en regularización dimensional.	124
8.7	Muestreo de $D_g^{K^\pm}(z, Q)$, la FF de un gluón fragmentando en kaones, con $S = 0, 1, 2$ qubits adicionales por variable para interpolar en regiones no entrenadas. (a) Distribución objetivo $D_g^{K^\pm}(z, Q)$. (b) Muestreo del modelo QCPM entrenado de $D_g^{K^\pm}(z, Q)$ con el mismo número de qubits ($S = 0$). (c,d) Muestreo de $D_g^{K^\pm}(z, Q)$ con registro extendido ($S = 1, 2$).	125
A.1	Quantum circuit of the <i>SwapTest</i>	127
B.1	Sampling of fragmentation functions of partons into kaons K^+ and K^- . . .	130
B.2	Sampling of fragmentation functions of partons into pions π^+ and π^- . . .	131
C.1	Von Neumann entropies of the circuits in z registers (top) and Q registers (bottom) with (w/ CC) and without (w/o CC) correlations between z and Q . . .	133
C.2	Purity of the quantum register $\mathcal{Z}$ for different z values. Orange points represent the training points in Chebyshev nodes and halfpoints. Blue line represents a continuous curve in the range $[-1, 1]$	134
C.3	Quantum mutual information between registers $\mathcal{Z}$ and Q for different z values. Orange points are training points in Chebyshev nodes and halfpoints. Blue line represents a continuous curve over the range $[-1, 1]$	135

List of Tables

4.1	Efficiencies of AP algorithms for Gaussian datasets with different number of clusters.	54
4.2	Efficiencies and number of <i>shots</i> of the quantum k_T -jet algorithms as a function of the parameter a	57
4.3	Time complexity of quantum and classical jet clustering algorithms when one or both subroutines are applied.	57
5.1	Integration of $1+x^2$ from $[0, 1]$ using QFIAE and FQMCI. With $\epsilon = I_{est}/I_{exact}$	72
5.2	Depths of the Quantum Fourier (QF) and Iterative Quantum Amplitude Estimation (IQAE) parts of the QFIAE method.	72
5.3	Quantum integration of a pentagon configuration using the FQMCI method, and comparison with a classical numerical integration [255]. With $n_{Fourier} = 10$, $n_{qubits} = 5$, $n_{shots} = 10^4$, $\epsilon = 0.001$, $\alpha = 0.05$. The uncertainty for the quantum integration has been calculated as in previous sections.	84
5.4	Quantum-integrated decay rates at NLO for the processes $H \rightarrow q\bar{q}(g)$, $\gamma^* \rightarrow q\bar{q}(g)$, and $\Phi \rightarrow \phi\phi(\phi)$ as a function of the final state mass, computed using QFIAE and LTD causal unitary. The ‘Hardware’ column reports results from the QNN on a quantum simulator and the IQAE on quantum hardware, while the ‘Simulator’ column shows results where both the QNN and IQAE run on quantum simulators. The ‘DREG’ column provides the exact analytic results at NLO accuracy.	89
8.1	Eficiencias de los algoritmos de clusterización de jet al compararse con los resultados que obtienen sus análogos clásicos.	119
8.2	Complejidad temporal de los algoritmos clásicos y cuánticos de clusterización de jets cuando se aplica una o ambas subrutinas cuánticas.	120
8.3	Integral renormalizada del tipo renacuajo $\mathcal{A}_1^{(1,R)}(m; \mu_{UV})$ evaluada en Qibo e IBM Quantum, como función del cociente μ_{UV}/m , donde se fija $m = 5$ GeV. La descomposición de Fourier y la integración mediante IQAE se realizan ya sea con un simulador cuántico (S) o en un dispositivo cuántico real (Q). En la primera fila ambas componentes se realizan con métodos clásicos (C).	121

Bibliography

- [1] Jorge J. Martínez de Lejarza et al. “Quantum Chebyshev Probabilistic Models for Fragmentation Functions”. In: (Mar. 2025).
(Cited on pp: xiii, 93, 105).
arXiv: [2503.16073 \[quant-ph\]](#).
- [2] Jorge J. Martínez de Lejarza et al. “Quantum integration of decay rates at second order in perturbation theory”. In: *Quantum Sci. Technol.* 10.2 (2025), p. 025026.
(Cited on pp: xiii, 41, 60, 90, 93).
DOI: [10.1088/2058-9565/ada9c5](#).
arXiv: [2409.12236 \[quant-ph\]](#).
- [3] Jorge J. Martínez de Lejarza et al. “Loop Feynman integration on a quantum computer”. In: *Phys. Rev. D* 110.7 (2024), p. 074031.
(Cited on pp: xiii, 41, 60, 90, 93).
DOI: [10.1103/PhysRevD.110.074031](#).
arXiv: [2401.03023 \[hep-ph\]](#).
- [4] Selomit Ramírez-Urbe et al. “Vacuum amplitudes and time-like causal unitary in the loop-tree duality”. In: *JHEP* 01 (2025), p. 103.
(Cited on pp: xiii, 84, 85).
DOI: [10.1007/JHEP01\(2025\)103](#).
arXiv: [2404.05492 \[hep-ph\]](#).
- [5] Jorge J. Martínez de Lejarza et al. “Quantum Fourier Iterative Amplitude Estimation”. In: *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 01. 2023, pp. 571–579.
(Cited on pp: xiii, 60, 61, 63, 90, 93, 113, 115).
DOI: [10.1109/QCE57702.2023.00071](#).
- [6] Jorge J. Martínez de Lejarza, Leandro Cieri, and Germán Rodrigo. “Quantum clustering and jet reconstruction at the LHC”. In: *Phys. Rev. D* 106.3 (2022), p. 036021.
(Cited on pp: xiii, 41, 57, 60).
DOI: [10.1103/PhysRevD.106.036021](#).
arXiv: [2204.06496 \[hep-ph\]](#).
- [7] Andrea Delgado et al. “Quantum Computing for Data Analysis in High-Energy Physics”. In: *2022 Snowmass Summer Study*. Mar. 2022.
(Cited on pp: xiii, 41, 60, 93).
arXiv: [2203.08805 \[physics.data-an\]](#).

- [8] Jorge J. Martínez de Lejarza, Leandro Cieri, and Germán Rodrigo. “Quantum jet clustering with LHC simulated data”. In: *PoS ICHEP2022* (2022), p. 241.
(Cited on pp: xiii, 41, 57, 60, 93).
DOI: [10.22323/1.414.0241](https://doi.org/10.22323/1.414.0241).
arXiv: [2209.08914](https://arxiv.org/abs/2209.08914) [hep-ph].
- [9] Konstantinos Pyretzidis, Jorge J. Martínez de Lejarza, and Germán Rodrigo. “Unlocking Multi-Dimensional Integration with Quantum Adaptive Importance Sampling”. In: (June 2025).
(Cited on pp: xiii, 41, 93).
arXiv: [2506.19965](https://arxiv.org/abs/2506.19965) [quant-ph].
- [10] Michael Nauenberg. “Max Planck and the birth of the quantum hypothesis”. In: *American Journal of Physics* 84 (Sept. 2016), pp. 709–720.
(Cited on p: 1).
DOI: [10.1119/1.4955146](https://doi.org/10.1119/1.4955146).
- [11] W. a Heisenberg. “Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik”. In: *Z. Phys.* 43 (1927), pp. 172–198.
(Cited on p: 1).
DOI: [10.1007/BF01397280](https://doi.org/10.1007/BF01397280).
- [12] A. Einstein. “Zur Elektrodynamik bewegter Körper”. In: *Annalen Phys.* 322.10 (1905), pp. 891–921.
(Cited on p: 2).
DOI: [10.1002/andp.19053221004](https://doi.org/10.1002/andp.19053221004).
- [13] Salvador Lucas Alba. *La máquina que cambió el mundo: Génesis, desarrollo y evolución del ordenador*. en. 2019.
(Cited on p: 2).
- [14] B. Jack Copeland. “The Church-Turing Thesis”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta and Uri Nodelman. Winter 2024. Metaphysics Research Lab, Stanford University, 2024.
(Cited on p: 2).
- [15] Liesbeth De Mol. “Turing Machines”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta and Uri Nodelman. Winter 2024. Metaphysics Research Lab, Stanford University, 2024.
(Cited on p: 2).
- [16] Richard P. Feynman. “Simulating physics with computers”. In: *Int. J. Theor. Phys.* 21 (1982). Ed. by L. M. Brown, pp. 467–488.
(Cited on p: 2).
DOI: [10.1007/BF02650179](https://doi.org/10.1007/BF02650179).
- [17] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2011. ISBN: 9781107002173.
(Cited on pp: 3, 22, 132, 134).
URL: <https://www.amazon.com/Quantum-Computation-Information-10th-Anniversary/dp/1107002176?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1107002176>.

- [18] Alberto Peruzzo et al. “A variational eigenvalue solver on a photonic quantum processor”. In: *Nature Commun.* 5.1 (2014), p. 4213.
(Cited on p: 4).
DOI: [10.1038/ncomms5213](https://doi.org/10.1038/ncomms5213).
arXiv: [1304.3061](https://arxiv.org/abs/1304.3061) [quant-ph].
- [19] Wikipedia. *Quantum chromodynamics - Wikipedia — en.wikipedia.org*. https://en.wikipedia.org/wiki/Quantum_chromodynamics. [Accessed 24-04-2025].
(Cited on p: 5).
- [20] Wikipedia. *Archivo:Elementary-particle-interactions-es.svg - Wikipedia, la enciclopedia libre — es.m.wikipedia.org*. <https://es.m.wikipedia.org/wiki/Archivo:Elementary-particle-interactions-es.svg>. [Accessed 24-04-2025].
(Cited on p: 5).
- [21] S. F. Novaes. “Standard model: An Introduction”. In: *10th Jorge Andre Swieca Summer School: Particle and Fields*. Jan. 1999, pp. 5–102.
(Cited on p: 6).
arXiv: [hep-ph/0001283](https://arxiv.org/abs/hep-ph/0001283).
- [22] J. D. Bjorken and S. L. Glashow. “Elementary Particles and SU(4)”. In: *Phys. Lett.* 11 (1964), pp. 255–257.
(Cited on p: 6).
DOI: [10.1016/0031-9163\(64\)90433-0](https://doi.org/10.1016/0031-9163(64)90433-0).
- [23] Martin L. Perl et al. “Evidence for Anomalous Lepton Production in $e^+ - e^-$ Annihilation”. In: *Phys. Rev. Lett.* 35 (1975), pp. 1489–1492.
(Cited on p: 6).
DOI: [10.1103/PhysRevLett.35.1489](https://doi.org/10.1103/PhysRevLett.35.1489).
- [24] G. Arnison et al. “Experimental Observation of Isolated Large Transverse Energy Electrons with Associated Missing Energy at $\sqrt{s} = 540$ GeV”. In: *Phys. Lett. B* 122 (1983), pp. 103–116.
(Cited on p: 6).
DOI: [10.1016/0370-2693\(83\)91177-2](https://doi.org/10.1016/0370-2693(83)91177-2).
- [25] G. Arnison et al. “Experimental Observation of Lepton Pairs of Invariant Mass Around 95-GeV/c² at the CERN SPS Collider”. In: *Phys. Lett. B* 126 (1983), pp. 398–410.
(Cited on p: 6).
DOI: [10.1016/0370-2693\(83\)90188-0](https://doi.org/10.1016/0370-2693(83)90188-0).
- [26] F. Abe et al. “Observation of top quark production in $\bar{p}p$ collisions”. In: *Phys. Rev. Lett.* 74 (1995), pp. 2626–2631.
(Cited on p: 6).
DOI: [10.1103/PhysRevLett.74.2626](https://doi.org/10.1103/PhysRevLett.74.2626).
arXiv: [hep-ex/9503002](https://arxiv.org/abs/hep-ex/9503002).
- [27] Georges Aad et al. “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC”. In: *Phys. Lett. B* 716 (2012), pp. 1–29.
(Cited on p: 6).
DOI: [10.1016/j.physletb.2012.08.020](https://doi.org/10.1016/j.physletb.2012.08.020).
arXiv: [1207.7214](https://arxiv.org/abs/1207.7214) [hep-ex].

- [28] Serguei Chatrchyan et al. “Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC”. In: *Phys. Lett. B* 716 (2012), pp. 30–61.
(Cited on p: 6).
DOI: [10.1016/j.physletb.2012.08.021](https://doi.org/10.1016/j.physletb.2012.08.021).
arXiv: [1207.7235 \[hep-ex\]](https://arxiv.org/abs/1207.7235).
- [29] Antonio Pich. “The Standard Model of Electroweak Interactions”. In: *2010 European School of High Energy Physics*. Jan. 2012, pp. 1–50.
(Cited on p: 6).
arXiv: [1201.0537 \[hep-ph\]](https://arxiv.org/abs/1201.0537).
- [30] Claudia Hagedorn. “Chapter 11: Physics beyond the Standard Model”. Electroweak course. Masters degree in Advanced Physics, Universitat de València (2021).
(Cited on p: 6).
- [31] Peter W. Higgs. “Broken Symmetries and the Masses of Gauge Bosons”. In: *Phys. Rev. Lett.* 13 (16 1964), pp. 508–509.
(Cited on p: 10).
DOI: [10.1103/PhysRevLett.13.508](https://doi.org/10.1103/PhysRevLett.13.508).
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.13.508>.
- [32] F. Englert and R. Brout. “Broken Symmetry and the Mass of Gauge Vector Mesons”. In: *Phys. Rev. Lett.* 13 (9 1964), pp. 321–323.
(Cited on p: 10).
DOI: [10.1103/PhysRevLett.13.321](https://doi.org/10.1103/PhysRevLett.13.321).
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.13.321>.
- [33] J. Goldstone. “Field Theories with Superconductor Solutions”. In: *Nuovo Cim.* 19 (1961), pp. 154–164.
(Cited on p: 10).
DOI: [10.1007/BF02812722](https://doi.org/10.1007/BF02812722).
- [34] A. Abada et al. “FCC Physics Opportunities: Future Circular Collider Conceptual Design Report Volume 1”. In: *Eur. Phys. J. C* 79.6 (2019), p. 474.
(Cited on pp: 11, 41).
DOI: [10.1140/epjc/s10052-019-6904-3](https://doi.org/10.1140/epjc/s10052-019-6904-3).
- [35] A. Abada et al. “FCC-ee: The Lepton Collider: Future Circular Collider Conceptual Design Report Volume 2”. In: *Eur. Phys. J. ST* 228.2 (2019), pp. 261–623.
(Cited on p: 11).
DOI: [10.1140/epjst/e2019-900045-4](https://doi.org/10.1140/epjst/e2019-900045-4).
- [36] A. Abada et al. “FCC-hh: The Hadron Collider: Future Circular Collider Conceptual Design Report Volume 3”. In: *Eur. Phys. J. ST* 228.4 (2019), pp. 755–1107.
(Cited on p: 11).
DOI: [10.1140/epjst/e2019-900087-0](https://doi.org/10.1140/epjst/e2019-900087-0).
- [37] A. Abada et al. “HE-LHC: The High-Energy Large Hadron Collider: Future Circular Collider Conceptual Design Report Volume 4”. In: *Eur. Phys. J. ST* 228.5 (2019), pp. 1109–1382.
(Cited on p: 11).
DOI: [10.1140/epjst/e2019-900088-6](https://doi.org/10.1140/epjst/e2019-900088-6).

- [38] Philip Bambade et al. “The International Linear Collider: A Global Project”. In: (Mar. 2019).
(Cited on p: 11).
arXiv: [1903.01629 \[hep-ex\]](#).
- [39] “The Compact Linear e^+e^- Collider (CLIC): Physics Potential”. In: (Dec. 2018).
Ed. by P. Roloff et al.
(Cited on pp: 11, 41).
arXiv: [1812.07986 \[hep-ex\]](#).
- [40] “CEPC Conceptual Design Report: Volume 1 - Accelerator”. In: (Sept. 2018).
(Cited on p: 11).
arXiv: [1809.00285 \[physics.acc-ph\]](#).
- [41] Mingyi Dong et al. “CEPC Conceptual Design Report: Volume 2 - Physics & Detector”. In: (Nov. 2018). Ed. by João Barreiro Guimarães da Costa et al.
(Cited on pp: 11, 41).
arXiv: [1811.10545 \[hep-ex\]](#).
- [42] CERN. *Facts and figures about the LHC*. Accessed: 2024-11-11.
(Cited on pp: 11, 93).
URL: <https://home.cern/resources/faqs/facts-and-figures-about-lhc>.
- [43] Kim Albertsson et al. “Machine Learning in High Energy Physics Community White Paper”. In: *J. Phys. Conf. Ser.* 1085.2 (2018), p. 022008.
(Cited on p: 11).
DOI: [10.1088/1742-6596/1085/2/022008](#).
arXiv: [1807.02876 \[physics.comp-ph\]](#).
- [44] IBM. *What Is Quantum Computing? | IBM — ibm.com*. <https://www.ibm.com/think/topics/quantum-computing>. [Accessed 02-01-2025]. 2024.
(Cited on p: 13).
- [45] Wikipedia. *Bloch sphere - Wikipedia — en.wikipedia.org*. https://en.wikipedia.org/wiki/Bloch_sphere. [Accessed 24-04-2025].
(Cited on p: 14).
- [46] *Logic Gate (AND, OR, XOR, NOT, NAND, NOR and XNOR) - GeeksforGeeks — geeksforgeeks.org*. <https://www.geeksforgeeks.org/logic-gates/>. [Accessed 03-01-2025].
(Cited on p: 15).
- [47] Cody Wabiszewski. *Digital Logic Gates | Using Individual Transistors — gsnetwork.com*. <https://www.gsnetwork.com/digital-logic-gates/>. [Accessed 24-04-2025].
(Cited on p: 16).
- [48] Peter W. Shor. “Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer”. In: *SIAM J. Sci. Statist. Comput.* 26 (1997), p. 1484.
(Cited on pp: 16, 17).
DOI: [10.1137/S0097539795293172](#).
arXiv: [quant-ph/9508027](#).

- [49] M. A. Nielsen et al. “Information theoretic approach to quantum error correction and reversible measurement”. In: *Proc. Roy. Soc. Lond. A* 454 (1998), p. 277.
(Cited on p: 16).
DOI: [10.1098/rspa.1998.0160](https://doi.org/10.1098/rspa.1998.0160).
arXiv: [quant-ph/9706064](https://arxiv.org/abs/quant-ph/9706064).
- [50] Wikipedia. *Archivo:Quantum Logic Gates.png - Wikipedia, la enciclopedia libre — es.m.wikipedia.org*. https://es.m.wikipedia.org/wiki/Archivo:Quantum_Logic_Gates.png. [Accessed 24-04-2025].
(Cited on p: 17).
- [51] Lov K. Grover. “A Fast quantum mechanical algorithm for database search”. In: (May 1996).
(Cited on pp: 17, 20, 96, 117).
arXiv: [quant-ph/9605043](https://arxiv.org/abs/quant-ph/9605043).
- [52] Harry Buhrman et al. “Quantum Fingerprinting”. In: *Phys. Rev. Lett.* 87.16 (2001), p. 167902.
(Cited on p: 19).
DOI: [10.1103/PhysRevLett.87.167902](https://doi.org/10.1103/PhysRevLett.87.167902).
arXiv: [quant-ph/0102001](https://arxiv.org/abs/quant-ph/0102001).
- [53] Christoph Dürr and Peter Høyer. “A Quantum algorithm for finding the minimum”. In: (July 1996).
(Cited on pp: 23, 50, 51).
arXiv: [quant-ph/9607014](https://arxiv.org/abs/quant-ph/9607014).
- [54] Guillermo Alonso-Linaje. *Basic arithmetic with the quantum Fourier transform (QFT) / PennyLane Demos — pennylane.ai*. https://pennylane.ai/qml/demos/tutorial_qft_arithmetics. [Accessed 14-01-2025]. 2022.
(Cited on p: 23).
- [55] Ashley Montanaro. “Quantum speedup of Monte Carlo methods”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 471.2181 (Sept. 2015), p. 20150301. ISSN: 1471-2946.
(Cited on p: 23).
DOI: [10.1098/rspa.2015.0301](https://doi.org/10.1098/rspa.2015.0301).
URL: <http://dx.doi.org/10.1098/rspa.2015.0301>.
- [56] Gilles Brassard et al. “Quantum amplitude amplification and estimation”. In: (May 2000).
(Cited on p: 23).
DOI: [10.1090/conm/305/05215](https://doi.org/10.1090/conm/305/05215).
arXiv: [quant-ph/0005055](https://arxiv.org/abs/quant-ph/0005055).
- [57] John Preskill. “Quantum Computing in the NISQ era and beyond”. In: *Quantum* 2 (2018), p. 79.
(Cited on p: 24).
DOI: [10.22331/q-2018-08-06-79](https://doi.org/10.22331/q-2018-08-06-79).
arXiv: [1801.00862](https://arxiv.org/abs/1801.00862) [quant-ph].
- [58] Juan Miguel Arrazola. *From NISQ to ISQ / PennyLane Blog — pennylane.ai*. <https://pennylane.ai/blog/2023/06/from-nisq-to-isq>. [Accessed 14-01-2025]. 2023.
(Cited on p: 24).

- [59] M. Cerezo et al. “Variational quantum algorithms”. In: *Nature Rev. Phys.* 3.9 (2021), pp. 625–644.
(Cited on p: 25).
DOI: [10.1038/s42254-021-00348-9](https://doi.org/10.1038/s42254-021-00348-9).
arXiv: [2012.09265](https://arxiv.org/abs/2012.09265) [quant-ph].
- [60] Maria Schuld et al. “Evaluating analytic gradients on quantum hardware”. In: *Phys. Rev. A* 99.3 (2019), p. 032331.
(Cited on p: 26).
DOI: [10.1103/PhysRevA.99.032331](https://doi.org/10.1103/PhysRevA.99.032331).
arXiv: [1811.11184](https://arxiv.org/abs/1811.11184) [quant-ph].
- [61] Ville Bergholm et al. “PennyLane: Automatic differentiation of hybrid quantum-classical computations”. In: (Nov. 2018).
(Cited on pp: 27, 73, 87, 122).
arXiv: [1811.04968](https://arxiv.org/abs/1811.04968) [quant-ph].
- [62] Josh Izaac. *Doubly stochastic gradient descent / PennyLane Demos* — [pennylane.ai. https://pennylane.ai/qml/demos/tutorial_doubly_stochastic](https://pennylane.ai/qml/demos/tutorial_doubly_stochastic). [Accessed 24-04-2025].
(Cited on p: 28).
- [63] Juan Carrasquilla and Roger G. Melko. “Machine learning phases of matter”. In: *Nature Physics* 13.5 (Feb. 2017), 431–434. ISSN: 1745-2481.
(Cited on p: 29).
DOI: [10.1038/nphys4035](https://doi.org/10.1038/nphys4035).
URL: <http://dx.doi.org/10.1038/nphys4035>.
- [64] Alexander Hentschel and Barry C. Sanders. “Machine Learning for Precise Quantum Measurement”. In: *Phys. Rev. Lett.* 104 (6 2010), p. 063603.
(Cited on p: 29).
DOI: [10.1103/PhysRevLett.104.063603](https://doi.org/10.1103/PhysRevLett.104.063603).
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.104.063603>.
- [65] Jacob Biamonte et al. “Quantum machine learning”. In: *Nature* 549.7671 (Sept. 2017), 195–202. ISSN: 1476-4687.
(Cited on p: 29).
DOI: [10.1038/nature23474](https://doi.org/10.1038/nature23474).
URL: <http://dx.doi.org/10.1038/nature23474>.
- [66] E. Miles Stoudenmire and David J. Schwab. *Supervised Learning with Quantum-Inspired Tensor Networks*. 2017.
(Cited on p: 29).
arXiv: [1605.05775](https://arxiv.org/abs/1605.05775) [stat.ML].
URL: <https://arxiv.org/abs/1605.05775>.
- [67] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. “An introduction to quantum machine learning”. In: *Contemporary Physics* 56.2 (Oct. 2014), 172–185. ISSN: 1366-5812.
(Cited on p: 29).
DOI: [10.1080/00107514.2014.964942](https://doi.org/10.1080/00107514.2014.964942).
URL: <http://dx.doi.org/10.1080/00107514.2014.964942>.

- [68] Maria Schuld. “Quantum machine learning for supervised pattern recognition.” In: 2017.
(Cited on pp: 29, 30).
URL: <https://api.semanticscholar.org/CorpusID:69768800>.
- [69] Wikipedia. *Quantum machine learning - Wikipedia* — *en.wikipedia.org*. https://en.wikipedia.org/wiki/Quantum_machine_learning. [Accessed 24-04-2025].
(Cited on p: 29).
- [70] Thomas Hubregtsen et al. “Training quantum embedding kernels on near-term quantum computers”. In: *Phys. Rev. A* 106.4 (2022), p. 042431.
(Cited on p: 32).
DOI: [10.1103/PhysRevA.106.042431](https://doi.org/10.1103/PhysRevA.106.042431).
arXiv: [2105.02276](https://arxiv.org/abs/2105.02276) [quant-ph].
- [71] Corinna Cortes and Vladimir Naumovich Vapnik. “Support-Vector Networks”. In: *Machine Learning* 20 (1995), pp. 273–297.
(Cited on p: 32).
URL: <https://api.semanticscholar.org/CorpusID:52874011>.
- [72] Peter-Jan Derks et al. *Training and evaluating quantum kernels / PennyLane Demos* — *pennylane.ai*. https://pennylane.ai/qml/demos/tutorial_kernels_module. [Accessed 24-04-2025].
(Cited on pp: 32, 33).
- [73] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. “Quantum Support Vector Machine for Big Data Classification”. In: *Phys. Rev. Lett.* 113.13 (2014), p. 130503.
(Cited on p: 33).
DOI: [10.1103/physrevlett.113.130503](https://doi.org/10.1103/physrevlett.113.130503).
arXiv: [1307.0471](https://arxiv.org/abs/1307.0471) [quant-ph].
- [74] M. ANDRECUT and M. K. ALI. “A QUANTUM NEURAL NETWORK MODEL”. In: *International Journal of Modern Physics C* 13.01 (2002), pp. 75–88.
(Cited on p: 34).
DOI: [10.1142/S0129183102002948](https://doi.org/10.1142/S0129183102002948).
eprint: <https://doi.org/10.1142/S0129183102002948>.
URL: <https://doi.org/10.1142/S0129183102002948>.
- [75] Massimo Panella and Giuseppe Martinelli. “Neural networks with quantum architecture and quantum learning”. In: *Int. J. Circuit Theory Appl.* 39.1 (2011), pp. 61–77.
(Cited on p: 34).
DOI: [10.1002/CTA.619](https://doi.org/10.1002/CTA.619).
URL: <https://doi.org/10.1002/cta.619>.
- [76] Yudong Cao, Gian Giacomo Guerreschi, and Alán Aspuru-Guzik. “Quantum Neuron: an elementary building block for machine learning on quantum computers”. In: (Nov. 2017).
(Cited on p: 34).
arXiv: [1711.11240](https://arxiv.org/abs/1711.11240) [quant-ph].
- [77] Edward Farhi and Hartmut Neven. “Classification with Quantum Neural Networks on Near Term Processors”. In: *arXiv: Quantum Physics* (2018).
(Cited on p: 34).
URL: <https://api.semanticscholar.org/CorpusID:119037649>.

- [78] Francesco Tacchino et al. “Quantum implementation of an artificial feed-forward neural network”. In: *Quantum Science and Technology* 5.4 (2020), p. 044010. (Cited on p: 34).
DOI: [10.1088/2058-9565/abb8e4](https://doi.org/10.1088/2058-9565/abb8e4).
URL: <https://dx.doi.org/10.1088/2058-9565/abb8e4>.
- [79] Elsy Gómez-Ramos and Francisco Venegas-Martínez. “A Review of Artificial Neural Networks: How Well Do They Perform in Forecasting Time Series?” In: 2013. (Cited on p: 34).
URL: <https://api.semanticscholar.org/CorpusID:58795942>.
- [80] Michael A. Nielsen. *Neural Networks and Deep Learning*. misc. 2018. (Cited on p: 34).
URL: <http://neuralnetworksanddeeplearning.com/>.
- [81] Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural Networks* 61 (2015), pp. 85–117. ISSN: 0893-6080. (Cited on p: 34).
DOI: <https://doi.org/10.1016/j.neunet.2014.09.003>.
URL: <https://www.sciencedirect.com/science/article/pii/S0893608014002135>.
- [82] Raúl Rojas. “The Backpropagation Algorithm”. In: 1996. (Cited on p: 35).
URL: <https://api.semanticscholar.org/CorpusID:62185690>.
- [83] Keras. *Keras documentation: The Sequential model*. https://keras.io/guides/sequential_model/. [Accessed 05-02-2025]. 2023. (Cited on p: 35).
- [84] Ali Javadi-Abhari et al. *Quantum computing with Qiskit*. 2024. (Cited on pp: 35, 51, 88).
DOI: [10.48550/arXiv.2405.08810](https://doi.org/10.48550/arXiv.2405.08810).
arXiv: [2405.08810](https://arxiv.org/abs/2405.08810) [quant-ph].
- [85] Adrián Pérez-Salinas et al. “Data re-uploading for a universal quantum classifier”. In: *Quantum* 4 (Feb. 2020), p. 226. ISSN: 2521-327X. (Cited on pp: 35, 68, 115).
DOI: [10.22331/q-2020-02-06-226](https://doi.org/10.22331/q-2020-02-06-226).
URL: <http://dx.doi.org/10.22331/q-2020-02-06-226>.
- [86] Adrián Pérez-Salinas. “Algorithmic strategies for seizing quantum computing”. PhD thesis. Barcelona, Autonomia U., Barcelona U., 2021. (Cited on p: 36).
arXiv: [2112.15175](https://arxiv.org/abs/2112.15175) [quant-ph].
- [87] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. “Effect of data encoding on the expressive power of variational quantum-machine-learning models”. In: *Phys. Rev. A* 103.3 (2021), p. 032430. (Cited on pp: 36, 69).
DOI: [10.1103/PhysRevA.103.032430](https://doi.org/10.1103/PhysRevA.103.032430).
arXiv: [2008.08605](https://arxiv.org/abs/2008.08605) [quant-ph].

- [88] Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. “Structure optimization for parameterized quantum circuits”. In: *Quantum* 5 (2021), p. 391. (Cited on pp: 36, 69).
DOI: [10.22331/q-2021-01-28-391](https://doi.org/10.22331/q-2021-01-28-391).
arXiv: [1905.09692](https://arxiv.org/abs/1905.09692) [quant-ph].
- [89] Sandeep Singh Sengar et al. *Generative Artificial Intelligence: A Systematic Review and Applications*. 2024. (Cited on p: 38).
arXiv: [2405.11029](https://arxiv.org/abs/2405.11029) [cs.LG].
URL: <https://arxiv.org/abs/2405.11029>.
- [90] Sam Bond-Taylor et al. “Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.11 (Nov. 2022), 7327–7347. ISSN: 1939-3539. (Cited on p: 38).
DOI: [10.1109/tpami.2021.3116668](https://doi.org/10.1109/tpami.2021.3116668).
URL: <http://dx.doi.org/10.1109/TPAMI.2021.3116668>.
- [91] Carlos A. Riofrío et al. “A Characterization of Quantum Generative Models”. In: *ACM Trans. Quant. Comput.* 5.2 (2024), p. 12. (Cited on p: 38).
DOI: [10.1145/3655027](https://doi.org/10.1145/3655027).
arXiv: [2301.09363](https://arxiv.org/abs/2301.09363) [quant-ph].
- [92] Christa Zoufal. “Generative Quantum Machine Learning”. PhD thesis. Zurich, ETH, 2021. (Cited on p: 38).
DOI: [10.3929/ethz-b-000514692](https://doi.org/10.3929/ethz-b-000514692).
arXiv: [2111.12738](https://arxiv.org/abs/2111.12738) [quant-ph].
- [93] Frank Arute et al. “Quantum supremacy using a programmable superconducting processor”. In: *Nature* 574.7779 (2019), pp. 505–510. (Cited on p: 38).
DOI: [10.1038/s41586-019-1666-5](https://doi.org/10.1038/s41586-019-1666-5).
arXiv: [1910.11333](https://arxiv.org/abs/1910.11333) [quant-ph].
- [94] *Quantum Circuit Born Machines / PennyLane Demos* — *pennylane.ai*. https://pennylane.ai/qml/demos/tutorial_qcbm. [Accessed 25-03-2025]. (Cited on p: 39).
- [95] Román Orús, Samuel Mugel, and Enrique Lizaso. “Quantum computing for finance: Overview and prospects”. In: *Rev. Phys.* 4 (2019), p. 100028. (Cited on p: 41).
DOI: [10.1016/j.revip.2019.100028](https://doi.org/10.1016/j.revip.2019.100028).
- [96] Junyu Liu and Yuan Xin. “Quantum simulation of quantum field theories as quantum chemistry”. In: *JHEP* 12 (2020), p. 011. (Cited on p: 41).
DOI: [10.1007/JHEP12\(2020\)011](https://doi.org/10.1007/JHEP12(2020)011).
arXiv: [2004.13234](https://arxiv.org/abs/2004.13234) [hep-th].

- [97] Eric T. Holland et al. “Optimal Control for the Quantum Simulation of Nuclear Dynamics”. In: *Phys. Rev. A* 101.6 (2020), p. 062307.
(Cited on p: 41).
DOI: [10.1103/PhysRevA.101.062307](https://doi.org/10.1103/PhysRevA.101.062307).
arXiv: [1908.08222](https://arxiv.org/abs/1908.08222) [quant-ph].
- [98] J. E. Lynn et al. “Quantum Monte Carlo Methods in Nuclear Physics: Recent Advances”. In: *Ann. Rev. Nucl. Part. Sci.* 69 (2019), pp. 279–305.
(Cited on p: 41).
DOI: [10.1146/annurev-nucl-101918-023600](https://doi.org/10.1146/annurev-nucl-101918-023600).
arXiv: [1901.04868](https://arxiv.org/abs/1901.04868) [nucl-th].
- [99] Ashley Montanaro. “Quantum speedup of Monte Carlo methods”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 471.2181 (2015), p. 20150301.
(Cited on p: 41).
DOI: [10.1098/rspa.2015.0301](https://doi.org/10.1098/rspa.2015.0301).
URL: <https://doi.org/10.1098/rspa.2015.0301>.
- [100] Christian Kokail et al. “Self-verifying variational quantum simulation of lattice models”. In: *Nature* 569.7756 (2019), pp. 355–360.
(Cited on p: 41).
DOI: [10.1038/s41586-019-1177-4](https://doi.org/10.1038/s41586-019-1177-4).
arXiv: [1810.03421](https://arxiv.org/abs/1810.03421) [quant-ph].
- [101] Ryan LaRose et al. “Variational quantum state diagonalization”. In: *npj Quantum Information* 5.1 (2019), p. 57.
(Cited on p: 41).
DOI: [10.1038/s41534-019-0167-6](https://doi.org/10.1038/s41534-019-0167-6).
URL: <https://doi.org/10.1038/s41534-019-0167-6>.
- [102] Carlos Bravo-Prieto, Diego García-Martín, and José I. Latorre. “Quantum singular value decomposer”. In: *Phys. Rev. A* 101 (6 2020), p. 062310.
(Cited on p: 41).
DOI: [10.1103/PhysRevA.101.062310](https://doi.org/10.1103/PhysRevA.101.062310).
URL: <https://link.aps.org/doi/10.1103/PhysRevA.101.062310>.
- [103] Alberto Di Meglio et al. “Quantum Computing for High-Energy Physics: State of the Art and Challenges”. In: *PRX Quantum* 5.3 (2024), p. 037001.
(Cited on pp: 41, 93).
DOI: [10.1103/PRXQuantum.5.037001](https://doi.org/10.1103/PRXQuantum.5.037001).
arXiv: [2307.03236](https://arxiv.org/abs/2307.03236) [quant-ph].
- [104] Richard Keith Ellis et al. “Physics Briefing Book: Input for the European Strategy for Particle Physics Update 2020”. In: (Oct. 2019).
(Cited on p: 41).
arXiv: [1910.11775](https://arxiv.org/abs/1910.11775) [hep-ex].
- [105] Annie Y. Wei et al. “Quantum Algorithms for Jet Clustering”. In: *Phys. Rev. D* 101.9 (2020), p. 094015.
(Cited on p: 41).
DOI: [10.1103/PhysRevD.101.094015](https://doi.org/10.1103/PhysRevD.101.094015).
arXiv: [1908.08949](https://arxiv.org/abs/1908.08949) [hep-ph].

- [106] Diogo Pires et al. “A Digital Quantum Algorithm for Jet Clustering in High-Energy Physics”. In: (Jan. 2021).
(Cited on pp: 41, 42, 51).
arXiv: [2101.05618 \[physics.data-an\]](#).
- [107] Diogo Pires, Yasser Omar, and João Seixas. “Adiabatic Quantum Algorithm for Multijet Clustering in High Energy Physics”. In: (Dec. 2020).
(Cited on p: 41).
arXiv: [2012.14514 \[hep-ex\]](#).
- [108] João Barata and Carlos A. Salgado. “A quantum strategy to compute the jet quenching parameter $\hat{q}$ ”. In: *Eur. Phys. J. C* 81.10 (2021), p. 862.
(Cited on pp: 41, 60).
DOI: [10.1140/epjc/s10052-021-09674-9](#).
arXiv: [2104.04661 \[hep-ph\]](#).
- [109] Adrián Pérez-Salinas et al. “Determining the proton content with a quantum computer”. In: *Phys. Rev. D* 103.3 (2021), p. 034027.
(Cited on p: 41).
DOI: [10.1103/PhysRevD.103.034027](#).
arXiv: [2011.13934 \[hep-ph\]](#).
- [110] Christian W. Bauer et al. “Quantum Algorithm for High Energy Physics Simulations”. In: *Phys. Rev. Lett.* 126.6 (2021), p. 062001.
(Cited on p: 41).
DOI: [10.1103/PhysRevLett.126.062001](#).
arXiv: [1904.03196 \[hep-ph\]](#).
- [111] Simon Williams et al. “A quantum walk approach to simulating parton showers”. In: (Sept. 2021).
(Cited on p: 41).
arXiv: [2109.13975 \[hep-ph\]](#).
- [112] Khadeejah Bepari et al. “Towards a quantum computing algorithm for helicity amplitudes and parton showers”. In: *Phys. Rev. D* 103.7 (2021), p. 076020.
(Cited on p: 41).
DOI: [10.1103/PhysRevD.103.076020](#).
arXiv: [2010.00046 \[hep-ph\]](#).
- [113] Wibe A. De Jong et al. “Quantum simulation of open quantum systems in heavy-ion collisions”. In: *Phys. Rev. D* 104.5 (2021), p. 051501.
(Cited on p: 41).
DOI: [10.1103/PhysRevD.104.L051501](#).
arXiv: [2010.03571 \[hep-ph\]](#).
- [114] Wen Guan et al. “Quantum Machine Learning in High Energy Physics”. In: *Mach. Learn. Sci. Tech.* 2 (2021), p. 011003.
(Cited on p: 41).
DOI: [10.1088/2632-2153/abc17d](#).
arXiv: [2005.08582 \[quant-ph\]](#).
- [115] Sau Lan Wu et al. “Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the LHC on IBM quantum computer simulator and hardware with 10 qubits”. In: *J. Phys. G* 48.12

- (2021), p. 125003.
(Cited on p: 41).
DOI: [10.1088/1361-6471/ac1391](https://doi.org/10.1088/1361-6471/ac1391).
arXiv: [2012.11560](https://arxiv.org/abs/2012.11560) [quant-ph].
- [116] Timo Felser et al. “Quantum-inspired machine learning on high-energy physics data”. In: *npj Quantum Inf.* 7.1 (2021), p. 111.
(Cited on p: 41).
DOI: [10.1038/s41534-021-00443-w](https://doi.org/10.1038/s41534-021-00443-w).
arXiv: [2004.13747](https://arxiv.org/abs/2004.13747) [stat.ML].
- [117] Steve Abel, Juan C. Criado, and Michael Spannowsky. “Completely quantum neural networks”. In: *Phys. Rev. A* 106.2 (2022), p. 022601.
(Cited on p: 41).
DOI: [10.1103/PhysRevA.106.022601](https://doi.org/10.1103/PhysRevA.106.022601).
arXiv: [2202.11727](https://arxiv.org/abs/2202.11727) [quant-ph].
- [118] Jack Y. Araz and Michael Spannowsky. “Classical versus quantum: Comparing tensor-network-based quantum circuits on Large Hadron Collider data”. In: *Phys. Rev. A* 106.6 (2022), p. 062423.
(Cited on p: 41).
DOI: [10.1103/PhysRevA.106.062423](https://doi.org/10.1103/PhysRevA.106.062423).
arXiv: [2202.10471](https://arxiv.org/abs/2202.10471) [quant-ph].
- [119] Vishal S. Ngairangbam, Michael Spannowsky, and Michihisa Takeuchi. “Anomaly detection in high-energy physics using a quantum autoencoder”. In: *Phys. Rev. D* 105.9 (2022), p. 095004.
(Cited on p: 41).
DOI: [10.1103/PhysRevD.105.095004](https://doi.org/10.1103/PhysRevD.105.095004).
arXiv: [2112.04958](https://arxiv.org/abs/2112.04958) [hep-ph].
- [120] Jack Y. Araz and Michael Spannowsky. “Quantum-inspired event reconstruction with Tensor Networks: Matrix Product States”. In: *JHEP* 08 (2021), p. 112.
(Cited on p: 41).
DOI: [10.1007/JHEP08\(2021\)112](https://doi.org/10.1007/JHEP08(2021)112).
arXiv: [2106.08334](https://arxiv.org/abs/2106.08334) [hep-ph].
- [121] Andrew Blance and Michael Spannowsky. “Quantum Machine Learning for Particle Physics using a Variational Quantum Classifier”. In: *JHEP* 02 (2021), p. 212.
(Cited on p: 41).
DOI: [10.1007/JHEP02\(2021\)212](https://doi.org/10.1007/JHEP02(2021)212).
arXiv: [2010.07335](https://arxiv.org/abs/2010.07335) [hep-ph].
- [122] Stephen P. Jordan, Keith S. M. Lee, and John Preskill. “Quantum Algorithms for Quantum Field Theories”. In: *Science* 336 (2012), pp. 1130–1133.
(Cited on p: 41).
DOI: [10.1126/science.1217069](https://doi.org/10.1126/science.1217069).
arXiv: [1111.3633](https://arxiv.org/abs/1111.3633) [quant-ph].
- [123] M. C. Bañuls et al. “Simulating Lattice Gauge Theories within Quantum Technologies”. In: *Eur. Phys. J. D* 74.8 (2020), p. 165.
(Cited on p: 41).
DOI: [10.1140/epjd/e2020-100571-8](https://doi.org/10.1140/epjd/e2020-100571-8).
arXiv: [1911.00003](https://arxiv.org/abs/1911.00003) [quant-ph].

- [124] Erez Zohar, J. Ignacio Cirac, and Benni Reznik. “Quantum Simulations of Lattice Gauge Theories using Ultracold Atoms in Optical Lattices”. In: *Rept. Prog. Phys.* 79.1 (2016), p. 014401.
(Cited on p: 41).
DOI: [10.1088/0034-4885/79/1/014401](https://doi.org/10.1088/0034-4885/79/1/014401).
arXiv: [1503.02312](https://arxiv.org/abs/1503.02312) [quant-ph].
- [125] Tim Byrnes and Yoshihisa Yamamoto. “Simulating lattice gauge theories on a quantum computer”. In: *Phys. Rev. A* 73 (2006), p. 022328.
(Cited on p: 41).
DOI: [10.1103/PhysRevA.73.022328](https://doi.org/10.1103/PhysRevA.73.022328).
arXiv: [quant-ph/0510027](https://arxiv.org/abs/quant-ph/0510027).
- [126] Selomit Ramírez-Urbe et al. “Quantum algorithm for Feynman loop integrals”. In: *JHEP* 05 (2022), p. 100.
(Cited on pp: 41, 60, 62, 73, 85, 93).
DOI: [10.1007/JHEP05\(2022\)100](https://doi.org/10.1007/JHEP05(2022)100).
arXiv: [2105.08703](https://arxiv.org/abs/2105.08703) [hep-ph].
- [127] Giuseppe Clemente et al. “Variational quantum eigensolver for causal loop Feynman diagrams and directed acyclic graphs”. In: *Phys. Rev. D* 108.9 (2023), p. 096035.
(Cited on pp: 41, 60, 62, 73, 85, 93).
DOI: [10.1103/PhysRevD.108.096035](https://doi.org/10.1103/PhysRevD.108.096035).
arXiv: [2210.13240](https://arxiv.org/abs/2210.13240) [hep-ph].
- [128] Selomit Ramírez-Urbe, Andrés E. Rentería-Olivo, and Germán Rodrigo. “Quantum querying based on multicontrolled Toffoli gates for causal Feynman loop configurations and directed acyclic graphs”. In: (Apr. 2024).
(Cited on pp: 41, 60, 62, 73, 85, 93).
arXiv: [2404.03544](https://arxiv.org/abs/2404.03544) [quant-ph].
- [129] Salvador A. Ochoa-Oregon et al. “Graph theory-based automated quantum algorithm for efficient querying of acyclic and multiloop causal configurations”. In: (Aug. 2025).
(Cited on pp: 41, 60, 62, 73, 85, 93).
arXiv: [2508.04019](https://arxiv.org/abs/2508.04019) [quant-ph].
- [130] Lyndon R. Evans. *The large hadron collider : a marvel of technology*. 2nd. EPFL Press, 2018.
(Cited on p: 41).
- [131] Johannes Albrecht et al. “A Roadmap for HEP Software and Computing R&D for the 2020s”. In: *Comput. Softw. Big Sci.* 3.1 (2019), p. 7.
(Cited on p: 41).
DOI: [10.1007/s41781-018-0018-8](https://doi.org/10.1007/s41781-018-0018-8).
arXiv: [1712.06982](https://arxiv.org/abs/1712.06982) [physics.comp-ph].
- [132] ATLAS Collaboration. *ATLAS Software and Computing HL-LHC Roadmap*. Tech. rep. 182. Geneva: CERN, 2022.
(Cited on p: 41).
URL: <https://cds.cern.ch/record/2802918>.
- [133] James MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical*

- statistics and probability*. Oakland, CA, USA. 1967.
(Cited on pp: 41, 44).
- [134] Geoffrey H. Ball and David J. Hall. “A clustering technique for summarizing multivariate data”. In: *Behavioral Science* 12.2 (1967), pp. 153–155.
(Cited on pp: 41, 44).
DOI: <https://doi.org/10.1002/bs.3830120210>.
- [135] Brendan J. Frey and Delbert Dueck. “Clustering by Passing Messages Between Data Points”. In: *Science* 315.5814 (2007), pp. 972–976.
(Cited on pp: 41, 45, 54).
DOI: [10.1126/science.1136800](https://doi.org/10.1126/science.1136800).
- [136] S. Catani et al. “New clustering algorithm for multi - jet cross-sections in e^+e^- annihilation”. In: *Phys. Lett. B* 269 (1991), pp. 432–438.
(Cited on p: 41).
DOI: [10.1016/0370-2693\(91\)90196-W](https://doi.org/10.1016/0370-2693(91)90196-W).
- [137] S. Catani et al. “Longitudinally invariant K_t clustering algorithms for hadron hadron collisions”. In: *Nucl. Phys. B* 406 (1993), pp. 187–224.
(Cited on p: 41).
DOI: [10.1016/0550-3213\(93\)90166-M](https://doi.org/10.1016/0550-3213(93)90166-M).
- [138] Stephen D. Ellis and Davison E. Soper. “Successive combination jet algorithm for hadron collisions”. In: *Phys. Rev. D* 48 (1993), pp. 3160–3166.
(Cited on p: 41).
DOI: [10.1103/PhysRevD.48.3160](https://doi.org/10.1103/PhysRevD.48.3160).
arXiv: [hep-ph/9305266](https://arxiv.org/abs/hep-ph/9305266).
- [139] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. “The anti- k_t jet clustering algorithm”. In: *JHEP* 04 (2008), p. 063.
(Cited on pp: 41, 43, 55).
DOI: [10.1088/1126-6708/2008/04/063](https://doi.org/10.1088/1126-6708/2008/04/063).
arXiv: [0802.1189](https://arxiv.org/abs/0802.1189) [[hep-ph](https://arxiv.org/abs/hep-ph)].
- [140] Yuri L. Dokshitzer et al. “Better jet clustering algorithms”. In: *JHEP* 08 (1997), p. 001.
(Cited on p: 41).
DOI: [10.1088/1126-6708/1997/08/001](https://doi.org/10.1088/1126-6708/1997/08/001).
arXiv: [hep-ph/9707323](https://arxiv.org/abs/hep-ph/9707323).
- [141] Petros Drineas et al. “Clustering Large Graphs via the Singular Value Decomposition: Theoretical Advances in Data Clustering (Guest Editors: Nina Mishra and Rajeev Motwani)”. In: *Machine Learning* 56 (Jan. 2004).
(Cited on p: 42).
DOI: [10.1023/B:MACH.0000033113.59016.96](https://doi.org/10.1023/B:MACH.0000033113.59016.96).
- [142] S. Lloyd. “Least squares quantization in PCM”. In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137.
(Cited on p: 42).
DOI: [10.1109/TIT.1982.1056489](https://doi.org/10.1109/TIT.1982.1056489).
- [143] David Arthur and Sergei Vassilvitskii. “K-Means++: The Advantages of Careful Seeding”. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on*

- Discrete Algorithms*. New Orleans, Louisiana, USA. 2007.
(Cited on pp: 42, 44).
- [144] S. Chekanov. “A New jet algorithm based on the k-means clustering for the reconstruction of heavy states from jets”. In: *Eur. Phys. J. C* 47 (2006), pp. 611–616.
(Cited on p: 42).
DOI: [10.1140/epjc/s2006-02618-3](https://doi.org/10.1140/epjc/s2006-02618-3).
arXiv: [hep-ph/0512027](https://arxiv.org/abs/hep-ph/0512027).
- [145] Jesse Thaler and Ken Van Tilburg. “Maximizing Boosted Top Identification by Minimizing N-subjettiness”. In: *JHEP* 02 (2012), p. 093.
(Cited on p: 42).
DOI: [10.1007/JHEP02\(2012\)093](https://doi.org/10.1007/JHEP02(2012)093).
arXiv: [1108.2701](https://arxiv.org/abs/1108.2701) [[hep-ph](#)].
- [146] Cheuk-Yin Wong et al. “Clustering properties of produced particles in high-energy pp collisions”. In: *Phys. Rev. D* 102.5 (2020), p. 054007.
(Cited on p: 42).
DOI: [10.1103/PhysRevD.102.054007](https://doi.org/10.1103/PhysRevD.102.054007).
arXiv: [1801.00759](https://arxiv.org/abs/1801.00759) [[hep-ph](#)].
- [147] M. Leone, Sumedha, and M. Weigt. “Clustering by soft-constraint affinity propagation: applications to gene-expression data”. In: *Bioinformatics* 23.20 (2007), pp. 2708–2715.
(Cited on p: 42).
DOI: [10.1093/bioinformatics/btm414](https://doi.org/10.1093/bioinformatics/btm414).
URL: <https://doi.org/10.1093Fbioinformatics2Fbtm414>.
- [148] M. Leone Sumedha and M. Weigt. “Unsupervised and semi-supervised clustering by message passing: soft-constraint affinity propagation”. In: *The European Physical Journal B* 66.1 (2008), pp. 125–135.
(Cited on p: 42).
DOI: [10.1140/epjb/e2008-00381-8](https://doi.org/10.1140/epjb/e2008-00381-8).
URL: <https://doi.org/10.11402Fepjb2Fe2008-00381-8>.
- [149] M Bailly-Bechet et al. “Clustering with shallow trees”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2009.12 (2009), P12010.
(Cited on p: 42).
DOI: [10.1088/1742-5468/2009/12/p12010](https://doi.org/10.1088/1742-5468/2009/12/p12010).
URL: <https://doi.org/10.10882F1742-54682F20092F122Fp12010>.
- [150] Omaira González-Martín et al. “Hints on the gradual re-sizing of the torus in AGN by decomposing IRS/Spitzer spectra”. In: *The Astrophysical Journal* 841.1 (2017), p. 37.
(Cited on p: 42).
DOI: [10.3847/1538-4357/aa6f16](https://doi.org/10.3847/1538-4357/aa6f16).
URL: <https://doi.org/10.38472F1538-43572Faa6f16>.
- [151] Stefano Carrazza and José I. Latorre. “Towards the compression of parton densities through machine learning algorithms”. In: *51st Rencontres de Moriond on QCD and High Energy Interactions*. May 2016, pp. 235–238.
(Cited on p: 42).
arXiv: [1605.04345](https://arxiv.org/abs/1605.04345) [[hep-ph](#)].

- [152] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. “FastJet User Manual”. In: *Eur. Phys. J. C* 72 (2012), p. 1896.
(Cited on pp: 42, 46).
DOI: [10.1140/epjc/s10052-012-1896-2](https://doi.org/10.1140/epjc/s10052-012-1896-2).
arXiv: [1111.6097](https://arxiv.org/abs/1111.6097) [hep-ph].
- [153] Andrew Blance and Michael Spannowsky. “Unsupervised event classification with graphs on classical and photonic quantum computers”. In: *JHEP* 21 (2020), p. 170.
(Cited on pp: 42, 51).
DOI: [10.1007/JHEP08\(2021\)170](https://doi.org/10.1007/JHEP08(2021)170).
arXiv: [2103.03897](https://arxiv.org/abs/2103.03897) [hep-ph].
- [154] Abhijat Sarma et al. “Quantum unsupervised and supervised learning on superconducting processors”. In: *Quantum Information and Computation* 20 (June 2020), pp. 541–552.
(Cited on pp: 42, 51).
DOI: [10.26421/QIC20.7-8-1](https://doi.org/10.26421/QIC20.7-8-1).
- [155] Jorge J. Martinez de Lejarza. *GitHub - gmlejarza/quantum-jet-clustering*. <https://github.com/gmlejarza/Quantum-jet-clustering>. [Accessed 31-03-2025].
(Cited on p: 42).
- [156] Deepak Kar. “Jet substructure: a discovery tool”. In: *60th Annual Conference of the South African Institute of Physics*. 2015, pp. 175–179.
(Cited on p: 43).
- [157] Andrea Banfi. *Hadronic Jets (Second Edition)*. 2053-2563. IOP Publishing, 2022.
ISBN: 978-0-7503-4737-2.
(Cited on p: 43).
DOI: [10.1088/978-0-7503-4737-2](https://doi.org/10.1088/978-0-7503-4737-2).
URL: <https://dx.doi.org/10.1088/978-0-7503-4737-2>.
- [158] ATLAS Collaboration. *EventDisplayStandAlone < AtlasPublic < TWiki — twiki.cern.ch*. <https://twiki.cern.ch/twiki/bin/view/AtlasPublic/EventDisplayStandAlone>. [Accessed 01-04-2025].
(Cited on p: 43).
- [159] Gavin P. Salam. “Towards Jetography”. In: *Eur. Phys. J. C* 67 (2010), pp. 637–686.
(Cited on p: 43).
DOI: [10.1140/epjc/s10052-010-1314-6](https://doi.org/10.1140/epjc/s10052-010-1314-6).
arXiv: [0906.1833](https://arxiv.org/abs/0906.1833) [hep-ph].
- [160] Georges Aad et al. “Performance of jet substructure techniques for large- R jets in proton-proton collisions at $\sqrt{s} = 7$ TeV using the ATLAS detector”. In: *JHEP* 09 (2013), p. 076.
(Cited on p: 43).
DOI: [10.1007/JHEP09\(2013\)076](https://doi.org/10.1007/JHEP09(2013)076).
arXiv: [1306.4945](https://arxiv.org/abs/1306.4945) [hep-ex].
- [161] Philippe Gras et al. “Systematics of quark/gluon tagging”. In: *JHEP* 07 (2017), p. 091.
(Cited on p: 43).
DOI: [10.1007/JHEP07\(2017\)091](https://doi.org/10.1007/JHEP07(2017)091).
arXiv: [1704.03878](https://arxiv.org/abs/1704.03878) [hep-ph].

- [162] Thomas Langerak et al. “Multiatlas-based segmentation with preregistration atlas selection”. In: *Medical Physics* 40 (Aug. 2013), p. 091701.
(Cited on p: 46).
DOI: [10.1118/1.4816654](https://doi.org/10.1118/1.4816654).
- [163] Matteo Cacciari and Gavin P. Salam. “Dispelling the N^3 myth for the k_T jet-finder”. In: *Physics Letters B* 641.1 (2006), 57–61. ISSN: 0370-2693.
(Cited on pp: 46, 55).
DOI: [10.1016/j.physletb.2006.08.037](https://doi.org/10.1016/j.physletb.2006.08.037).
URL: <http://dx.doi.org/10.1016/j.physletb.2006.08.037>.
- [164] Harry Buhrman et al. “Quantum Fingerprinting”. In: *Phys. Rev. Lett.* 87 (16 2001), p. 167902.
(Cited on pp: 47, 127).
DOI: [10.1103/PhysRevLett.87.167902](https://doi.org/10.1103/PhysRevLett.87.167902).
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.87.167902>.
- [165] W. Bartel et al. “Experimental Evidence for Differences in p_T Between Quark Jets and Gluon Jets”. In: *Phys. Lett. B* 123 (1983), pp. 460–466.
(Cited on pp: 48, 112).
DOI: [10.1016/0370-2693\(83\)90994-2](https://doi.org/10.1016/0370-2693(83)90994-2).
- [166] S. Bethke et al. “New jet cluster algorithms: Next-to-leading order QCD and hadronization corrections”. In: *Nucl. Phys. B* 370 (1992). [Erratum: Nucl.Phys.B 523, 681–681 (1998)], pp. 310–334.
(Cited on pp: 48, 112).
DOI: [10.1016/0550-3213\(92\)90289-N](https://doi.org/10.1016/0550-3213(92)90289-N).
- [167] German Rodrigo, Mikhail S. Bilenky, and Arcadi Santamaria. “Quark mass effects for jet production in $e^+ e^-$ collisions at the next-to-leading order: Results and applications”. In: *Nucl. Phys. B* 554 (1999), pp. 257–297.
(Cited on pp: 48, 112).
DOI: [10.1016/S0550-3213\(99\)00293-X](https://doi.org/10.1016/S0550-3213(99)00293-X).
arXiv: [hep-ph/9905276](https://arxiv.org/abs/hep-ph/9905276).
- [168] Lov K. Grover. “Quantum mechanics helps in searching for a needle in a haystack”. In: *Phys. Rev. Lett.* 79 (1997), pp. 325–328.
(Cited on pp: 49, 60, 114).
DOI: [10.1103/PhysRevLett.79.325](https://doi.org/10.1103/PhysRevLett.79.325).
arXiv: [quant-ph/9706033](https://arxiv.org/abs/quant-ph/9706033).
- [169] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. “Quantum algorithms for supervised and unsupervised machine learning”. In: (2013).
(Cited on pp: 50, 113).
arXiv: [1307.0411](https://arxiv.org/abs/1307.0411) [[quant-ph](https://arxiv.org/abs/quant-ph)].
- [170] V. Giovannetti, S. Lloyd, and L. Maccone. “Quantum Random Access Memory”. In: *Physical Review Letters* 100.16 (2008). ISSN: 1079-7114.
(Cited on pp: 50, 113).
DOI: [160501](https://doi.org/10.1103/PhysRevLett.100.160501).
URL: <http://dx.doi.org/10.1103/PhysRevLett.100.160501>.
- [171] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. “Architectures for a quantum random access memory”. In: *Physical Review A* 78.5 (2008). ISSN:

- 1094-1622.
(Cited on pp: 50, 113).
DOI: [10.1103/physreva.78.052310](https://doi.org/10.1103/physreva.78.052310).
URL: <http://dx.doi.org/10.1103/PhysRevA.78.052310>.
- [172] Francesco DeMartini et al. “Experimental quantum private queries with linear optics”. In: *Phys. Rev. A* 80 (1 2009), p. 010302.
(Cited on pp: 50, 113).
DOI: [10.1103/PhysRevA.80.010302](https://doi.org/10.1103/PhysRevA.80.010302).
URL: <https://link.aps.org/doi/10.1103/PhysRevA.80.010302>.
- [173] I. Chiorescu et al. “Magnetic strong coupling in a spin-photon system and transition to classical regime”. In: *Physical Review B* 82.2 (2010). ISSN: 1550-235X.
(Cited on pp: 50, 113).
DOI: [10.1103/physrevb.82.024413](https://doi.org/10.1103/physrevb.82.024413).
URL: <http://dx.doi.org/10.1103/PhysRevB.82.024413>.
- [174] D. I. Schuster et al. “High-Cooperativity Coupling of Electron-Spin Ensembles to Superconducting Cavities”. In: *Phys. Rev. Lett.* 105 (14 2010), p. 140501.
(Cited on pp: 50, 113).
DOI: [10.1103/PhysRevLett.105.140501](https://doi.org/10.1103/PhysRevLett.105.140501).
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.105.140501>.
- [175] Y. Kubo et al. “Strong Coupling of a Spin Ensemble to a Superconducting Resonator”. In: *Phys. Rev. Lett.* 105 (14 2010), p. 140502.
(Cited on pp: 50, 113).
DOI: [10.1103/PhysRevLett.105.140502](https://doi.org/10.1103/PhysRevLett.105.140502).
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.105.140502>.
- [176] Hua Wu et al. “Storage of Multiple Coherent Microwave Excitations in an Electron Spin Ensemble”. In: *Phys. Rev. Lett.* 105 (14 2010), p. 140503.
(Cited on pp: 50, 113).
DOI: [10.1103/PhysRevLett.105.140503](https://doi.org/10.1103/PhysRevLett.105.140503).
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.105.140503>.
- [177] Patrick J. Coles et al. “Quantum Algorithm Implementations for Beginners”. In: (2018).
(Cited on p: 50).
arXiv: [1804.03719](https://arxiv.org/abs/1804.03719).
URL: <http://arxiv.org/abs/1804.03719>.
- [178] Richard L Smith. “Extreme value theory”. In: *Handbook of applicable mathematics* 7 (1990).
(Cited on pp: 50, 113).
- [179] E. J. Gumbel. *Statistics of Extremes*. Columbia University Press, 1958.
(Cited on pp: 50, 113).
- [180] Thomas M. Reiss R-D. *Statistical Analysis of Extreme Values With Applications to Insurance, Finance, Hidrology and Other Fields*. Birkhauser Verlag, 2007. ISBN: 9783764372309.
(Cited on pp: 51, 113).

- [181] Stuart Coles. *An introduction to statistical modeling of extreme values*. Springer-Verlag, 2001. ISBN: 1-85233-459-2.
(Cited on pp: 51, 113).
- [182] Castillo E. et al. *Extreme Value and Related Models with Applications in Engineering and Science*. Wiley, 2004. ISBN: 9780471671725.
(Cited on pp: 51, 113).
- [183] J. A. Hartigan and M. A. Wong. “Algorithm AS 136: A K-Means Clustering Algorithm”. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1 (1979), pp. 100–108. ISSN: 00359254, 14679876.
(Cited on p: 51).
URL: <http://www.jstor.org/stable/2346830> (visited on [visited on 07/22/2025]).
- [184] Dawid Kopczyk. “Quantum machine learning for data scientists”. In: (2018).
(Cited on p: 51).
arXiv: [1804.10068](https://arxiv.org/abs/1804.10068) [quant-ph].
- [185] R. Brun and F. Rademakers. “ROOT: An object oriented data analysis framework”. In: *Nucl. Instrum. Meth. A* 389 (1997). Ed. by M. Werlen and D. Perret-Gallix, pp. 81–86.
(Cited on pp: 53, 118).
DOI: [10.1016/S0168-9002\(97\)00048-X](https://doi.org/10.1016/S0168-9002(97)00048-X).
- [186] Gudrun Heinrich. “Collider Physics at the Precision Frontier”. In: *Phys. Rept.* 922 (2021), pp. 1–69.
(Cited on p: 59).
DOI: [10.1016/j.physrep.2021.03.006](https://doi.org/10.1016/j.physrep.2021.03.006).
arXiv: [2009.00516](https://arxiv.org/abs/2009.00516) [hep-ph].
- [187] K. G. Chetyrkin, A. L. Kataev, and F. V. Tkachov. “Higher Order Corrections to Sigma-t ($e^+ e^- \rightarrow$ Hadrons) in Quantum Chromodynamics”. In: *Phys. Lett. B* 85 (1979), pp. 277–279.
(Cited on p: 59).
DOI: [10.1016/0370-2693\(79\)90596-3](https://doi.org/10.1016/0370-2693(79)90596-3).
- [188] K. G. Chetyrkin and F. V. Tkachov. “Integration by Parts: The Algorithm to Calculate beta Functions in 4 Loops”. In: *Nucl. Phys. B* 192 (1981), pp. 159–204.
(Cited on p: 59).
DOI: [10.1016/0550-3213\(81\)90199-1](https://doi.org/10.1016/0550-3213(81)90199-1).
- [189] A. V. Kotikov. “Differential equations method: New technique for massive Feynman diagrams calculation”. In: *Phys. Lett. B* 254 (1991), pp. 158–164.
(Cited on p: 59).
DOI: [10.1016/0370-2693\(91\)90413-K](https://doi.org/10.1016/0370-2693(91)90413-K).
- [190] Ettore Remiddi. “Differential equations for Feynman graph amplitudes”. In: *Nuovo Cim. A* 110 (1997), pp. 1435–1452.
(Cited on p: 59).
DOI: [10.1007/BF03185566](https://doi.org/10.1007/BF03185566).
arXiv: [hep-th/9711188](https://arxiv.org/abs/hep-th/9711188).

- [191] A. B. Goncharov. “Multiple polylogarithms and mixed Tate motives”. In: (Mar. 2001).
(Cited on p: 59).
arXiv: [math/0103059](#).
- [192] Luise Adams, Ekta Chaubey, and Stefan Weinzierl. “Simplifying Differential Equations for Multiscale Feynman Integrals beyond Multiple Polylogarithms”. In: *Phys. Rev. Lett.* 118.14 (2017), p. 141602.
(Cited on p: 59).
DOI: [10.1103/PhysRevLett.118.141602](#).
arXiv: [1702.04279 \[hep-ph\]](#).
- [193] Johannes Broedel et al. “Elliptic multiple zeta values and one-loop superstring amplitudes”. In: *JHEP* 07 (2015), p. 112.
(Cited on p: 59).
DOI: [10.1007/JHEP07\(2015\)112](#).
arXiv: [1412.5535 \[hep-th\]](#).
- [194] Johannes Broedel et al. “Elliptic polylogarithms and iterated integrals on elliptic curves. Part I: general formalism”. In: *JHEP* 05 (2018), p. 093.
(Cited on p: 59).
DOI: [10.1007/JHEP05\(2018\)093](#).
arXiv: [1712.07089 \[hep-th\]](#).
- [195] Luise Adams, Christian Bogner, and Stefan Weinzierl. “The two-loop sunrise graph in two space-time dimensions with arbitrary masses in terms of elliptic dilogarithms”. In: *J. Math. Phys.* 55.10 (2014), p. 102301.
(Cited on p: 59).
DOI: [10.1063/1.4896563](#).
arXiv: [1405.5640 \[hep-ph\]](#).
- [196] Hjalte Frellesvig and Stefan Weinzierl. “On ϵ -factorised bases and pure Feynman integrals”. In: *SciPost Phys.* 16.6 (2024), p. 150.
(Cited on p: 59).
DOI: [10.21468/SciPostPhys.16.6.150](#).
arXiv: [2301.02264 \[hep-th\]](#).
- [197] Lennard Görge et al. “On a procedure to derive ϵ -factorised differential equations beyond polylogarithms”. In: *JHEP* 07 (2023), p. 206.
(Cited on p: 59).
DOI: [10.1007/JHEP07\(2023\)206](#).
arXiv: [2305.14090 \[hep-th\]](#).
- [198] Claude Duhr et al. “Yangian-Invariant Fishnet Integrals in Two Dimensions as Volumes of Calabi-Yau Varieties”. In: *Phys. Rev. Lett.* 130.4 (2023), p. 041602.
(Cited on p: 59).
DOI: [10.1103/PhysRevLett.130.041602](#).
arXiv: [2209.05291 \[hep-th\]](#).
- [199] Claude Duhr et al. “The ice cone family and iterated integrals for Calabi-Yau varieties”. In: *JHEP* 02 (2023), p. 228.
(Cited on p: 59).
DOI: [10.1007/JHEP02\(2023\)228](#).
arXiv: [2212.09550 \[hep-th\]](#).

- [200] Jacob L. Bourjaily et al. “Functions Beyond Multiple Polylogarithms for Precision Collider Physics”. In: *Snowmass 2021*. Mar. 2022.
(Cited on p: 59).
arXiv: [2203.07088 \[hep-ph\]](#).
- [201] Herschel A. Chawdhry et al. “NNLO QCD corrections to three-photon production at the LHC”. In: *JHEP* 02 (2020), p. 057.
(Cited on pp: 59, 60).
DOI: [10.1007/JHEP02\(2020\)057](#).
arXiv: [1911.00479 \[hep-ph\]](#).
- [202] Michal Czakon, Alexander Mitov, and Rene Poncelet. “Next-to-Next-to-Leading Order Study of Three-Jet Production at the LHC”. In: *Phys. Rev. Lett.* 127.15 (2021). [Erratum: Phys.Rev.Lett. 129, 119901 (2022), Erratum: Phys.Rev.Lett. 129, 119901 (2022)], p. 152001.
(Cited on pp: 59, 60).
DOI: [10.1103/PhysRevLett.127.152001](#).
arXiv: [2106.05331 \[hep-ph\]](#).
- [203] Samuel Abreu et al. “Two-loop master integrals for pseudo-scalar quarkonium and leptonium production and decay”. In: *JHEP* 09 (2022), p. 194.
(Cited on p: 59).
DOI: [10.1007/JHEP09\(2022\)194](#).
arXiv: [2206.03848 \[hep-ph\]](#).
- [204] Samuel Abreu et al. “Two-loop form factors for pseudo-scalar quarkonium production and decay”. In: *JHEP* 02 (2023), p. 250.
(Cited on p: 59).
DOI: [10.1007/JHEP02\(2023\)250](#).
arXiv: [2211.08838 \[hep-ph\]](#).
- [205] Simon Badger et al. “Isolated photon production in association with a jet pair through next-to-next-to-leading order in QCD”. In: *JHEP* 10 (2023), p. 071.
(Cited on p: 59).
DOI: [10.1007/JHEP10\(2023\)071](#).
arXiv: [2304.06682 \[hep-ph\]](#).
- [206] Roman N. Lee, Alexander V. Smirnov, and Vladimir A. Smirnov. “Solving differential equations for Feynman integrals by expansions near singular points”. In: *JHEP* 03 (2018), p. 008.
(Cited on p: 59).
DOI: [10.1007/JHEP03\(2018\)008](#).
arXiv: [1709.07525 \[hep-ph\]](#).
- [207] Manoj K. Mandal and Xiaoran Zhao. “Evaluating multi-loop Feynman integrals numerically through differential equations”. In: *JHEP* 03 (2019), p. 190.
(Cited on p: 59).
DOI: [10.1007/JHEP03\(2019\)190](#).
arXiv: [1812.03060 \[hep-ph\]](#).

- [208] Francesco Moriello. “Generalised power series expansions for the elliptic planar families of Higgs + jet production at two loops”. In: *JHEP* 01 (2020), p. 150.
(Cited on p: 59).
DOI: [10.1007/JHEP01\(2020\)150](https://doi.org/10.1007/JHEP01(2020)150).
arXiv: [1907.13234](https://arxiv.org/abs/1907.13234) [hep-ph].
- [209] Matteo Becchetti et al. “Next-to-leading order corrections to light-quark mixed QCD-EW contributions to Higgs boson production”. In: *Phys. Rev. D* 103.5 (2021), p. 054037.
(Cited on p: 59).
DOI: [10.1103/PhysRevD.103.054037](https://doi.org/10.1103/PhysRevD.103.054037).
arXiv: [2010.09451](https://arxiv.org/abs/2010.09451) [hep-ph].
- [210] Roberto Bonciani et al. “Mixed Strong-Electroweak Corrections to the Drell-Yan Process”. In: *Phys. Rev. Lett.* 128.1 (2022), p. 012002.
(Cited on p: 59).
DOI: [10.1103/PhysRevLett.128.012002](https://doi.org/10.1103/PhysRevLett.128.012002).
arXiv: [2106.11953](https://arxiv.org/abs/2106.11953) [hep-ph].
- [211] Tommaso Armadillo et al. “Two-loop mixed QCD-EW corrections to neutral current Drell-Yan”. In: *JHEP* 05 (2022), p. 072.
(Cited on p: 59).
DOI: [10.1007/JHEP05\(2022\)072](https://doi.org/10.1007/JHEP05(2022)072).
arXiv: [2201.01754](https://arxiv.org/abs/2201.01754) [hep-ph].
- [212] Matteo Becchetti et al. “Two-loop form factors for diphoton production in quark annihilation channel with heavy quark mass dependence, in preparation”. In: ().
(Cited on p: 59).
- [213] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM Journal on Computing* 26.5 (1997), pp. 1484–1509.
(Cited on p: 60).
DOI: [10.1137/S0097539795293172](https://doi.org/10.1137/S0097539795293172).
eprint: <https://doi.org/10.1137/S0097539795293172>.
URL: <https://doi.org/10.1137/S0097539795293172>.
- [214] Tadashi Kadowaki and Hidetoshi Nishimori. “Quantum annealing in the transverse Ising model”. In: *Phys. Rev. E* 58 (5 1998), pp. 5355–5363.
(Cited on p: 60).
DOI: [10.1103/PhysRevE.58.5355](https://doi.org/10.1103/PhysRevE.58.5355).
URL: <https://link.aps.org/doi/10.1103/PhysRevE.58.5355>.
- [215] Erez Zohar, J Ignacio Cirac, and Benni Reznik. “Quantum simulations of lattice gauge theories using ultracold atoms in optical lattices”. In: *Reports on Progress in Physics* 79.1 (2015), p. 014401.
(Cited on p: 60).
DOI: [10.1088/0034-4885/79/1/014401](https://doi.org/10.1088/0034-4885/79/1/014401).
URL: <https://dx.doi.org/10.1088/0034-4885/79/1/014401>.

- [216] Stephen P. Jordan, Keith S. M. Lee, and John Preskill. “Quantum Algorithms for Quantum Field Theories”. In: *Science* 336.6085 (2012), pp. 1130–1133.
(Cited on p: 60).
DOI: [10.1126/science.1217069](https://doi.org/10.1126/science.1217069).
eprint: <https://www.science.org/doi/pdf/10.1126/science.1217069>.
URL: <https://www.science.org/doi/abs/10.1126/science.1217069>.
- [217] Annie Y. Wei et al. “Quantum algorithms for jet clustering”. In: *Phys. Rev. D* 101 (9 2020), p. 094015.
(Cited on pp: 60, 93).
DOI: [10.1103/PhysRevD.101.094015](https://doi.org/10.1103/PhysRevD.101.094015).
URL: <https://link.aps.org/doi/10.1103/PhysRevD.101.094015>.
- [218] Andrea Delgado and Jesse Thaler. “Quantum annealing for jet clustering with thrust”. In: *Phys. Rev. D* 106 (9 2022), p. 094016.
(Cited on pp: 60, 93).
DOI: [10.1103/PhysRevD.106.094016](https://doi.org/10.1103/PhysRevD.106.094016).
URL: <https://link.aps.org/doi/10.1103/PhysRevD.106.094016>.
- [219] João Barata et al. “Quantum simulation of in-medium QCD jets: Momentum broadening, gluon production, and entropy growth”. In: *Phys. Rev. D* 108.5 (2023), p. 056023.
(Cited on p: 60).
DOI: [10.1103/PhysRevD.108.056023](https://doi.org/10.1103/PhysRevD.108.056023).
arXiv: [2307.01792](https://arxiv.org/abs/2307.01792) [hep-ph].
- [220] João Barata et al. “Medium induced jet broadening in a quantum computer”. In: *Phys. Rev. D* 106.7 (2022), p. 074013.
(Cited on p: 60).
DOI: [10.1103/PhysRevD.106.074013](https://doi.org/10.1103/PhysRevD.106.074013).
arXiv: [2208.06750](https://arxiv.org/abs/2208.06750) [hep-ph].
- [221] Adrián Pérez-Salinas et al. “Determining the proton content with a quantum computer”. In: *Phys. Rev. D* 103 (3 2021), p. 034027.
(Cited on pp: 60, 93).
DOI: [10.1103/PhysRevD.103.034027](https://doi.org/10.1103/PhysRevD.103.034027).
URL: <https://link.aps.org/doi/10.1103/PhysRevD.103.034027>.
- [222] Khadeejah Bepari et al. “Quantum walk approach to simulating parton showers”. In: *Phys. Rev. D* 106.5 (2022), p. 056002.
(Cited on p: 60).
DOI: [10.1103/PhysRevD.106.056002](https://doi.org/10.1103/PhysRevD.106.056002).
arXiv: [2109.13975](https://arxiv.org/abs/2109.13975) [hep-ph].
- [223] Vasilis Belis et al. “Quantum anomaly detection in the latent space of proton collision events at the LHC”. In: *Commun. Phys.* 7.1 (2024), p. 334.
(Cited on pp: 60, 93).
DOI: [10.1038/s42005-024-01811-6](https://doi.org/10.1038/s42005-024-01811-6).
arXiv: [2301.10780](https://arxiv.org/abs/2301.10780) [quant-ph].
- [224] Julian Schuhmacher et al. “Unravelling physics beyond the standard model with classical and quantum anomaly detection”. In: *Mach. Learn. Sci. Tech.* 4.4 (2023),

- p. 045031.
(Cited on pp: 60, 93).
DOI: [10.1088/2632-2153/ad07f7](https://doi.org/10.1088/2632-2153/ad07f7).
arXiv: [2301.10787](https://arxiv.org/abs/2301.10787) [hep-ex].
- [225] Elie Bermot et al. “Quantum Generative Adversarial Networks For Anomaly Detection In High Energy Physics”. In: *2023 International Conference on Quantum Computing and Engineering*. Apr. 2023.
(Cited on pp: 60, 93).
DOI: [10.1109/QCE57702.2023.00045](https://doi.org/10.1109/QCE57702.2023.00045).
arXiv: [2304.14439](https://arxiv.org/abs/2304.14439) [quant-ph].
- [226] Gabriele Agliardi et al. “Quantum integration of elementary particle processes”. In: *Physics Letters B* 832 (2022), p. 137228. ISSN: 0370-2693.
(Cited on pp: 60, 70, 73, 93).
DOI: <https://doi.org/10.1016/j.physletb.2022.137228>.
URL: <https://www.sciencedirect.com/science/article/pii/S0370269322003628>.
- [227] Michael Schenk et al. “Hybrid actor-critic algorithm for quantum reinforcement learning at CERN beam lines”. In: *Quantum Sci. Technol.* 9.2 (2024), p. 025012.
(Cited on p: 60).
DOI: [10.1088/2058-9565/ad261b](https://doi.org/10.1088/2058-9565/ad261b).
arXiv: [2209.11044](https://arxiv.org/abs/2209.11044) [quant-ph].
- [228] J. Jesus Aguilera-Verdugo et al. “Open Loop Amplitudes and Causality to All Orders and Powers from the Loop-Tree Duality”. In: *Phys. Rev. Lett.* 124.21 (2020), p. 211602.
(Cited on pp: 60, 62, 78, 85, 121).
DOI: [10.1103/PhysRevLett.124.211602](https://doi.org/10.1103/PhysRevLett.124.211602).
arXiv: [2001.03564](https://arxiv.org/abs/2001.03564) [hep-ph].
- [229] Stefano Catani et al. “From loops to trees by-passing Feynman’s theorem”. In: *JHEP* 09 (2008), p. 065.
(Cited on pp: 60, 62, 85, 120).
DOI: [10.1088/1126-6708/2008/09/065](https://doi.org/10.1088/1126-6708/2008/09/065).
arXiv: [0804.3170](https://arxiv.org/abs/0804.3170) [hep-ph].
- [230] Isabella Bierenbaum et al. “A Tree-Loop Duality Relation at Two Loops and Beyond”. In: *JHEP* 10 (2010), p. 073.
(Cited on pp: 60, 62, 85).
DOI: [10.1007/JHEP10\(2010\)073](https://doi.org/10.1007/JHEP10(2010)073).
arXiv: [1007.0194](https://arxiv.org/abs/1007.0194) [hep-ph].
- [231] Isabella Bierenbaum et al. “Tree-Loop Duality Relation beyond simple poles”. In: *JHEP* 03 (2013), p. 025.
(Cited on pp: 60, 62).
DOI: [10.1007/JHEP03\(2013\)025](https://doi.org/10.1007/JHEP03(2013)025).
arXiv: [1211.5048](https://arxiv.org/abs/1211.5048) [hep-ph].
- [232] Gilles Brassard et al. *Quantum amplitude amplification and estimation*. 2002.
(Cited on pp: 60, 64).
DOI: [10.1090/conm/305/05215](https://doi.org/10.1090/conm/305/05215).
URL: <https://doi.org/10.1090%2Fconm%2F305%2F05215>.

- [233] Ashley Montanaro. “Quantum speedup of Monte Carlo methods”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 471.2181 (2015).
(Cited on pp: 61, 65).
DOI: [10.1098/rspa.2015.0301](https://doi.org/10.1098/rspa.2015.0301).
URL: <https://doi.org/10.1098/rspa.2015.0301>.
- [234] Pooja Rao et al. “Quantum amplitude estimation algorithms on IBM quantum devices”. In: *Quantum Communications and Quantum Imaging XVIII*. Ed. by Keith S. Deacon. Vol. 11507. International Society for Optics and Photonics. SPIE, 2020, 115070O.
(Cited on p: 61).
DOI: [10.1117/12.2568748](https://doi.org/10.1117/12.2568748).
URL: <https://doi.org/10.1117/12.2568748>.
- [235] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2011. ISBN: 9781107002173.
(Cited on p: 61).
URL: <https://www.amazon.com/Quantum-Computation-Information-10th-Anniversary/dp/1107002176?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1107002176>.
- [236] Philip Intallura et al. “A Survey of Quantum Alternatives to Randomized Algorithms: Monte Carlo Integration and Beyond”. In: (Mar. 2023).
(Cited on p: 61).
arXiv: [2303.04945](https://arxiv.org/abs/2303.04945) [quant-ph].
- [237] Yohichi Suzuki et al. “Amplitude estimation without phase estimation”. In: *Quantum Information Processing* 19.2 (2020).
(Cited on pp: 61, 65).
DOI: [10.1007/s11128-019-2565-2](https://doi.org/10.1007/s11128-019-2565-2).
URL: <https://doi.org/10.1007/s11128-019-2565-2>.
- [238] Chu-Ryang Wie. “Simpler quantum counting”. In: *Quantum Inf. Comput.* 19 (2019), pp. 967–983.
(Cited on p: 61).
URL: <https://api.semanticscholar.org/CorpusID:197545059>.
- [239] Kirill Plekhanov et al. “Variational quantum amplitude estimation”. In: *Quantum* 6 (2022), p. 670.
(Cited on p: 61).
DOI: [10.22331/q-2022-03-17-670](https://doi.org/10.22331/q-2022-03-17-670).
arXiv: [2109.03687](https://arxiv.org/abs/2109.03687) [quant-ph].
- [240] Alexei Y. Kitaev, A. H. Shen, and Mikhail N. Vyalyi. “Classical and Quantum Computation”. In: *Graduate Studies in Mathematics*. 2002.
(Cited on p: 61).

- [241] Scott Aaronson and Patrick Rall. “Quantum Approximate Counting, Simplified”. In: *Symposium on Simplicity in Algorithms (SOSA)*, pp. 24–32. (Cited on p: 61).
DOI: [10.1137/1.9781611976014.5](https://doi.org/10.1137/1.9781611976014.5).
eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9781611976014.5>.
URL: <https://epubs.siam.org/doi/abs/10.1137/1.9781611976014.5>.
- [242] Kumar J. B. Ghosh et al. “Energy Risk Analysis With Dynamic Amplitude Estimation and Piecewise Approximate Quantum Compiling”. In: *IEEE Trans. Quantum Eng.* 5 (2024), pp. 1–17. (Cited on p: 61).
DOI: [10.1109/TQE.2024.3425969](https://doi.org/10.1109/TQE.2024.3425969).
arXiv: [2305.09501](https://arxiv.org/abs/2305.09501) [quant-ph].
- [243] Dmitry Grinko et al. “Iterative quantum amplitude estimation”. In: *npj Quantum Information* 7.1 (2021). (Cited on pp: 61, 64).
DOI: [10.1038/s41534-021-00379-1](https://doi.org/10.1038/s41534-021-00379-1).
URL: <https://doi.org/10.1038/s41534-021-00379-1>.
- [244] Steven Herbert. “Quantum Monte Carlo Integration: The Full Advantage in Minimal Circuit Depth”. In: *Quantum* 6 (2022), p. 823. (Cited on pp: 61, 67, 71).
DOI: [10.22331/q-2022-09-29-823](https://doi.org/10.22331/q-2022-09-29-823).
URL: <https://doi.org/10.22331/q-2022-09-29-823>.
- [245] R. P. Feynman. “Quantum theory of gravitation”. In: *Acta Phys. Polon.* 24 (1963). Ed. by Jong-Ping Hsu and D. Fine, pp. 697–722. (Cited on p: 62).
- [246] Sebastian Buchta et al. “On the singular behaviour of scattering amplitudes in quantum field theory”. In: *JHEP* 11 (2014), p. 014. (Cited on p: 62).
DOI: [10.1007/JHEP11\(2014\)014](https://doi.org/10.1007/JHEP11(2014)014).
arXiv: [1405.7850](https://arxiv.org/abs/1405.7850) [hep-ph].
- [247] J. Jesús Aguilera-Verdugo et al. “Causality, unitarity thresholds, anomalous thresholds and infrared singularities from the loop-tree duality at higher orders”. In: *JHEP* 12 (2019), p. 163. (Cited on p: 62).
DOI: [10.1007/JHEP12\(2019\)163](https://doi.org/10.1007/JHEP12(2019)163).
arXiv: [1904.08389](https://arxiv.org/abs/1904.08389) [hep-ph].
- [248] J. Jesus Aguilera-Verdugo et al. “Causal representation of multi-loop Feynman integrands within the loop-tree duality”. In: *JHEP* 01 (2021), p. 069. (Cited on pp: 62, 80, 85, 122).
DOI: [10.1007/JHEP01\(2021\)069](https://doi.org/10.1007/JHEP01(2021)069).
arXiv: [2006.11217](https://arxiv.org/abs/2006.11217) [hep-ph].
- [249] Selomit Ramírez-Urbe et al. “Universal opening of four-loop scattering amplitudes to trees”. In: *JHEP* 04 (2021), p. 129. (Cited on pp: 62, 85).
DOI: [10.1007/JHEP04\(2021\)129](https://doi.org/10.1007/JHEP04(2021)129).
arXiv: [2006.13818](https://arxiv.org/abs/2006.13818) [hep-ph].

- [250] J. Jesús Aguilera-Verdugo et al. “Mathematical properties of nested residues and their application to multi-loop scattering amplitudes”. In: *JHEP* 02 (2021), p. 112. (Cited on pp: 62, 85).
DOI: [10.1007/JHEP02\(2021\)112](https://doi.org/10.1007/JHEP02(2021)112).
arXiv: [2010.12971](https://arxiv.org/abs/2010.12971) [hep-ph].
- [251] German F. R. Sborlini. “Geometrical approach to causality in multiloop amplitudes”. In: *Phys. Rev. D* 104.3 (2021), p. 036014. (Cited on pp: 62, 85).
DOI: [10.1103/PhysRevD.104.036014](https://doi.org/10.1103/PhysRevD.104.036014).
arXiv: [2102.05062](https://arxiv.org/abs/2102.05062) [hep-ph].
- [252] William J. Torres Bobadilla. “Loop-tree duality from vertices and edges”. In: *JHEP* 04 (2021), p. 183. (Cited on pp: 62, 85).
DOI: [10.1007/JHEP04\(2021\)183](https://doi.org/10.1007/JHEP04(2021)183).
arXiv: [2102.05048](https://arxiv.org/abs/2102.05048) [hep-ph].
- [253] Félix Driencourt-Mangin, Germán Rodrigo, and Germán F. R. Sborlini. “Universal dual amplitudes and asymptotic expansions for $gg \rightarrow H$ and $H \rightarrow \gamma\gamma$ in four dimensions”. In: *Eur. Phys. J. C* 78.3 (2018), p. 231. (Cited on p: 62).
DOI: [10.1140/epjc/s10052-018-5692-5](https://doi.org/10.1140/epjc/s10052-018-5692-5).
arXiv: [1702.07581](https://arxiv.org/abs/1702.07581) [hep-ph].
- [254] Judith Plenter and Germán Rodrigo. “Asymptotic expansions through the loop-tree duality”. In: *Eur. Phys. J. C* 81.4 (2021), p. 320. (Cited on p: 62).
DOI: [10.1140/epjc/s10052-021-09094-9](https://doi.org/10.1140/epjc/s10052-021-09094-9).
arXiv: [2005.02119](https://arxiv.org/abs/2005.02119) [hep-ph].
- [255] Sebastian Buchta et al. “Numerical implementation of the loop-tree duality method”. In: *Eur. Phys. J. C* 77.5 (2017), p. 274. (Cited on pp: 62, 84).
DOI: [10.1140/epjc/s10052-017-4833-6](https://doi.org/10.1140/epjc/s10052-017-4833-6).
arXiv: [1510.00187](https://arxiv.org/abs/1510.00187) [hep-ph].
- [256] Felix Driencourt-Mangin et al. “Interplay between the loop-tree duality and helicity amplitudes”. In: *Phys. Rev. D* 105.1 (2022), p. 016012. (Cited on p: 62).
DOI: [10.1103/PhysRevD.105.016012](https://doi.org/10.1103/PhysRevD.105.016012).
arXiv: [1911.11125](https://arxiv.org/abs/1911.11125) [hep-ph].
- [257] Roger J. Hernandez-Pinto, German F. R. Sborlini, and German Rodrigo. “Towards gauge theories in four dimensions”. In: *JHEP* 02 (2016), p. 044. (Cited on p: 62).
DOI: [10.1007/JHEP02\(2016\)044](https://doi.org/10.1007/JHEP02(2016)044).
arXiv: [1506.04617](https://arxiv.org/abs/1506.04617) [hep-ph].
- [258] German F. R. Sborlini et al. “Four-dimensional unsubtraction from the loop-tree duality”. In: *JHEP* 08 (2016), p. 160. (Cited on p: 62).
DOI: [10.1007/JHEP08\(2016\)160](https://doi.org/10.1007/JHEP08(2016)160).
arXiv: [1604.06699](https://arxiv.org/abs/1604.06699) [hep-ph].

- [259] German F. R. Sborlini, Felix Driencourt-Mangin, and German Rodrigo. “Four-dimensional unsubtraction with massive particles”. In: *JHEP* 10 (2016), p. 162.
(Cited on p: 62).
DOI: [10.1007/JHEP10\(2016\)162](https://doi.org/10.1007/JHEP10(2016)162).
arXiv: [1608.01584](https://arxiv.org/abs/1608.01584) [hep-ph].
- [260] Renato Maria Prisco and Francesco Tramontano. “Dual subtractions”. In: *JHEP* 06 (2021), p. 089.
(Cited on p: 62).
DOI: [10.1007/JHEP06\(2021\)089](https://doi.org/10.1007/JHEP06(2021)089).
arXiv: [2012.05012](https://arxiv.org/abs/2012.05012) [hep-ph].
- [261] José de Jesús Aguilera-Verdugo et al. “A Stroll through the Loop-Tree Duality”. In: *Symmetry* 13.6 (2021), p. 1029.
(Cited on p: 62).
DOI: [10.3390/sym13061029](https://doi.org/10.3390/sym13061029).
arXiv: [2104.14621](https://arxiv.org/abs/2104.14621) [hep-ph].
- [262] Felix Driencourt-Mangin et al. “Universal four-dimensional representation of $H \rightarrow \gamma\gamma$ at two loops through the Loop-Tree Duality”. In: *JHEP* 02 (2019), p. 143.
(Cited on p: 62).
DOI: [10.1007/JHEP02\(2019\)143](https://doi.org/10.1007/JHEP02(2019)143).
arXiv: [1901.09853](https://arxiv.org/abs/1901.09853) [hep-ph].
- [263] C. G. Bollini and J. J. Giambiagi. “Dimensional Renormalization: The Number of Dimensions as a Regularizing Parameter”. In: *Nuovo Cim. B* 12 (1972), pp. 20–26.
(Cited on pp: 62, 87, 124).
DOI: [10.1007/BF02895558](https://doi.org/10.1007/BF02895558).
- [264] Gerard 't Hooft and M. J. G. Veltman. “Regularization and Renormalization of Gauge Fields”. In: *Nucl. Phys. B* 44 (1972), pp. 189–213.
(Cited on pp: 62, 87, 124).
DOI: [10.1016/0550-3213\(72\)90279-9](https://doi.org/10.1016/0550-3213(72)90279-9).
- [265] Jorge J. Martínez de Lejarza. *Tutorial: Quantum Fourier Iterative Amplitude Estimation*. June 2023.
(Cited on pp: 63, 69, 74).
DOI: [10.5281/zenodo.8130980](https://doi.org/10.5281/zenodo.8130980).
URL: <https://doi.org/10.5281/zenodo.8130980>.
- [266] *GitHub - gmlejarza/Quantum-Fourier-Iterative-Amplitude-Estimation*. <https://github.com/gmlejarza/Quantum-Fourier-Iterative-Amplitude-Estimation>. [Accessed 04-04-2025].
(Cited on p: 63).
- [267] G. Brassard and P. Hoyer. “An exact quantum polynomial-time algorithm for Simon's problem”. In: *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems*. IEEE Comput. Soc.
(Cited on p: 64).
DOI: [10.1109/istcs.1997.595153](https://doi.org/10.1109/istcs.1997.595153).
URL: <https://doi.org/10.1109/2Fistcs.1997.595153>.

- [268] Lov K. Grover. “Quantum Computers Can Search Rapidly by Using Almost Any Transformation”. In: *Phys. Rev. Lett.* 80 (19 1998), pp. 4329–4332.
(Cited on p: 64).
DOI: [10.1103/PhysRevLett.80.4329](https://doi.org/10.1103/PhysRevLett.80.4329).
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.80.4329>.
- [269] Lov K. Grover. “A Fast Quantum Mechanical Algorithm for Database Search”. In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, 212–219. ISBN: 0897917855.
(Cited on p: 64).
DOI: [10.1145/237814.237866](https://doi.org/10.1145/237814.237866).
URL: <https://doi.org/10.1145/237814.237866>.
- [270] Zoe Holmes et al. “Connecting ansatz expressibility to gradient magnitudes and barren plateaus”. In: *PRX Quantum* 3.1 (2022), p. 010313.
(Cited on p: 67).
- [271] 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).
(Cited on pp: 68, 115).
DOI: [10.1103/physreva.103.032430](https://doi.org/10.1103/physreva.103.032430).
URL: <https://doi.org/10.1103/physreva.103.032430>.
- [272] Berta Casas and Alba Cervera-Lierta. “Multidimensional Fourier series with quantum circuits”. In: *Physical Review A* 107.6 (2023).
(Cited on pp: 68, 73).
DOI: [10.1103/physreva.107.062612](https://doi.org/10.1103/physreva.107.062612).
URL: <https://doi.org/10.1103/physreva.107.062612>.
- [273] Parfait Atchade-Adelomou and Kent Larson. “Fourier series weight in quantum machine learning”. In: (Jan. 2023).
(Cited on p: 68).
arXiv: [2302.00105](https://arxiv.org/abs/2302.00105) [quant-ph].
- [274] Stavros Efthymiou et al. “Qibo: a framework for quantum simulation with hardware acceleration”. In: *Quantum Science and Technology* 7.1 (2021), p. 015018.
(Cited on pp: 69, 73, 74, 88).
DOI: [10.1088/2058-9565/ac39f5](https://doi.org/10.1088/2058-9565/ac39f5).
URL: <https://doi.org/10.1088/2058-9565/ac39f5>.
- [275] Andrew W. Cross et al. “Validating quantum computers using randomized model circuits”. In: *Phys. Rev. A* 100 (3 2019), p. 032328.
(Cited on p: 72).
DOI: [10.1103/PhysRevA.100.032328](https://doi.org/10.1103/PhysRevA.100.032328).
URL: <https://link.aps.org/doi/10.1103/PhysRevA.100.032328>.
- [276] Thomas Lubinski et al. “Application-Oriented Performance Benchmarks for Quantum Computing”. In: *IEEE Transactions on Quantum Engineering* 4 (2023), pp. 1–32.
(Cited on pp: 72, 88).
DOI: [10.1109/TQE.2023.3253761](https://doi.org/10.1109/TQE.2023.3253761).

- [277] IonQ. *Algorithmic Qubits: A Better Single-Number Metric*. <https://ionq.com/resources/algorithmic-qubits-a-better-single-number-metric>. Accessed: 2023-04-30. 2023.
(Cited on pp: 72, 88).
- [278] Matteo Robbiati et al. “A quantum analytical Adam descent through parameter shift rule using Qibo”. In: *PoS ICHEP2022* (2022), p. 206.
(Cited on p: 74).
DOI: [10.22323/1.414.0206](https://doi.org/10.22323/1.414.0206).
arXiv: [2210.10787](https://arxiv.org/abs/2210.10787) [quant-ph].
- [279] Matteo Robbiati et al. “Real-time error mitigation for variational optimization on quantum hardware”. In: (Nov. 2023).
(Cited on p: 74).
arXiv: [2311.05680](https://arxiv.org/abs/2311.05680) [quant-ph].
- [280] Stavros Efthymiou et al. “Qibolab: an open-source hybrid quantum operating system”. In: *Quantum* 8 (2024), p. 1247.
(Cited on pp: 74, 120).
DOI: [10.22331/q-2024-02-12-1247](https://doi.org/10.22331/q-2024-02-12-1247).
arXiv: [2308.06313](https://arxiv.org/abs/2308.06313) [quant-ph].
- [281] Andrea Pasquale et al. “Towards an open-source framework to perform quantum calibration and characterization”. In: *21th International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI meets Reality*. Mar. 2023.
(Cited on pp: 74, 121).
arXiv: [2303.10397](https://arxiv.org/abs/2303.10397) [quant-ph].
- [282] Qiskit contributors. *Qiskit: An Open-source Framework for Quantum Computing*. 2023.
(Cited on pp: 74, 121).
DOI: [10.5281/zenodo.2573505](https://doi.org/10.5281/zenodo.2573505).
- [283] Nathan Earnest, Caroline Tornow, and Daniel J. Egger. “Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware”. In: *Physical Review Research* 3.4 (2021).
(Cited on pp: 74, 88).
DOI: [10.1103/physrevresearch.3.043088](https://doi.org/10.1103/physrevresearch.3.043088).
URL: <https://doi.org/10.1103/physrevresearch.3.043088>.
- [284] IBMQ. *Qiskit Runtime Estimator primitive service*. https://qiskit.org/ecosystem/ibm-runtime/stubs/qiskit_ibm_runtime.Estimator.html. Accessed: 2023-11-3. 2023.
(Cited on pp: 74, 88).
- [285] André Melo, Nathan Earnest-Noble, and Francesco Tacchino. “Pulse-efficient quantum machine learning”. In: *Quantum* 7 (Oct. 2023), p. 1130. ISSN: 2521-327X.
(Cited on p: 77).
DOI: [10.22331/q-2023-10-09-1130](https://doi.org/10.22331/q-2023-10-09-1130).
URL: <https://doi.org/10.22331/q-2023-10-09-1130>.

- [286] Zvi Bern, Lance J. Dixon, and David A. Kosower. “Dimensionally regulated pentagon integrals”. In: *Nucl. Phys. B* 412 (1994), pp. 751–816.
(Cited on pp: 82, 83).
DOI: [10.1016/0550-3213\(94\)90398-0](https://doi.org/10.1016/0550-3213(94)90398-0).
arXiv: [hep-ph/9306240](https://arxiv.org/abs/hep-ph/9306240).
- [287] Francesco Tramontano. “Pentagon integrals for heavy quark physics”. In: *Phys. Rev. D* 67 (2003), p. 114005.
(Cited on p: 82).
DOI: [10.1103/PhysRevD.67.114005](https://doi.org/10.1103/PhysRevD.67.114005).
arXiv: [hep-ph/0211390](https://arxiv.org/abs/hep-ph/0211390).
- [288] Ansgar Denner and S. Dittmaier. “Reduction of one loop tensor five point integrals”. In: *Nucl. Phys. B* 658 (2003), pp. 175–202.
(Cited on p: 82).
DOI: [10.1016/S0550-3213\(03\)00184-6](https://doi.org/10.1016/S0550-3213(03)00184-6).
arXiv: [hep-ph/0212259](https://arxiv.org/abs/hep-ph/0212259).
- [289] Vittorio Del Duca et al. “The One-loop pentagon to higher orders in epsilon”. In: *JHEP* 01 (2010), p. 042.
(Cited on p: 82).
DOI: [10.1007/JHEP01\(2010\)042](https://doi.org/10.1007/JHEP01(2010)042).
arXiv: [0905.0097 \[hep-th\]](https://arxiv.org/abs/0905.0097).
- [290] Costas G. Papadopoulos. “Simplified differential equations approach for Master Integrals”. In: *JHEP* 07 (2014), p. 088.
(Cited on p: 82).
DOI: [10.1007/JHEP07\(2014\)088](https://doi.org/10.1007/JHEP07(2014)088).
arXiv: [1401.6057 \[hep-ph\]](https://arxiv.org/abs/1401.6057).
- [291] D. B. Melrose. “Reduction of Feynman diagrams”. In: *Nuovo Cim.* 40 (1965), pp. 181–213.
(Cited on p: 82).
DOI: [10.1007/BF02832919](https://doi.org/10.1007/BF02832919).
- [292] Gerard 't Hooft and M. J. G. Veltman. “Scalar One Loop Integrals”. In: *Nucl. Phys. B* 153 (1979), pp. 365–401.
(Cited on p: 82).
DOI: [10.1016/0550-3213\(79\)90605-9](https://doi.org/10.1016/0550-3213(79)90605-9).
- [293] W. L. van Neerven and J. A. M. Vermaseren. “Large loop integrals”. In: *Phys. Lett. B* 137 (1984), pp. 241–244.
(Cited on p: 82).
DOI: [10.1016/0370-2693\(84\)90237-5](https://doi.org/10.1016/0370-2693(84)90237-5).
- [294] G. J. van Oldenborgh and J. A. M. Vermaseren. “New Algorithms for One Loop Integrals”. In: *Z. Phys. C* 46 (1990), pp. 425–438.
(Cited on p: 82).
DOI: [10.1007/BF01621031](https://doi.org/10.1007/BF01621031).
- [295] William J. Torres Bobadilla. “Lotty – The loop-tree duality automation”. In: *Eur. Phys. J. C* 81.6 (2021), p. 514.
(Cited on p: 83).
DOI: [10.1140/epjc/s10052-021-09235-0](https://doi.org/10.1140/epjc/s10052-021-09235-0).
arXiv: [2103.09237 \[hep-ph\]](https://arxiv.org/abs/2103.09237).

- [296] Selomit Ramírez-Urbe et al. “Rewording Theoretical Predictions at Colliders with Vacuum Amplitudes”. In: *Phys. Rev. Lett.* 133.21 (2024), p. 211901.
(Cited on pp: 84, 85).
DOI: [10.1103/PhysRevLett.133.211901](https://doi.org/10.1103/PhysRevLett.133.211901).
arXiv: [2404.05491](https://arxiv.org/abs/2404.05491) [hep-ph].
- [297] Selomit Ramírez-Urbe et al. “From Five-Loop Scattering Amplitudes to Open Trees with the Loop-Tree Duality”. In: *Symmetry* 14.12 (2022), p. 2571.
(Cited on p: 85).
DOI: [10.3390/sym14122571](https://doi.org/10.3390/sym14122571).
arXiv: [2211.03163](https://arxiv.org/abs/2211.03163) [hep-ph].
- [298] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. 2018.
(Cited on pp: 87, 102).
URL: <http://github.com/jax-ml/jax>.
- [299] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980 (2014).
(Cited on pp: 88, 102).
URL: <https://api.semanticscholar.org/CorpusID:6628106>.
- [300] Daniel de Florian, Rodolfo Sassot, and Marco Stratmann. “Global analysis of fragmentation functions for protons and charged hadrons”. In: *Phys. Rev. D* 76 (2007), p. 074033.
(Cited on p: 92).
DOI: [10.1103/PhysRevD.76.074033](https://doi.org/10.1103/PhysRevD.76.074033).
arXiv: [0707.1506](https://arxiv.org/abs/0707.1506) [hep-ph].
- [301] Christine A. Aidala et al. “Global Analysis of Fragmentation Functions for Eta Mesons”. In: *Phys. Rev. D* 83 (2011), p. 034002.
(Cited on p: 92).
DOI: [10.1103/PhysRevD.83.034002](https://doi.org/10.1103/PhysRevD.83.034002).
arXiv: [1009.6145](https://arxiv.org/abs/1009.6145) [hep-ph].
- [302] Daniel de Florian, Rodolfo Sassot, and Marco Stratmann. “Global analysis of fragmentation functions for pions and kaons and their uncertainties”. In: *Phys. Rev. D* 75 (2007), p. 114010.
(Cited on p: 92).
DOI: [10.1103/PhysRevD.75.114010](https://doi.org/10.1103/PhysRevD.75.114010).
arXiv: [hep-ph/0703242](https://arxiv.org/abs/hep-ph/0703242).
- [303] John C. Collins, Davison E. Soper, and George F. Sterman. “Factorization of Hard Processes in QCD”. In: *Adv. Ser. Direct. High Energy Phys.* 5 (1989), pp. 1–91.
(Cited on p: 92).
DOI: [10.1142/9789814503266_0001](https://doi.org/10.1142/9789814503266_0001).
arXiv: [hep-ph/0409313](https://arxiv.org/abs/hep-ph/0409313).
- [304] John C. Collins and Davison E. Soper. “Parton Distribution and Decay Functions”. In: *Nucl. Phys. B* 194 (1982), pp. 445–492.
(Cited on p: 92).
DOI: [10.1016/0550-3213\(82\)90021-9](https://doi.org/10.1016/0550-3213(82)90021-9).
- [305] Elke C. Aschenauer et al. “Semi-inclusive deep-inelastic scattering, parton distributions, and fragmentation functions at a future electron-ion collider”. In:

- Physical Review D* 99.9 (2019). Cited by: 26; All Open Access, Green Open Access, Hybrid Gold Open Access.
(Cited on p: 92).
DOI: [10.1103/PhysRevD.99.094004](https://doi.org/10.1103/PhysRevD.99.094004).
URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85066947666&doi=10.1103%2fPhysRevD.99.094004&partnerID=40&md5=2fdbe22a723d31038bd6cf13cfad22de>.
- [306] Juan Rojo et al. “The PDF4LHC report on PDFs and LHC data: Results from Run I and preparation for Run II”. In: *J. Phys. G* 42 (2015), p. 103103.
(Cited on p: 92).
DOI: [10.1088/0954-3899/42/10/103103](https://doi.org/10.1088/0954-3899/42/10/103103).
arXiv: [1507.00556](https://arxiv.org/abs/1507.00556) [hep-ph].
- [307] S. Albino et al. “Parton fragmentation in the vacuum and in the medium”. In: (Apr. 2008).
(Cited on p: 92).
arXiv: [0804.2021](https://arxiv.org/abs/0804.2021) [hep-ph].
- [308] M. Glück, E. Reya, and A. Vogt. “Dynamical parton distributions revisited”. In: *Eur. Phys. J. C* 5 (1998), pp. 461–470.
(Cited on p: 92).
DOI: [10.1007/s100520050289](https://doi.org/10.1007/s100520050289).
arXiv: [hep-ph/9806404](https://arxiv.org/abs/hep-ph/9806404).
- [309] S. Kretzer. “Fragmentation functions from flavor inclusive and flavor tagged e^+e^- annihilations”. In: *Phys. Rev. D* 62 (2000), p. 054001.
(Cited on p: 92).
DOI: [10.1103/PhysRevD.62.054001](https://doi.org/10.1103/PhysRevD.62.054001).
arXiv: [hep-ph/0003177](https://arxiv.org/abs/hep-ph/0003177).
- [310] Bernd A. Kniehl, G. Kramer, and B. Potter. “Fragmentation functions for pions, kaons, and protons at next-to-leading order”. In: *Nucl. Phys. B* 582 (2000), pp. 514–536.
(Cited on p: 92).
DOI: [10.1016/S0550-3213\(00\)00303-5](https://doi.org/10.1016/S0550-3213(00)00303-5).
arXiv: [hep-ph/0010289](https://arxiv.org/abs/hep-ph/0010289).
- [311] M. Hirai et al. “Determination of fragmentation functions and their uncertainties”. In: *Phys. Rev. D* 75 (2007), p. 094009.
(Cited on p: 92).
DOI: [10.1103/PhysRevD.75.094009](https://doi.org/10.1103/PhysRevD.75.094009).
arXiv: [hep-ph/0702250](https://arxiv.org/abs/hep-ph/0702250).
- [312] Emanuele R. Nocera. “Towards a Neural Network Determination of Charged Pion Fragmentation Functions”. In: *22nd International Symposium on Spin Physics*. Jan. 2017.
(Cited on p: 92).
arXiv: [1701.09186](https://arxiv.org/abs/1701.09186) [hep-ph].
- [313] Valerio Bertone et al. “Towards a Neural Network determination of Pion Fragmentation Functions”. In: *Proceedings, Parton Radiation and Fragmentation from LHC to FCC-ee: CERN, Geneva, Switzerland, November 22-23, 2016*. 2017,

- pp. 19–25.
(Cited on p: 92).
URL: http://inspirehep.net/record/1512989/files/1512294_19-25.pdf.
- [314] Valerio Bertone et al. “A determination of the fragmentation functions of pions, kaons, and protons with faithful uncertainties”. In: *Eur. Phys. J. C* 77.8 (2017), p. 516.
(Cited on pp: 92, 95).
DOI: [10.1140/epjc/s10052-017-5088-y](https://doi.org/10.1140/epjc/s10052-017-5088-y).
arXiv: [1706.07049](https://arxiv.org/abs/1706.07049) [hep-ph].
- [315] Maryam Soleymaninia et al. “Fragmentation functions for $\Xi^-/\Xi^- +$ using neural networks”. In: *Nucl. Phys. A* 1029 (2023), p. 122564.
(Cited on p: 92).
DOI: [10.1016/j.nuclphysa.2022.122564](https://doi.org/10.1016/j.nuclphysa.2022.122564).
arXiv: [2202.05586](https://arxiv.org/abs/2202.05586) [hep-ph].
- [316] Richard D. Ball et al. “An open-source machine learning framework for global analyses of parton distributions”. In: *Eur. Phys. J. C* 81.10 (2021), p. 958.
(Cited on p: 92).
DOI: [10.1140/epjc/s10052-021-09747-9](https://doi.org/10.1140/epjc/s10052-021-09747-9).
arXiv: [2109.02671](https://arxiv.org/abs/2109.02671) [hep-ph].
- [317] V. N. Gribov and L. N. Lipatov. “Deep inelastic e p scattering in perturbation theory”. In: *Sov. J. Nucl. Phys.* 15 (1972), pp. 438–450.
(Cited on pp: 92, 94).
- [318] L. N. Lipatov. “The parton model and perturbation theory”. In: *Yad. Fiz.* 20 (1974), pp. 181–198.
(Cited on pp: 92, 94).
- [319] Guido Altarelli and G. Parisi. “Asymptotic Freedom in Parton Language”. In: *Nucl. Phys. B* 126 (1977), pp. 298–318.
(Cited on pp: 92, 94).
DOI: [10.1016/0550-3213\(77\)90384-4](https://doi.org/10.1016/0550-3213(77)90384-4).
- [320] Yuri L. Dokshitzer. “Calculation of the Structure Functions for Deep Inelastic Scattering and $e^+ e^-$ Annihilation by Perturbation Theory in Quantum Chromodynamics.” In: *Sov. Phys. JETP* 46 (1977), pp. 641–653.
(Cited on pp: 92, 94).
- [321] Germán Rodrigo. “Quantum Algorithms in Particle Physics”. In: *Acta Phys. Polon. Supp.* 17.2 (2024), 2–A14.
(Cited on p: 93).
DOI: [10.5506/APhysPolBSupp.17.2-A14](https://doi.org/10.5506/APhysPolBSupp.17.2-A14).
arXiv: [2401.16208](https://arxiv.org/abs/2401.16208) [hep-ph].
- [322] Jorge J. Martínez de Lejarza, Leandro Cieri, and Germán Rodrigo. “Quantum clustering and jet reconstruction at the LHC”. In: *Phys. Rev. D* 106 (3 2022), p. 036021.
(Cited on p: 93).
DOI: [10.1103/PhysRevD.106.036021](https://doi.org/10.1103/PhysRevD.106.036021).
URL: <https://link.aps.org/doi/10.1103/PhysRevD.106.036021>.

- [323] Vasilis Belis et al. “Guided quantum compression for high dimensional data classification”. In: *Mach. Learn. Sci. Tech.* 5.3 (2024), p. 035010.
(Cited on p: 93).
DOI: [10.1088/2632-2153/ad5fdd](https://doi.org/10.1088/2632-2153/ad5fdd).
arXiv: [2402.09524](https://arxiv.org/abs/2402.09524) [quant-ph].
- [324] Vasileios Belis et al. “Higgs analysis with quantum classifiers”. In: *EPJ Web Conf.* 251 (2021), p. 03070.
(Cited on p: 93).
DOI: [10.1051/epjconf/202125103070](https://doi.org/10.1051/epjconf/202125103070).
arXiv: [2104.07692](https://arxiv.org/abs/2104.07692) [quant-ph].
- [325] Juan M. Cruz-Martinez, Matteo Robbiati, and Stefano Carrazza. “Multi-variable integration with a variational quantum circuit”. In: *Quantum Sci. Technol.* 9.3 (2024), p. 035053.
(Cited on p: 93).
DOI: [10.1088/2058-9565/ad5866](https://doi.org/10.1088/2058-9565/ad5866).
arXiv: [2308.05657](https://arxiv.org/abs/2308.05657) [quant-ph].
- [326] Zhong-Bo Kang et al. “Partonic distribution functions and amplitudes using tensor network methods”. In: (Jan. 2025).
(Cited on p: 93).
arXiv: [2501.09738](https://arxiv.org/abs/2501.09738) [hep-ph].
- [327] Andrea Delgado, Francisco Rios, and Kathleen E. Hamilton. “Identifying overparameterization in Quantum Circuit Born Machines”. In: (July 2023).
(Cited on p: 93).
arXiv: [2307.03292](https://arxiv.org/abs/2307.03292) [quant-ph].
- [328] Andrea Delgado et al. “Towards Designing Scalable Quantum-Enhanced Generative Networks for Neutrino Physics Experiments with Liquid Argon Time Projection Chambers”. In: (Oct. 2024).
(Cited on p: 93).
arXiv: [2410.12650](https://arxiv.org/abs/2410.12650) [quant-ph].
- [329] Cenk Tüysüz et al. “Learning to generate high-dimensional distributions with low-dimensional quantum Boltzmann machines”. In: (Oct. 2024).
(Cited on p: 93).
arXiv: [2410.16363](https://arxiv.org/abs/2410.16363) [quant-ph].
- [330] Manuel S. Rudolph et al. “Trainability barriers and opportunities in quantum generative modeling”. In: *npj Quantum Inf.* 10.1 (2024), p. 116.
(Cited on p: 93).
DOI: [10.1038/s41534-024-00902-0](https://doi.org/10.1038/s41534-024-00902-0).
arXiv: [2305.02881](https://arxiv.org/abs/2305.02881) [quant-ph].
- [331] Mohammad H. Amin et al. “Quantum Boltzmann Machine”. In: *Phys. Rev. X* 8 (2018), p. 021050.
(Cited on p: 93).
DOI: [10.1103/PhysRevX.8.021050](https://doi.org/10.1103/PhysRevX.8.021050).
URL: <https://link.aps.org/doi/10.1103/PhysRevX.8.021050>.

- [332] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. “Variational quantum Boltzmann machines”. In: *Quantum Machine Intelligence* 3.7 (2021). (Cited on p: 93).
DOI: doi.org/10.1007/s42484-020-00033-7.
URL: <https://link.springer.com/article/10.1007/s42484-020-00033-7>.
- [333] Luuk Coopmans and Marcello Benedetti. “On the sample complexity of quantum Boltzmann machine learning”. In: *Commun. Phys.* 7.274 (2024). (Cited on p: 93).
DOI: [10.1038/s42005-024-01763-x](https://doi.org/10.1038/s42005-024-01763-x).
URL: <https://doi.org/10.1038/s42005-024-01763-x>.
- [334] Seth Lloyd and Christian Weedbrook. “Quantum Generative Adversarial Learning”. In: *Phys. Rev. Lett.* 121 (4 2018), p. 040502. (Cited on p: 93).
DOI: [10.1103/PhysRevLett.121.040502](https://doi.org/10.1103/PhysRevLett.121.040502).
URL: <https://link.aps.org/doi/10.1103/PhysRevLett.121.040502>.
- [335] Pierre-Luc Dallaire-Demers and Nathan Killoran. “Quantum generative adversarial networks”. In: *Phys. Rev. A* 98 (1 2018), p. 012324. (Cited on p: 93).
DOI: [10.1103/PhysRevA.98.012324](https://doi.org/10.1103/PhysRevA.98.012324).
URL: <https://link.aps.org/doi/10.1103/PhysRevA.98.012324>.
- [336] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. “Quantum Generative Adversarial Networks for learning and loading random distributions”. In: *npj Quantum Information* 5.1 (2019), p. 103. ISSN: 2056-6387. (Cited on p: 93).
DOI: [10.1038/s41534-019-0223-2](https://doi.org/10.1038/s41534-019-0223-2).
URL: <https://doi.org/10.1038/s41534-019-0223-2>.
- [337] He-Liang Huang et al. “Experimental Quantum Generative Adversarial Networks for Image Generation”. In: *Phys. Rev. Applied* 16 (2 2021), p. 024051. (Cited on p: 93).
DOI: [10.1103/PhysRevApplied.16.024051](https://doi.org/10.1103/PhysRevApplied.16.024051).
URL: <https://link.aps.org/doi/10.1103/PhysRevApplied.16.024051>.
- [338] Jin-Guo Liu and Lei Wang. “Differentiable learning of quantum circuit Born machines”. In: *Phys. Rev. A* 98 (6 2018), p. 062324. (Cited on p: 93).
DOI: [10.1103/PhysRevA.98.062324](https://doi.org/10.1103/PhysRevA.98.062324).
URL: <https://link.aps.org/doi/10.1103/PhysRevA.98.062324>.
- [339] Marcello Benedetti et al. “A generative modeling approach for benchmarking and training shallow quantum circuits”. In: *npj Quantum Information* 5.1 (2019), p. 45. ISSN: 2056-6387. (Cited on p: 93).
DOI: [10.1038/s41534-019-0157-8](https://doi.org/10.1038/s41534-019-0157-8).
URL: <https://www.nature.com/articles/s41534-019-0157-8>.
- [340] Marcello Benedetti et al. “Erratum: Parameterized quantum circuits as machine learning models (2019 Quant. Sci. Tech. 4 043001)”. In: *Quantum Science and*

- Technology* 5.1 (2019), p. 019601.
(Cited on p: 93).
DOI: [10.1088/2058-9565/ab5944](https://doi.org/10.1088/2058-9565/ab5944).
URL: <https://dx.doi.org/10.1088/2058-9565/ab5944>.
- [341] Chelsea A. Williams et al. “Quantum Chebyshev Transform: Mapping, Embedding, Learning and Sampling Distributions”. In: (June 2023).
(Cited on pp: 93, 95, 97, 98, 116).
arXiv: [2306.17026](https://arxiv.org/abs/2306.17026) [quant-ph].
- [342] Oleksandr Kyriienko, Annie E. Paine, and Vincent E. Elfving. “Protocols for trainable and differentiable quantum generative modeling”. In: *Phys. Rev. Res.* 6 (3 2024), p. 033291.
(Cited on p: 93).
DOI: [10.1103/PhysRevResearch.6.033291](https://doi.org/10.1103/PhysRevResearch.6.033291).
URL: <https://link.aps.org/doi/10.1103/PhysRevResearch.6.033291>.
- [343] Hsin-Yu Wu, Vincent E. Elfving, and Oleksandr Kyriienko. “Multidimensional Quantum Generative Modeling by Quantum Hartley Transform”. In: *Adv. Quantum Technol.* 8.3 (2025), p. 2400337.
(Cited on p: 93).
DOI: [10.1002/qute.202400337](https://doi.org/10.1002/qute.202400337).
arXiv: [2406.03856](https://arxiv.org/abs/2406.03856) [quant-ph].
- [344] Jorge J. Martinez de Lejarza. *GitHub - CERN-IT-INNOVATION/QChPM: Quantum Chebyshev Probabilistic Models for Fragmentation Functions* — *github.com*. <https://github.com/CERN-IT-INNOVATION/QChPM>. [Accessed 12-08-2025]. 2025.
(Cited on p: 93).
- [345] R. Keith Ellis, W. James Stirling, and B. R. Webber. *QCD and collider physics*. Vol. 8. Cambridge University Press, Feb. 2011. ISBN: 978-0-511-82328-2, 978-0-521-54589-1.
(Cited on p: 94).
DOI: [10.1017/CB09780511628788](https://doi.org/10.1017/CB09780511628788).
- [346] D. de Florian, M. Stratmann, and W. Vogelsang. “QCD analysis of unpolarized and polarized Lambda baryon production in leading and next-to-leading order”. In: *Phys. Rev. D* 57 (1998), pp. 5811–5824.
(Cited on p: 94).
DOI: [10.1103/PhysRevD.57.5811](https://doi.org/10.1103/PhysRevD.57.5811).
arXiv: [hep-ph/9711387](https://arxiv.org/abs/hep-ph/9711387).
- [347] A. A. Almasy, S. Moch, and A. Vogt. “On the Next-to-Next-to-Leading Order Evolution of Flavour-Singlet Fragmentation Functions”. In: *Nucl. Phys. B* 854 (2012), pp. 133–152.
(Cited on p: 95).
DOI: [10.1016/j.nuclphysb.2011.08.028](https://doi.org/10.1016/j.nuclphysb.2011.08.028).
arXiv: [1107.2263](https://arxiv.org/abs/1107.2263) [hep-ph].

- [348] S. Moch and A. Vogt. “On third-order timelike splitting functions and top-mediated Higgs decay into hadrons”. In: *Phys. Lett. B* 659 (2008), pp. 290–296.
(Cited on p: 95).
DOI: [10.1016/j.physletb.2007.10.069](https://doi.org/10.1016/j.physletb.2007.10.069).
arXiv: [0709.3899](https://arxiv.org/abs/0709.3899) [hep-ph].
- [349] A. Mitov, S. Moch, and A. Vogt. “Next-to-Next-to-Leading Order Evolution of Non-Singlet Fragmentation Functions”. In: *Phys. Lett. B* 638 (2006), pp. 61–67.
(Cited on p: 95).
DOI: [10.1016/j.physletb.2006.05.005](https://doi.org/10.1016/j.physletb.2006.05.005).
arXiv: [hep-ph/0604053](https://arxiv.org/abs/hep-ph/0604053).
- [350] Raban Iten et al. “Quantum circuits for isometries”. In: *Physical Review A* 93.3 (Mar. 2016). ISSN: 2469-9934.
(Cited on pp: 96, 116).
DOI: [10.1103/physreva.93.032318](https://doi.org/10.1103/physreva.93.032318).
URL: <http://dx.doi.org/10.1103/PhysRevA.93.032318>.
- [351] Oleksandr Kyriienko, Annie E. Paine, and Vincent E. Elfving. “Protocols for trainable and differentiable quantum generative modeling”. In: *Phys. Rev. Res.* 6.3 (2024), p. 033291.
(Cited on pp: 96, 117).
DOI: [10.1103/PhysRevResearch.6.033291](https://doi.org/10.1103/PhysRevResearch.6.033291).
arXiv: [2202.08253](https://arxiv.org/abs/2202.08253) [quant-ph].
- [352] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. “Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision”. In: *SIAM Journal on Computing* 46.6 (2017), pp. 1920–1950.
(Cited on pp: 96, 117).
DOI: [10.1137/16M1087072](https://doi.org/10.1137/16M1087072).
URL: <https://doi.org/10.1137/16M1087072>.
- [353] Andreas Klappenecker and Martin Roetteler. “Discrete Cosine Transforms on Quantum Computers”. In: *IEEE R8-EURASIP Symposium on Image and Signal Processing and Analysis*, pp. 464–468 (Nov. 2001).
(Cited on pp: 97, 117).
arXiv: [quant-ph/0111038](https://arxiv.org/abs/quant-ph/0111038).
- [354] Ville Bergholm et al. “PennyLane: Automatic differentiation of hybrid quantum-classical computations”. In: (Nov. 2018).
(Cited on p: 102).
arXiv: [1811.04968](https://arxiv.org/abs/1811.04968) [quant-ph].
- [355] Andy Buckley et al. “LHAPDF6: parton density access in the LHC precision era”. In: *Eur. Phys. J. C* 75 (2015), p. 132.
(Cited on p: 102).
DOI: [10.1140/epjc/s10052-015-3318-8](https://doi.org/10.1140/epjc/s10052-015-3318-8).
arXiv: [1412.7420](https://arxiv.org/abs/1412.7420) [hep-ph].
- [356] Andy Buckley et al. *LHAPDF: Main Page — lhpdf.org*. <https://www.lhapdf.org>. [Accessed 16-01-2025]. 2015.
(Cited on p: 102).
- [357] Stefano Scali, Chukwudubem Umeano, and Oleksandr Kyriienko. “The topology of data hides in quantum thermal states”. In: *APL Quantum* 1.3 (July 2024), p. 036106.

- ISSN: 2835-0103.
 (Cited on p: 105).
 DOI: [10.1063/5.0209201](https://doi.org/10.1063/5.0209201).
 eprint: https://pubs.aip.org/aip/apq/article-pdf/doi/10.1063/5.0209201/20040419/036106_1_5.0209201.pdf.
 URL: <https://doi.org/10.1063/5.0209201>.
- [358] Annie E. Paine, Vincent E. Elfving, and Oleksandr Kyriienko. “Quantum kernel methods for solving regression problems and differential equations”. In: *Phys. Rev. A* 107.3 (2023), p. 032428.
 (Cited on p: 105).
 DOI: [10.1103/PhysRevA.107.032428](https://doi.org/10.1103/PhysRevA.107.032428).
 arXiv: [2203.08884](https://arxiv.org/abs/2203.08884) [quant-ph].
- [359] J. von Neumann. “Thermodynamik quantenmechanischer Gesamtheiten”. ger. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1927 (1927), pp. 273–291.
 (Cited on p: 132).
 URL: <http://eudml.org/doc/59231>.
- [360] Charles H. Bennett et al. “Concentrating partial entanglement by local operations”. In: *Phys. Rev. A* 53 (4 1996), pp. 2046–2052.
 (Cited on p: 132).
 DOI: [10.1103/PhysRevA.53.2046](https://doi.org/10.1103/PhysRevA.53.2046).
 URL: <https://link.aps.org/doi/10.1103/PhysRevA.53.2046>.
- [361] Sam A. Hill and William K. Wootters. “Entanglement of a Pair of Quantum Bits”. In: *Phys. Rev. Lett.* 78 (26 1997), pp. 5022–5025.
 (Cited on p: 132).
 DOI: [10.1103/PhysRevLett.78.5022](https://doi.org/10.1103/PhysRevLett.78.5022).
 URL: <https://link.aps.org/doi/10.1103/PhysRevLett.78.5022>.
- [362] Xanadu. *qml.math.vn_entropy* — PennyLane — docs.pennylane.ai. https://docs.pennylane.ai/en/stable/code/api/pennylane.math.vn_entropy.html.
 [Accessed 28-01-2025].
 (Cited on p: 132).
- [363] Xiao-Lan Zong et al. “Monogamy of Quantum Entanglement”. In: *Frontiers in Physics* 10 (June 2022). ISSN: 2296-424X.
 (Cited on p: 133).
 DOI: [10.3389/fphy.2022.880560](https://doi.org/10.3389/fphy.2022.880560).
 URL: <http://dx.doi.org/10.3389/fphy.2022.880560>.
- [364] Gregg Jaeger. *Quantum Information: An Overview*. New York, NY, USA: Springer, 2007.
 (Cited on p: 134).