

Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (38.th cycle)

Practical Hybrid Quantum-Classical Algorithms for Noisy Intermediate Scale Quantum Devices

Applications and Implementations on Neutral-Atom Quantum Computing Platforms

Chiara Vercellino

* * * * *

Supervisors
Prof. Bartolomeo Montrucchio, Supervisor
Dr. Olivier Terzo, Co-supervisor

Politecnico di Torino
2026

This thesis is licensed under a Creative Commons License, Attribution - Noncommercial-NoDerivative Works 4.0 International: see www.creativecommons.org. The text may be reproduced for non-commercial purposes, provided that credit is given to the original author.

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . . . .

Chiara Vercellino

Turin, 2026

Summary

Quantum computing is often presented as a disruptive technology with the potential to accelerate problems that are hard for classical machines. Yet, in the current NISQ (Noisy Intermediate-Scale Quantum) era, the main challenge is no longer only the design of formally correct quantum algorithms, but their translation into practical workflows compatible with real hardware and meaningful application domains. This thesis addresses that challenge by investigating practical hybrid quantum-classical methodologies for neutral-atom quantum processors, with particular attention to analog computation, hardware-aware embedding, combinatorial optimization, and quantum machine learning.

A first contribution concerns the development of embedding strategies for neutral-atom platforms. Since, in analog processors, computation is tightly linked to geometry and distance-dependent interactions, mapping abstract problems onto the physical register is a central part of the algorithmic pipeline. This thesis studies methods to construct feasible and scalable embeddings for graph-based problems, showing how data representation and hardware constraints directly affect the quality and applicability of quantum routines.

Building on this foundation, the thesis explores hybrid approaches to quantum optimization. The main focus is on graph-based combinatorial problems, in particular vertex coloring and maximum-weighted independent set, which are relevant both from a theoretical viewpoint and for industrial applications such as Physical Cell Identity assignment and the design of space–terrestrial integrated networks. Several increasingly advanced hybrid frameworks are developed, combining classical optimization techniques with quantum subroutines on neutral-atom hardware. The results show that these approaches can improve solution quality while preserving resource efficiency, and underscore the importance of tight problem formulations, strong classical bounds, and hardware-aware branching or decomposition strategies.

The final major research direction concerns quantum machine learning. Two case studies are considered. The first investigates graph classification through quantum evolution kernels on neutral-atom processors, applied to protein classification. The results demonstrate that quantum dynamics can define informative similarity measures for graph-structured data, with hardware execution on Aquila's quantum processor achieving performance comparable to strong classical baselines. The second case study

extends beyond graph data and studies credit card default prediction through quantum reservoir computing. In this setting, the quantum system acts as a nonlinear feature generator within a hybrid learning pipeline. While hardware noise affects performance more significantly than in the kernel-based case, the results show that quantum-generated features remain useful and that classical post-processing can partially mitigate imperfections.

Overall, the thesis shows that the practical value of current quantum computing lies not in isolated quantum routines, but in carefully designed hybrid pipelines in which quantum and classical resources complement each other. The results highlight embedding as a fundamental problem, confirm the relevance of hardware-aware algorithm design, and show that neutral-atom platforms already support meaningful application-driven studies in both optimization and machine learning. These findings contribute to narrowing the gap between theoretical quantum computing and realistic industrial use cases, while also identifying the main limitations and opportunities for future research in the NISQ regime.

Acknowledgements

This work was supported by several institutions and organizations, whose contributions were essential to the research presented in this thesis.

The author gratefully acknowledges the *Pawsey Supercomputing Research Centre* and *Amazon Web Services Braket* for granting access to quantum resources. Access to high-performance computing resources was provided by *CINECA* consortium, whose infrastructure supported a significant portion of the computational experiments.

The author further acknowledges *Pasqal* and *QuEra Computing* for providing valuable insights into their respective quantum hardware architectures and methods, as well as for granting access to their quantum platforms.

This doctoral research was made possible through *LINKS Foundation*, which hosted and enabled the PhD program. Part of the results presented in this thesis were developed within the *QRISTIN* project, funded by the Next Generation Internet initiative *NGI Sargasso* under the Horizon Europe programme (Grant Agreement No. 101092887).

Finally, the author wishes to express sincere gratitude to the supervisors, Prof. Bartolomeo Montrucchio and Dr. Olivier Terzo, colleagues, with special mention to Giacomo Vitali, and collaborators whose guidance and support were invaluable throughout this research.

Contents

List of Tables x

List of Figures xii

List of Acronyms xvi

1 Introduction 1
- 1.1 Hybrid Quantum-Classical Algorithms . . . 2
  - 1.1.1 Variational and Iterative Hybrid Frameworks . . . 3
  - 1.1.2 Hardware-Aware Algorithm Design . . . 4
- 1.2 Noisy Intermediate-Scale Quantum Devices . . . 4
  - 1.2.1 Comparative Perspective . . . 8
  - 1.2.2 Neutral Atom Quantum Processors . . . 8
- 1.3 Thesis Contributions and Organization . . . 9

2 Hardware-Aware Encoding for Hybrid Quantum Algorithms on Neutral-Atom Devices 13
- 2.1 Background and Related Approaches . . . 17
- 2.2 Hardware-Induced Connectivity in Neutral-Atom Quantum Processors 18
  - 2.2.1 From Hardware Constraints to Embedding Requirements . . . 19
  - 2.2.2 Representable Graph Classes and Unit-Disk Abstractions . . . 20
- 2.3 The Constrained Unit-Disk Graph Embedding as an Optimization Problem 22
  - 2.3.1 Mathematical Programming Formulation . . . 22
  - 2.3.2 Computational Complexity and Hardness . . . 24
- 2.4 Neural Optimization for Embedding under Hardware Constraints . . . 25
  - 2.4.1 Initialization of Vertex Coordinates . . . 26
  - 2.4.2 Neural Architecture Design . . . 28
  - 2.4.3 Embedding Loss Function and Constraint Penalization . . . 29
  - 2.4.4 Adapting the Embedding to Real Quantum Processors . . . 31
  - 2.4.5 Beyond Two-Dimensional Embeddings . . . 33
- 2.5 Empirical Evaluation . . . 35
  - 2.5.1 Benchmark Datasets and Experimental Setup . . . 36

2.5.2 Results Analysis . . . 39
2.6 Limitations, Practical Adaptations, and Outlook . . . 48
2.6.1 Integration into Hybrid Quantum-Classical Pipelines . . . 50

3 Hybrid Quantum Optimization on Neutral Atom Platforms 53
3.1 Hybrid Quantum-Classical Optimization . . . 54
3.1.1 Analog Quantum Evolution under Rydberg Blockade . . . 55
3.2 Application I: PCI Optimization in Wireless Networks . . . 57
3.2.1 Solving the Maximum Independent Set with Neutral-Atom QPUs 58
3.2.2 Iterative MIS-Based Coloring . . . 60
3.2.3 The BBQ-MIS algorithm for Vertex Coloring . . . 61
3.2.4 Comparison between Greedy-it-MIS and BBQ-mIS . . . 67
3.2.5 Quantum-Classical Branch-and-Price for Vertex Coloring . . . 69
3.2.6 Comparison of Hybrid Quantum-Classical Methods for Vertex Coloring . . . 77
3.3 Application II: Space–Terrestrial Network Design . . . 86
3.3.1 Combinatorial Structure of Satellite–Ground Integration . . . 87
3.3.2 Solving MWIS with Aquila . . . 91
3.3.3 Performance Evaluation . . . 93
3.4 Discussion and Perspectives on Hybrid Quantum-Classical Optimization 97

4 Hybrid Quantum Machine Learning on Neutral Atom Platforms 99
4.1 Neutral Atom Dynamics for Hybrid Learning . . . 100
4.1.1 Rydberg Interactions as High-Dimensional Feature Maps . . . 101
4.1.2 Analog Quantum Reservoir Computing on Neutral-Atom Platforms . . . 102
4.2 Application I: Protein Graph Classification . . . 103
4.2.1 Protein Graph Representation and Dataset Preparation . . . 104
4.2.2 Quantum Kernel Learning on Neutral-Atom Hardware . . . 107
4.2.3 Results on Quantum Kernel Learning . . . 110
4.3 Application II: Credit Risk Prediction . . . 118
4.3.1 Quantum Reservoir for Credit Card Default Classification . . . 119
4.3.2 Preprocessing . . . 121
4.3.3 Reservoir Encoding and Feature Extraction . . . 124
4.3.4 Classification . . . 127
4.3.5 Performance Evaluation and Hardware Validation . . . 131
4.4 Discussion and Perspectives on Hybrid Quantum Machine Learning . . 138

5 Conclusion 141
5.1 Summary of Contributions . . . 141
5.2 Embedding as a Key Part of Quantum Pipelines . . . 142
5.3 Quantum-Classical Hybridization as a Paradigm . . . 143

5.4 Limitations and Open Challenges 144
5.5 Future Directions 145
5.6 Closing Remarks 146

Bibliography **147**

List of Tables

2.1 Antenna-dataset embeddings. All distances are in μm. 42
2.2 Structural properties and optimization performance for the QUBO protein-folding instances. 46
2.3 Embedding distance statistics before (^{0}) and after (^{f}) optimization for the DEN model. All distances are in μm. 47
3.1 Results on larger instances using classical MIS heuristics. 68
3.2 Percentage of benchmark instances solved to optimality by each method (QCBP, BBQ-mIS, and HCG), aggregated across the full evaluation dataset. 79
3.3 Average relative optimality gap (%) achieved by each method (QCBP, BBQ-mIS, and HCG), disaggregated by graph size n. Lower values indicate solutions closer to the certified optimum. 79
3.4 Median number of ILP fallback calls in QCBP, disaggregated by graph size n and instance type (UD vs. non-UD). 82
3.5 Geometric constraints of the Aquila QPU register. 91
4.1 Graph size statistics of the PROTEINS dataset, highlighting the high variability in instance scale. 105
4.2 Composition of the PROTEINS12 and PROTEINS256 subsets after unit-disk filtering, reporting the number of instances per class: enzyme (Class 1) vs. non-enzyme (Class 2). 107
4.3 Classification performance on PROTEINS12 for the baseline quantum kernel Λ_a, the Bayesian-optimized kernel Λ_{BO}, and the classical SPK, evaluated across four standard metrics. 115
4.4 Classification performance on PROTEINS12 for emulation and Aquila hardware execution under Λ_a and Λ_{BO}, alongside the classical SPK baseline. 117
4.5 Classification performance on PROTEINS256, comparing the Aquila hardware execution of Λ_{BO} against the classical SPK baseline. 117
4.6 Sensitivity of classification performance on PROTEINS256 to the number of measurement shots, evaluated under Λ_{BO} on Aquila. 118
4.7 Feature groupings found by agglomerative clustering on CARDS_30000. 124
4.8 Training class distribution by resampling strategy for CARDS_30000. . 128
4.9 Training class distribution by resampling strategy for CARDS_2500. . . 128

4.10 Training class distribution by resampling strategy for CARDS_1000. . . 128
4.11 Hyperparameter configurations 130
4.12 Training time comparison (in seconds) across the evaluated classifiers on CARDS_30000 under various resampling strategies. The DNN is trained directly on the original 12 features, whereas the remaining classifiers operate on the feature set produced by the QRC (detuning encoding). Consistent with expectations, oversampling methods lead to longer training times, while undersampling reduces them. 133

List of Figures

2.1 A two-dimensional array of neutral atom qubits in which only atoms separated by less than the Rydberg blockade radius interact. 15
2.2 Embedding of a graph into a neutral-atom quantum register, where the adjacency matrix in Fig. 2.2a defines the target interaction pattern. Not all spatial configurations reproduce the desired connectivity. 19
2.3 Geometric constraints on feasible embeddings: the maximum clique K_7 within the interaction radius (left), and the maximum number of neighbors under hexagonal packing with $D_{\text{min}} = 4\ \mu m$ and $D_{\text{adj}} \approx 10.26\ \mu m$ (right). 22
2.4 Neural optimization framework for constrained graph embedding on neutral-atom registers. 27
2.5 Architecture of the Distance Encoder Network. An autoencoder transforms input coordinates into new ones; a fixed-weight distance encoder then computes pairwise distances used by the loss function. Example shown for a 5-vertex graph. 30
2.6 Modified DEN architecture for embedding on the Aquila QPU. The model computes squared pairwise and vertical distances to enforce the row-spacing constraint. Example for a 5-vertex graph. 34
2.7 Row-spacing penalty function ELF_{row}: zero when vertices share the same row or when their vertical separation exceeds $4\ \mu m$. 34
2.8 Scaling of computational time and first feasibility epoch with graph size and embedding dimensionality for the DEN model. 40
2.9 Feasibility rate across the synthetic graph dataset for $N = 2$ and $N = 3$: the DEN model outperforms *Ipopt*, while Fruchterman–Reingold and scaling initializations perform comparably. 41
2.10 Largest adjacency-gap values obtained by each solver on the synthetic dataset, grouped by graph size n (blue rectangles). 41
2.11 Embedding of sample $19G_1cc$ onto the Orion Alpha register. 43
2.12 Aquila-feasibility rate of the DEN model, grouped by number of vertices n. Bin labels indicate the total number of graph samples.. 44
2.13 Gap $d_{\overline{\text{adj}}} - d_{\text{adj}}$ distributions grouped by n for the PROTEINS_12 and PROTEINS_16 datasets. 44

2.14 DEN embedding of sample 115 from the PROTEINS_256 dataset. 45
2.15 Embedding of the QUBO protein-folding instance $\mathcal{G}_{17_7}$: Fruchterman–Reingold initialization (a), feasible 2D embedding (b), and loss function over epochs (c). 47
2.16 Embedding of the full MIS antenna problem in 3D: initial positions with $z = 0$ (a) and resulting feasible 3D embedding (b). 48
2.17 3D QUBO embedding for vertex coloring on $\mathcal{G}_{GC}$: original graph (left), initial qubit configuration (center), and feasible embedding after optimization (right). 49
3.1 Branch-and-Bound tree structure, pruning strategy, and exploration policy underlying the BBQ-mIS algorithm. 64
3.2 VC performance of *Greedy-it-MIS* and BBQ-mIS on all graph instances, with *Gurobi* as the optimal reference and *Pulser* as the MIS solver. . . . 68
3.3 Workflow of the hybrid classical-quantum column generation procedure [92]. 74
3.4 Hybrid Quantum-Classical Branch-and-Price Algorithm (QCBP). 76
3.5 QPU shot consumption across methods, compared on UD and non-UD graph instances. 80
3.6 Branch-and-Bound node distributions grouped by graph size n. 81
3.7 Proportion of improving columns generated by the Orion Alpha QPU as a function of PSP size n'. 83
3.8 Percentage of maximal independent sets found through PSP solutions via QPU, grouped by PSP size n'. 84
3.9 Sampled bitstring frequencies and associated reduced costs across four HCG iterations on the Orion Alpha QPU. 85
3.10 Final coloring of a 10-vertex VC instance solved with QCBP on the Orion Alpha QPU. 86
3.11 Adiabatic pulse schedule on Aquila for a MWIS instance on a 3-vertex graph. Channel *Ch[0]* broadcasts the global drive, comprising the Rabi frequency $\Omega(t)$ and the global detuning $\delta(t)$. Channel *Ch[1]* encodes the local detuning $\delta_i^{\text{loc}}(t)$, with site-specific amplitudes reflected by the color-coded qubit assignments. 93
3.12 Per-instance performance gap of QAA relative to the greedy heuristic (orange) and the exact solver (blue), with bootstrap confidence regions obtained by subsampling 100 out of 300 measurements. 95
3.13 Number of non-independent bitstrings as a function of graph size $|\mathcal{V}|$, illustrating the degradation in quantum encoding fidelity for larger and harder-to-embed instances. 96
3.14 Downstream impact of SSP solution quality on GSP and SAP objective distributions, comparing pipelines seeded with QAA versus exact solver outputs. 96

4.1 Distribution of graph sizes in the PROTEINS dataset, in terms of node count (left) and edge count (right). 105
4.2 Original non-UD graph (left) and its UD embedding (right) produced by the adapted DEN model, with the row-discretized atom layout imposed by the Aquila architecture. 107
4.3 Schematic of the layered time evolution, depicting the successive application of each Hamiltonian and the associated parameters Λ. 108
4.4 Hybrid quantum-classical workflow for Bayesian optimization of the kernel parameters Λ, combining classical emulation with a Gaussian process surrogate model. 111
4.5 Sampled energy distributions produced by the quantum kernel evolution for representative enzyme and non-enzyme graph instances from PROTEINS12. 112
4.6 Kernel matrix of PROTEINS12 dataset, with $\mu = 1$. 113
4.7 Control waveforms generated by the baseline and Bayesian-optimized parameter sets, illustrating the effect of optimization on the pulse shape. 114
4.8 Energy distributions induced by the baseline parameter set Λ_a and the Bayesian-optimized set Λ_{BO}, highlighting the shift in the sampled distribution following optimization. 114
4.9 Energy distributions for a representative PROTEINS12 instance, contrasting noiseless classical emulation with hardware execution on Aquila under Λ_{BO}. 116
4.10 Hybrid quantum-classical pipeline for credit risk prediction. 120
4.11 PCA scree plot for CARDS_30000, reporting the cumulative variance retained as a function of the number of retained components. 122
4.12 Correlation circle of feature loadings on the first two principal components for CARDS_30000, with colors denoting cluster membership. . . 123
4.13 Illustrative pulse sequences for position encoding (a) and local detuning encoding (b), with the latter displayed on a reduced qubit register for visual clarity. 126
4.14 F1 score vs. number of Hamiltonian simulations, with dotted lines indicating noiseless statevector emulation results as a reference baseline. . 132
4.15 Classification benchmark on CARDS_30000, comparing emulated QRC (detuning encoding), CRC, and FC against the DNN baseline. Shaded area denotes the DNN standard deviation; the red line marks the peak performance across all classifiers. 133
4.16 Comparison of register encodings: (a) column, (b) hook-shaped, (c) replicated column, (d) replicated hooks. 134
4.17 Pearson correlation between Aquila QRC outputs and shot-based emulation as a function of shot count and timestep. 135

4.18 Classification benchmark on CARDS_1000, contrasting Aquila hardware and emulated QRC under detuning and position encoding, CRC, and FC against the DNN baseline. 136
4.19 QRC feature space on CARDS_2500 under detuning encoding, contrasting Aquila hardware outputs with statevector (SV) and shot-based emulation. 137
4.20 Timestep-resolved QRC features for a representative CARDS_2500 sample, illustrating the progressive divergence between Aquila hardware and noiseless emulation due to decoherence. 137
4.21 Classification benchmark on CARDS_2500, contrasting Aquila hardware and emulated QRC (detuning encoding), CRC, and FC against the DNN baseline. 138

List of Acronyms

BB Branch-and-Bound

BP Branch-and-Price

CC Cluster Centroids

CG Column Generation

CRC Classical Reservoir Computing

DEN Distance Encoder Network

DNN Deep Neural Network

ELF Embedding Loss Function

FC Fully Classical

GSP Gateway Selection Problem

HCG Hybrid Column Generation

ILP Integer Linear Program

IS Independent Set

KNN K-Nearest Neighbours

K-SMOTE K-Means SMOTE

LB lower bound

LEO Low Earth Orbit

MILP Mixed-Integer Linear Program

MIP Mixed-Integer Program

MIS Maximum Independent Set

mIS maximal Independent Set

ML Machine Learning

MWIS Maximum-Weighted Independent Set

NB Naïve Bayes

NISQ Noisy Intermediate-Scale Quantum

NN Neural Network

PCA Principal Component Analysis

PCI Principal Cell Identifier

PSP Pricinig Sub-Problem

QAOA Quantum Approximate Optimization Algorithm

QAA Quantum Adiabatic Algorithm

QCBP Quantum-Classical Branch-and-Price

QEK Quantum Evolution Kernel

QML Quantum Machine Learning

QPU Quantum Processing Unit

QRC Quantum Reservoir Computing

QUBO Quadratic Unconstrained Binary Optimization

RC Reservoir Computing

RMP Reduced Master Problem

RNN Recurrent Neural Network

SAP Spectrum Assignment Problem

SGD Stochastic Gradient Descent

SMOTE Synthetic Minority Over-sampling Technique

SPK Shortest Path Kernel

SSP Satellite Selection Problem

STIN Space-Terrestrial Integrated Network

SVM Support Vector Machine

UB Upper Bound

UD unit-disk

UDG Unit-Disk Graph

VC Vertex Coloring

Chapter 1

Introduction

In recent years, quantum computing has attracted increasing attention as a promising paradigm capable of addressing computational problems that are intractable for classical machines. In particular, quantum algorithms have been proposed for applications such as material simulation [1, 2, 3], combinatorial optimization [4, 5, 6], and data analysis [7, 8], where potential computational advantages have been theoretically established.

Despite these advances, a significant gap persists between the formal development of quantum algorithms and their practical deployment. A large portion of the literature assumes idealized models of computation, relying on fault-tolerant quantum hardware with a large number of qubits and negligible noise. Under these assumptions, quantum algorithms can demonstrate remarkable theoretical speedups. However, current quantum devices operate in the so-called NISQ (Noisy Intermediate-Scale Quantum) regime [9], where limited qubit counts, short coherence times, and imperfect control severely restrict the set of feasible computations. As a result, many theoretically well-founded algorithms remain out of reach for implementation on available hardware [10].

Beyond hardware limitations, an additional challenge arises when attempting to translate quantum algorithms into real-world applications. Even when a quantum procedure is implementable, mapping it to a meaningful problem often requires non-trivial design choices, including data encoding, representation, and the definition of suitable computational primitives. This issue is particularly evident in data-driven contexts, where the way classical information is embedded into quantum systems has a direct impact on the effectiveness of the approach [11]. More broadly, the difficulty of connecting abstract quantum models with practical problem settings remains one of the main obstacles to the adoption of quantum computing technologies [12].

From an application perspective, this disconnect is critical. Industrial use cases require solutions that are robust, scalable, and compatible with existing computational workflows. However, quantum algorithms are often designed without explicitly considering these constraints, resulting in approaches that are either impractical on current hardware or difficult to integrate into real-world pipelines. Bridging this gap therefore

requires a shift in perspective, from developing algorithms under ideal assumptions to designing methodologies that explicitly account for hardware limitations and application requirements.

In this context, hybrid quantum-classical approaches have emerged as a natural strategy to cope with the constraints of NISQ devices. By combining quantum evolutions with classical methods, these algorithms aim to extract useful computational features from quantum systems while delegating learning and decision-making tasks to their classical counterpart [13]. This paradigm enables the exploitation of quantum resources within practical settings, where noise, limited system size, and imperfect control must be explicitly taken into account.

1.1 Hybrid Quantum-Classical Algorithms

Hybrid quantum-classical algorithms are computational frameworks in which quantum and classical resources are combined within a single workflow. Rather than aiming for a fully quantum solution, these approaches decompose a task into subcomponents that can be executed on different computational paradigms, assigning to each the operations for which it is best suited. In practice, this typically results in a pipeline where a quantum processor is used to perform specific transformations—such as state preparation, evolution, or feature generation—while classical routines handle tasks such as preprocessing, optimization, and decision-making. The interaction between the two components can take different forms, ranging from iterative feedback loops, as in variational algorithms, to sequential pipelines, where quantum outputs are post-processed classically.

This hybridization is motivated not only by hardware limitations, but also by the structure of real-world problems. Many applications do not admit a direct quantum formulation, and require intermediate steps such as data encoding, dimensionality reduction, or integration with existing models. By embedding quantum components within a classical framework, hybrid algorithms provide the flexibility needed to address these challenges while still leveraging the distinctive properties of quantum systems.

A key feature of hybrid approaches is the separation between quantum-driven transformations and classical learning or optimization. Quantum systems can generate complex, high-dimensional representations through their intrinsic dynamics, but extracting useful information from these representations is often difficult due to noise and measurement constraints. Classical algorithms complement this process by providing robust and efficient tools to process, interpret, and exploit the generated features.

As a result, hybrid quantum-classical algorithms represent a shift in perspective: instead of pursuing purely quantum speedups under ideal assumptions, they aim to integrate quantum resources into practical computational workflows, enabling the extraction of useful information from imperfect, intermediate-scale devices.

1.1.1 Variational and Iterative Hybrid Frameworks

A prominent class of hybrid algorithms is based on variational and iterative schemes, where a quantum system with tunable parameters is embedded within a classical optimization loop [13, 14]. These methods exploit the fact that, while current quantum hardware cannot support deep and fault-tolerant computations, it can still implement controlled evolutions whose behavior can be adjusted through a small number of parameters.

In this setting, the quantum device is used to generate a state or a set of measurements that depend on a configurable set of parameters. These parameters determine how the system evolves, for instance through gate sequences in digital platforms or through control fields in analog devices. The quality of a given parameter configuration is evaluated by measuring the quantum system and computing a cost function that reflects the objective of the problem. The optimization process is inherently iterative. At each step, the quantum processor is queried to evaluate the current configuration, while a classical optimizer updates the parameters based on the observed outcomes. This creates a feedback loop in which quantum and classical computations are tightly interleaved. The overall performance of the algorithm depends not only on the expressivity of the quantum model, but also on the efficiency of the classical optimization and on the reliability of the measurement process.

This paradigm has been explored across several domains. In optimization, hybrid methods have been applied to problems such as portfolio optimization and scheduling, including implementations on quantum annealers [15]. In parallel, significant research has focused on improving optimization strategies tailored to parameterized quantum models, for instance by reducing the number of evaluations or improving convergence properties [16, 17].

In machine learning, variational frameworks are used to construct models in which quantum systems act as nonlinear transformations of the input data, supporting tasks such as classification and time-series prediction [18]. Similarly, in quantum chemistry and materials science, these approaches are employed to approximate ground states of molecular systems by combining quantum state preparation with classical optimization routines [19].

Despite their versatility, variational and iterative hybrid algorithms present several challenges. The optimization landscape can be highly complex and affected by noise, leading to slow convergence or sensitivity to initialization. Moreover, the need to estimate quantities through repeated measurements introduces statistical fluctuations, which can degrade the stability of the optimization process. These aspects highlight the importance of carefully designing both the quantum model and the classical optimization strategy when targeting practical implementations on NISQ devices.

1.1.2 Hardware-Aware Algorithm Design

The effectiveness of hybrid algorithms is tightly linked to the characteristics of the underlying hardware. Quantum devices differ significantly in terms of connectivity, control, and available interactions, and these differences directly influence the class of problems that can be efficiently implemented.

This variability is particularly evident across hardware platforms, such as superconducting qubits, trapped ions, and neutral atoms, each of which supports different computational primitives. Even within the same technology, implementations can vary substantially depending on the provider, leading to different constraints on system size, geometry, and control precision.

Neutral-atom platforms provide a clear example of this interplay between algorithm design and hardware constraints. These systems can implement analog quantum evolutions governed by programmable Hamiltonians, where computation is encoded into interaction patterns and control parameters. This requires a fundamentally different approach compared to gate-based quantum computing, as the problem must be adapted to the physical interactions supported by the device.

As a consequence, the design of hybrid quantum algorithms cannot be decoupled from the hardware on which they are executed. Effective solutions must account for device-specific constraints, including geometry, noise sources, and control limitations. This hardware-aware perspective is essential for moving from abstract algorithm design to practical implementations on NISQ devices.

1.2 Noisy Intermediate-Scale Quantum Devices

The expression NISQ refers to quantum processors that operate with a number of physical qubits large enough to move beyond proof-of-principle experiments, but still far from the fault-tolerant regime required by fully error-corrected quantum computation. In practice, these devices are characterized by limited coherence times, imperfect gate or control fidelities, restricted connectivity or geometry, and a strong dependence of algorithmic performance on hardware noise and compilation overhead. For this reason, the term does not identify a single technology, but rather a broad technological stage shared by several different hardware platforms [9, 10].

From an algorithmic perspective, the NISQ regime has two main consequences. First, circuit depth and execution time must remain limited, since noise accumulates quickly and can erase any computational advantage. Second, the architecture of the hardware becomes a central design variable: qubit connectivity, measurement model, native interactions, and control modalities all strongly affect which problems can be implemented efficiently. As a result, a meaningful overview of NISQ computing benefits from being organized by hardware platform rather than by abstract algorithmic model alone [10].

The current technological landscape can be grouped into a few major families: superconducting circuits, trapped ions, neutral atoms, photonic processors, semiconductor spin qubits, and quantum annealers. These platforms differ substantially in how qubits are encoded, how interactions are generated, and what type of scaling they currently support.

Superconducting quantum processors encode qubits in nonlinear microwave circuits fabricated using lithographic techniques and operated at millikelvin temperatures. In these systems, the qubit is typically realized as an anharmonic oscillator (e.g., a transmon), where two energy levels are isolated and used as the computational basis states. Quantum operations are implemented through microwave pulses that drive transitions between these states, while coupling elements enable two-qubit gates between neighboring circuits. A key advantage of this platform is its compatibility with established semiconductor manufacturing processes, which has enabled rapid scaling in terms of qubit count and integration with classical control electronics. In addition, gate operations are extremely fast (on the order of tens of nanoseconds), allowing many operations to be performed within the coherence time of the qubits. These features have made superconducting circuits one of the leading platforms for gate-based NISQ computing [20]. However, these benefits come with important limitations. Qubits are typically arranged on a chip with fixed, local connectivity, meaning that interactions are naturally restricted to nearest neighbors and must be extended via additional routing operations. Moreover, coherence times remain limited (typically in the microsecond range), which constrains the depth of executable circuits. The system also requires complex cryogenic infrastructure and frequent calibration to mitigate effects such as crosstalk and parameter drift. As a result, achieving reliable performance often depends on careful error mitigation and hardware-aware compilation strategies [20].

Trapped-ion quantum computers encode qubits in the internal electronic states of ions confined in electromagnetic traps, typically using radiofrequency (Paul) or Penning trap configurations. The ions are suspended in vacuum and manipulated using precisely controlled laser pulses, which drive transitions between the qubit states and enable coherent control. Single-qubit operations are implemented through resonant laser interactions, while entangling gates are achieved by coupling the internal states of the ions to their collective motional modes. In this way, the vibrational motion of the ion chain acts as a shared quantum bus, allowing interactions between any pair of ions in the trap. This mechanism provides an effective all-to-all connectivity, which significantly simplifies the implementation of quantum circuits compared to architectures with local connectivity. A key advantage of trapped-ion systems is the high degree of control over the qubits, resulting in very long coherence times (ranging from milliseconds to seconds) and high-fidelity gate operations. In addition, ions are naturally identical particles, which leads to excellent qubit uniformity and reduces device variability. These characteristics have made trapped ions one of the most precise platforms for gate-based NISQ computing [21]. However, these benefits come at the cost of slower operation speeds, since laser-driven gates are typically orders of magnitude slower than

microwave-based gates in superconducting systems. The reliance on complex optical setups also increases experimental overhead and limits scalability. As the number of ions in a single trap increases, controlling and addressing individual qubits becomes more challenging, and the collective motional modes become harder to manage. For this reason, current scaling strategies often rely on modular architectures, where multiple ion traps are interconnected through photonic links or ion shuttling techniques [21].

Neutral-atom processors encode qubits in individual atoms held in optical tweezers and manipulated through laser-driven transitions, often exploiting Rydberg interactions to generate entanglement. This platform is particularly attractive because atoms are naturally identical, arrays can be reconfigured geometrically, and the interaction graph can often be shaped in a hardware-native manner. As a consequence, neutral atoms are especially well suited to analog quantum simulation and to digital-analog approaches where geometry and interaction range are part of the computation itself [22, 23]. The strengths of neutral atoms are geometric programmability, promising scalability, and a natural suitability for graph-structured Hamiltonians. Their weaknesses include sensitivity to atom loss and control noise, device-specific geometric constraints, and the need for careful problem-to-hardware mapping [22, 23].

Photonic quantum computing uses photons as carriers of quantum information, with qubits typically encoded in degrees of freedom such as polarization, or alternatively using continuous-variable states of the electromagnetic field. Quantum operations are implemented through linear optical elements (e.g., beam splitters and phase shifters), together with measurement-induced nonlinearities, where detection events are used to probabilistically generate entanglement and implement gates. A distinctive feature of this platform is that photons interact very weakly with the environment, which results in extremely low decoherence and allows quantum information to be preserved over long distances. This makes photonic systems naturally suited for quantum communication and distributed quantum computing. Moreover, photonic circuits can often operate at or near room temperature and can be integrated using optical and semiconductor fabrication technologies, enabling the development of scalable and potentially networked architectures [24]. However, the weak interaction between photons also represents a fundamental limitation. In many architectures, two-qubit gates are inherently probabilistic and rely on post-selection or auxiliary resources, which significantly increases the overhead required for scalable computation. In addition, high-performance operation depends on the availability of efficient single-photon sources and detectors, which remain technologically demanding. As a result, current photonic quantum processors are often better suited to specialized tasks, such as sampling or communication protocols, rather than general-purpose NISQ computation [24].

Semiconductor spin qubits encode quantum information in the spin degree of freedom of electrons or holes confined in semiconductor nanostructures, such as quantum dots. These devices are typically fabricated using silicon or germanium heterostructures, where electrostatic gates are used to trap and control individual charge carriers.

The two spin states of the electron (spin-up and spin-down) form the computational basis, while quantum operations are implemented using magnetic or electric control fields, often through electron spin resonance or spin-orbit coupling mechanisms. A key motivation for this platform is its compatibility with existing semiconductor manufacturing technologies. In principle, spin qubits can be integrated using processes similar to those employed in conventional microelectronics, enabling the possibility of very high qubit densities and large-scale integration on a chip. Moreover, spins in semiconductors can exhibit relatively long coherence times, especially in isotopically purified materials, which reduces the impact of environmental noise [25]. However, the practical realization of large-scale spin-qubit processors remains challenging. Device performance is highly sensitive to fabrication imperfections, variability in the electrostatic environment, and charge noise, all of which can affect qubit stability and reproducibility. In addition, coupling and controlling multiple qubits in a scalable and uniform manner is still an open problem. As a result, despite their strong long-term potential, semiconductor spin qubits are currently at an earlier stage of development compared to more mature NISQ platforms [25].

Quantum annealers are specialized quantum devices designed to solve optimization problems by exploiting the physics of energy minimization. In these systems, a computational problem is first mapped onto an Ising or QUBO (Quadratic Unconstrained Binary Optimization) Hamiltonian, where the solution corresponds to the ground state of the system. The device is then initialized in a simple quantum state and slowly evolved according to a time-dependent Hamiltonian, ideally remaining close to its instantaneous ground state throughout the process. If the evolution is sufficiently adiabatic, the final configuration encodes a low-energy (and potentially optimal) solution to the original problem [26]. Unlike gate-based quantum computers, annealers do not implement arbitrary quantum circuits. Instead, they rely on a fixed hardware graph, where qubits represent binary variables and couplers encode pairwise interactions. This makes the mapping from the target problem to the hardware a central step, often requiring embedding techniques to fit the problem graph onto the device connectivity. The main industrial realization of this approach is provided by *D-Wave*, whose systems feature thousands of qubits arranged in sparse but structured connectivity graphs. The primary advantage of quantum annealers is their ability to handle large-scale optimization instances in a hardware-native way, making them attractive for certain classes of combinatorial problems. However, this comes at the cost of limited generality: only problems that can be expressed in quadratic binary form are directly supported, and the absence of full programmability prevents their use as universal quantum computers. In addition, performance is sensitive to noise, embedding overhead, and the specifics of the annealing schedule, which can limit the quality of the obtained solutions in practice [26].

1.2.1 Comparative Perspective

From a practical point of view, no single NISQ platform is dominant across all criteria. Each platform offers distinct advantages in terms of scalability, coherence, connectivity, or programmability, but also presents specific limitations that constrain its applicability [9, 10].

For the purposes of this thesis, this diversity directly motivates a hardware-aware view of algorithm design: the practical value of a quantum approach depends not only on the formal correctness of the algorithm, but also on how naturally the target problem can be represented on a specific device. This point is particularly important for neutral-atom systems, where geometry, interaction range, and control modalities are part of the computational model itself.

1.2.2 Neutral Atom Quantum Processors

Among the different NISQ platforms, this thesis focuses on neutral-atom quantum processors, and in particular on their analog computational mode. These systems use individual neutral atoms (typically ^{87}Rb) trapped in optical tweezers and arranged in programmable two- or three-dimensional arrays. Each atom encodes a qubit in two internal states: the ground state $|g\rangle$ and a highly excited Rydberg state $|r\rangle$.

Quantum dynamics are induced by laser pulses that drive transitions between $|g\rangle$ and $|r\rangle$, while interactions between atoms arise from strong dipole-dipole (Van der Waals) interactions when atoms are excited to the Rydberg state [27, 28]. In contrast to gate-based approaches, the focus of this work is on analog quantum processing, where the computation is defined by the continuous-time evolution of the system under a programmable Hamiltonian.

Considering a system of n atoms with positions $overrightarrowp_i$, driven by laser fields characterized by a Rabi frequency Ω and detuning δ, the system evolves according to the Hamiltonian

$$\mathcal{H} = \frac{\Omega\hbar}{2}\sum_{i=1}^{n}(|g_i\rangle\langle r_i| + |r_i\rangle\langle g_i|) - \delta\hbar\sum_{i=1}^{n}\hat{n}_i + \sum_{i<j}V_{ij} \tag{1.1}$$

$$V_{ij} = C_6\frac{\hat{n}_i\hat{n}_j}{\|\vec{p}_i - \vec{p}_j\|^6} \tag{1.2}$$

$$\hat{n}_i = |r_i\rangle\langle r_i| \tag{1.3}$$

where $\hat{n}_i$ counts the population in the Rydberg state and C_6 is the interaction coefficient, determined by the chosen Rydberg level (e.g., $C_6 = 862690 \times 2\pi$ MHz μm^6 for $|r\rangle = |70S_{1/2}\rangle$ in ^{87}Rb [29]).

The interaction term V_{ij} decays rapidly with distance, which gives rise to a key physical mechanism known as the *Rydberg blockade*. This effect can be characterized

by the blockade radius

$$r_b = \left(\frac{C_6}{\hbar \sqrt{\Omega^2 + \delta^2}} \right)^{1/6}, \tag{1.4}$$

which defines a threshold distance below which two atoms cannot be simultaneously excited to the state $|r\rangle$. When $|\vec{p}_i - \vec{p}_j| < r_b$, the doubly excited state $|r_i r_j\rangle$ is energetically suppressed, effectively inducing a strong, distance-dependent constraint on the system dynamics [30, 31]. This blockade mechanism provides a natural way to encode interactions and constraints directly into the geometry of the atomic array. In particular, by arranging atoms in space and tuning the laser parameters (Ω, δ), it is possible to engineer effective spin Hamiltonians, such as Ising models, where interactions are defined by the distances between atoms. Since many combinatorial optimization problems can be mapped to Ising or QUBO formulations [32, 33], neutral-atom systems offer a direct physical implementation of such problems.

From a computational perspective, this means that the problem is not executed through a sequence of discrete gates, but rather encoded into the Hamiltonian itself. The system evolution then explores the corresponding energy landscape, enabling the extraction of solutions or features through measurement. This property makes neutral-atom platforms particularly suitable for optimization, simulation, and machine learning tasks involving graph-structured data, where geometry and interactions play a central role. More generally, the programmability of atom positions [34] and the distance-dependent connectivity allow one to map graphs onto the quantum register by associating vertices to atoms and defining edges through spatial proximity, i.e., whenever $|\vec{p}_i - \vec{p}_j| < r_b$. This tight interplay between geometry and computation is one of the defining features of neutral-atom quantum processors and motivates their use throughout this thesis.

1.3 Thesis Contributions and Organization

This thesis investigates how quantum computing can be brought closer to practical and industrial applications, with a particular focus on analog neutral-atom quantum processors in the NISQ regime. The work is structured around three main research directions, which together address different aspects of the pipeline required to translate real-world problems into executable quantum workflows.

The first contribution, presented in Chapter 2, focuses on *embedding methodologies*, namely the problem of mapping application-specific instances onto the physical constraints of neutral-atom hardware. This includes the development of techniques to encode graph structures and optimization problems into spatial configurations and control parameters of the quantum system. The results build upon a series of works, including

- *Neural-powered unit disk graph embedding: qubits connectivity for some QUBO problems* [35],

- *Neural optimization for quantum architectures: graph embedding problems with Distance Encoder Networks* [36],
- *Harnessing DEN models for quantum computing tasks on neutral atom QPUs* [37],

where increasingly advanced embedding strategies are proposed to improve scalability and hardware compatibility.

The second part of the thesis, detailed in Chapter 3, addresses *quantum optimization* on neutral-atom platforms. Given the natural ability of these systems to represent graphs through spatially dependent interactions, the focus is on combinatorial graph optimization problems. In particular, the work explores progressively more sophisticated approaches to the vertex coloring problem, in connection to the PCI (Principal Cell Identifier) assignment problem. Beyond vertex coloring, this research direction also investigates formulations based on the MWIS (Maximum-Weighted Independent Set), which arise naturally in satellite-terrestrial network optimization tasks. In particular, within the NGI Sargasso QRISTIN project, MWIS-based models are employed to address optimization problems in STINs (Space-Terrestrial Integrated Networks), where resource allocation and interference management can be cast into graph-based formulations suitable for quantum processing. The main references include

- *Towards Optimal Graph Coloring Using Rydberg Atoms* [38],
- *BBQ-mIS: a parallel quantum algorithm for graph col- oring problems* [39],
- *Hybrid Quantum-Classical Branch-and-Price Method for the Vertex Coloring Problem* [40],
- *Quantum-Assisted Design of Space-Terrestrial Inte- grated Networks* [41].

The third research direction, presented in Chapter 4, focuses on *quantum machine learning*. This part of the thesis explores how neutral-atom quantum processors can be used to extract features and support learning tasks within hybrid quantum-classical pipelines. The first application considers graph classification using quantum kernels, with a specific focus on distinguishing protein structures represented as graphs. The second application extends beyond graph-structured data, investigating credit card default prediction through a quantum reservoir computing framework, where the quantum system acts as a nonlinear feature generator. The main references for this chapter are

- *Harnessing a 256-qubit Neutral Atom Simulator for Graph Classification* [42],
- *Quantum Reservoir Computing for Credit Card Default Prediction on a Neutral Atom Platform* [43].

Finally, Chapter 5 summarizes the main findings of the thesis, highlighting the key contributions, practical insights, and limitations encountered when working with current quantum hardware. It also outlines potential directions for future research, with particular emphasis on improving scalability, robustness to noise, and the integration of quantum methods into real-world computational pipelines.

Chapter 2

Hardware-Aware Encoding for Hybrid Quantum Algorithms on Neutral-Atom Devices

A central challenge in the practical deployment of quantum algorithms concerns the representation of computational problems on physical quantum hardware. While many problems of interest can be expressed in compact mathematical forms, such as QUBO (Quadratic Unconstrained Binary Optimization) formulations [33], the translation of these abstractions into a form compatible with a given quantum processor is far from trivial. Each quantum computing platform imposes specific constraints on qubit connectivity, interaction strength, spatial arrangement, and operational primitives. As a consequence, the process of mapping logical variables and their interactions onto physical qubits requires carefully designed encoding strategies that explicitly account for the hardware characteristics.

This mapping procedure is referred to as *embedding*. In the broadest sense, embedding consists in translating the interaction graph of a computational problem into a structure that can be physically realized on the target quantum device. The quality of the embedding has a direct impact on the performance of quantum algorithms, since it determines the number of qubits required, the strength and locality of interactions, and the overall feasibility of the computation. For hybrid quantum classical algorithms in particular, where classical procedures interact repeatedly with the quantum processor, the embedding stage often becomes a critical component of the overall workflow.

Although the notion of embedding is common to most quantum computing technologies, its practical implementation strongly depends on the physical properties of the underlying hardware. Different platforms impose different connectivity patterns, interaction mechanisms, and architectural constraints, which in turn determine how logical variables and their couplings can be mapped onto physical qubits. As a result, the embedding strategy must be carefully tailored to the characteristics of the target

device. In many architectures, such as superconducting circuits, the connectivity between qubits is defined by fixed or partially reconfigurable interaction graphs. In these cases, the embedding process primarily focuses on adapting the structure of the computational problem to the predefined hardware topology, or on relying on dedicated transpilation strategies that transform abstract quantum circuits into hardware compliant implementations. These compilation procedures typically modify the circuit in order to respect the connectivity constraints between qubits and the specific gate set supported by the architecture. This process may involve the insertion of additional operations, such as swap gates used to move quantum information across the device, as well as the decomposition of gates into sequences of native hardware operations. As a consequence, the embedding and transpilation stages become tightly coupled, jointly ensuring that the logical description of the algorithm can be executed on the physical quantum processor while preserving, as much as possible, the intended computational structure.

Neutral atom quantum processors follow a different paradigm, in which the connectivity between qubits is not imposed by a fixed wiring pattern but instead emerges from the geometric arrangement of atoms in space. Since interactions are mediated through the Rydberg blockade mechanism [44], the presence or absence of an edge between two qubits depends directly on the distance between the corresponding atoms. Consequently, the structure of the interaction graph implemented by the hardware is intrinsically geometric and can be controlled by adjusting the spatial configuration of the qubits within the atomic register. This distinctive feature naturally leads to a representation of computational problems in terms of geometric graphs. In particular, many problems that can be expressed as interaction graphs may be mapped onto spatial configurations of atoms such that the induced connectivity matches the logical structure required by the algorithms. Within this framework, graph based formulations become especially relevant, as they provide a direct and intuitive bridge between abstract computational models and the geometric constraints imposed by the hardware.

Graph representations therefore play a particularly important role in this context. Many optimization, machine learning, and scientific computing problems can be naturally expressed as graphs in which vertices represent variables and edges encode pairwise interactions. Consequently, identifying effective strategies for representing graphs on neutral atoms quantum hardware is an essential step toward enabling practical applications of analog quantum computing.

More specifically, the connection between graph representations and analog neutral-atom QPUs (Quantum Processing Units) arises from the physical realization of qubits in these systems. Qubits are encoded in individual atoms trapped in optical tweezer arrays and manipulated through laser-driven transitions. Interactions between qubits are mediated by exciting atoms to highly excited Rydberg states $|r\rangle$. In particular, the Rydberg blockade mechanism described in Chapter 1 prevents two atoms located within the blockade radius r_b from being simultaneously excited [45]. This phenomenon induces a geometry-dependent interaction pattern among qubits: two qubits effectively interact

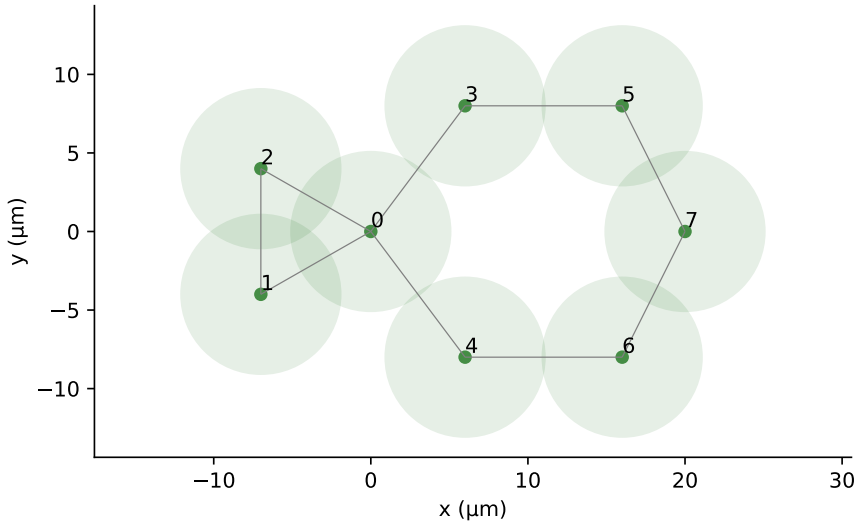


Figure 2.1: A two-dimensional array of neutral atom qubits in which only atoms separated by less than the Rydberg blockade radius interact.

whenever their spatial separation is smaller than r_b. From a graph-theoretical perspective, this interaction structure naturally corresponds to a UDG (Unit-Disk Graph). Unit disk graphs belong to the class of geometric graphs in which vertices represent points in the plane and edges connect pairs of vertices whose Euclidean distance is below a fixed threshold [46].

Definition 1 (Unit-disk graph)**.** *Consider n circles, with radius r, in the plane. The intersection graph [47] of these circles is a unit-disk graph with n vertexes where each vertex corresponds to a circle centre and the vertexes share an edge only if the corresponding circles intersect.*

In the context of neutral-atom devices, vertices correspond to atomic positions (i.e., qubits) in the register, while edges represent interactions induced by the Rydberg blockade. Figure 2.1 illustrates this concept. A set of atoms arranged in a two-dimensional array induces an interaction graph in which edges appear only between atoms whose distance is smaller than the blockade radius. In the figure, the circles drawn around each atom have radius equal to half of the blockade radius, so that overlaps between circles indicate interacting pairs. The resulting connectivity structure therefore follows the geometric constraints imposed by the physical interaction mechanism of the hardware.

This intrinsic geometric structure constitutes one of the most distinctive features of neutral atom quantum processors. On the one hand, it enables the direct physical implementation of several graph-based problems whose interaction structure can be expressed as a UDG. On the other hand, it introduces a nontrivial encoding challenge whenever the interaction graph of the problem does not naturally satisfy the unit-disk property. In such cases, it becomes necessary to determine a spatial arrangement of qubits whose induced adjacency pattern matches the target interaction graph.

Formally, the problem can be described as follows. Given a graph representing the interactions of a computational problem, the goal is to assign coordinates to each vertex such that the resulting geometric graph reproduces the desired adjacency relations under the unit distance constraint imposed by the hardware. While computing the adjacency matrix of a given set of coordinates is straightforward, the inverse problem of finding coordinates that realize a target graph is considerably more difficult.

The work presented in this chapter focuses on the development of hardware aware encoding strategies designed specifically for neutral atom quantum processors. The chapter summarizes the evolution of the embedding methodology that was progressively developed to address increasingly complex scenarios. These methods were designed with the goal of bridging the gap between abstract problem graphs and the geometric constraints imposed by neutral atom real-world hardware.

In particular, the approaches introduced throughout this line of work share several common objectives.

- Represent the interaction graph of a computational problem through spatial qubit configurations.
- Respect the unit-disk constraint imposed by the Rydberg blockade mechanism.
- Adapt the embedding to machine-specific constraints to enable practical execution of hybrid quantum-classical algorithms on real QPUs.

The effectiveness of the embedding strategy has been evaluated on a variety of application scenarios involving both artificial and real-world datasets, and targeting actual neutral atom quantum processors. In particular, we investigated the embedding of graphs onto the registers of two currently available quantum processing units, namely the Orion Alpha processor developed by PASQAL [48] and the Aquila processor developed by QuEra [29]. These devices impose different constraints in terms of register size, geometric layout, and interaction radius, making them representative testbeds for hardware aware encoding techniques.

These results illustrate the potential of hardware-aware embedding strategies to enable the execution of meaningful computational tasks on current neutral atom quantum hardware. More importantly, they demonstrate that the embedding stage can be treated as an independent research problem, whose solution significantly expands the range of applications that can be addressed on a given quantum platform. For this reason, the present chapter focuses on providing a coherent overview of the embedding methodologies developed throughout this research activity. The emphasis is placed on the conceptual evolution of the encoding strategies and on their adaptation to the constraints of neutral atom quantum processors.

2.1 Background and Related Approaches

Several strategies have been proposed in the literature to address the embedding of computational problems onto quantum hardware. A common direction consists in introducing auxiliary resources in order to emulate more flexible connectivity patterns between qubits. While these approaches can effectively increase the expressive power of the hardware, they typically require additional physical qubits or more sophisticated device configurations. This overhead can significantly limit the size of the problems that can be tackled on current NISQ (Noisy Intermediate-Scale Quantum) neutral atoms devices, where the number of available qubits remains a key constraint.

One line of research focuses on exploiting local field manipulations and engineered interaction structures to enhance qubit connectivity. For instance, in [49], the authors introduce an architecture in which local interactions are used to realize effective all-to-all connectivity between logical qubits. In this approach, logical variables are encoded through multiple physical qubits arranged on a square lattice, while local constraints enforce the consistency of the representation. Although this design significantly increases the connectivity available to the logical problem, it also introduces substantial overhead in terms of the number of required physical qubits.

Another approach is presented in [50], where the authors propose an embedding scheme in three dimensions relying on Ising ferromagnetic quantum wires that allow distant qubits to interact through mediated couplings. This design results in a programmable architecture arranged on a regular cubic lattice. However, the implementation of such quantum wires requires multiple physical qubits to represent each logical variable and relies heavily on the precise control of local fields to maintain the desired interaction structure. Further developments along this direction are discussed in [51], where Kim et al. investigate the use of quantum wires in Rydberg atom systems for solving maximum independent set problems. Their approach removes the need for local fields in certain configurations, but still requires auxiliary wire atoms arranged in a three-dimensional structure. Moreover, the resulting architecture constrains embeddings to configurations in which adjacent qubits are placed at fixed distances, effectively introducing a discretization of the embedding space and limiting the possibility of fully exploiting the continuous geometric flexibility offered by the Rydberg blockade mechanism.

An alternative strategy for embedding QUBO problems, particularly in the context of quantum annealing, relies on the concept of graph minors [52]. This approach is widely used in architectures characterized by fixed connectivity graphs, most notably in the D-Wave systems, where qubit interactions are implemented through programmable couplers. In this setting, the embedding problem reduces to determining whether the interaction graph of the QUBO instance can be represented as a minor of the hardware connectivity graph. To address this task in practice, heuristic algorithms have been developed to search for suitable graph minors without performing exhaustive exploration of the solution space. In particular, the D-Wave system employs probabilistic heuristic

techniques to efficiently identify minor embeddings for large problem instances [53]. A related approach was proposed by V. Choi [54], who investigated minor embedding strategies for representing Ising Hamiltonians within adiabatic quantum computation frameworks, with a focus on solving maximum independent set problems. In these methods, logical variables are mapped onto chains of physical qubits whose interactions are enforced through the appropriate tuning of qubit biases and coupler strengths, again introducing an overhead in the number of physical qubits.

While these strategies have proven effective for architectures characterized by fixed or engineered connectivity, they share a common limitation: the reliance on auxiliary qubits or redundant encodings to reproduce the desired interaction graph. This limitation becomes particularly critical in the context of neutral-atom QPUs, where the geometric nature of interactions offers a fundamentally different paradigm. Instead of adapting the logical problem to a rigid or augmented hardware topology, neutral-atom systems allow the interaction graph to be shaped directly through the spatial arrangement of qubits. This observation motivates the development of embedding techniques that exploit the continuous geometric degrees of freedom of the atomic register, while avoiding the overhead associated with auxiliary qubits. In this work, we adopt this perspective and focus on encoding strategies that aim to preserve a direct correspondence between logical variables and physical qubits, thereby maximizing the effective use of available quantum resources.

2.2 Hardware-Induced Connectivity in Neutral-Atom Quantum Processors

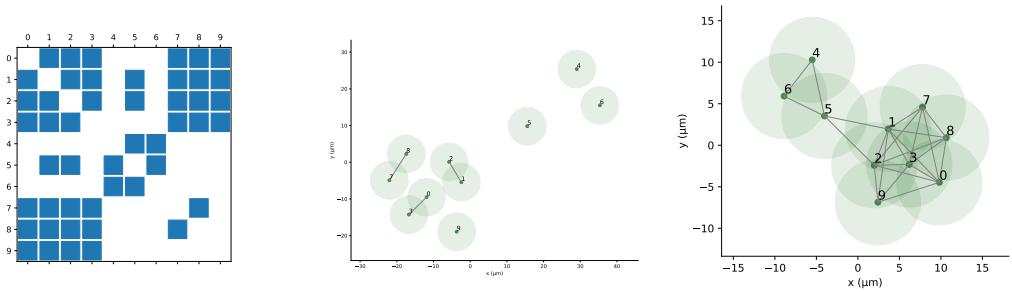
While the approaches discussed in the previous section rely on augmenting hardware connectivity through additional resources, in this work we adopt a different perspective tailored to neutral-atom quantum processors. In these systems, qubit interactions are not fixed nor artificially extended, but instead emerge directly from their spatial arrangement within the atomic register. This naturally translates the embedding problem into a combinatorial geometric mapping.

Given a graph describing a computational problem, the objective is to assign spatial coordinates to each vertex such that the induced interaction pattern matches the desired adjacency structure. Unlike abstract graph models, interactions cannot be freely specified, as they are constrained by inter-atomic distances. As a result, only a subset of graphs admits a feasible geometric realization within the quantum register.

This perspective leads to a direct encoding strategy in which logical variables are mapped one-to-one onto physical qubits, avoiding the use of auxiliary resources. The embedding task therefore consists in identifying a spatial configuration that satisfies the hardware-induced constraints while preserving the target connectivity.

To formalize the problem, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the graph associated with the adjacency matrix A, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ the set of edges. The complement

graph is denoted by $\mathcal{G}' = (\mathcal{V}, \bar{\mathcal{E}})$. Each vertex corresponds to a qubit, and $\vec{p}_i = (x_i, y_i)$ or $\vec{p}_i = (x_i, y_i, z_i)$ denote the spatial coordinates of the corresponding atom in a two- or three-dimensional register, respectively, for all $i \in \mathcal{V}$.



(a) Adjacency matrix **(b)** Unfeasible graph embedding **(c)** Feasible graph embedding

Figure 2.2: Embedding of a graph into a neutral-atom quantum register, where the adjacency matrix in Fig. 2.2a defines the target interaction pattern. Not all spatial configurations reproduce the desired connectivity.

In practice, the embedding process starts from the desired interaction pattern encoded in A and searches for a spatial configuration of atoms that realizes it. As illustrated in Figs. 2.2b and 2.2c, different placements may lead to infeasible or feasible embeddings, where feasibility means that the induced UDG exactly reproduces the adjacency structure defined by A. The goal is therefore to determine coordinates such that the induced geometric graph matches the target adjacency structure under the constraints imposed by the hardware.

2.2.1 From Hardware Constraints to Embedding Requirements

Designing embedding strategies for neutral-atom quantum processors requires explicitly incorporating the physical constraints imposed by the hardware. In the initial formulation of this research activity, the embedding problem was defined with respect to a reference device configuration corresponding to Pasqal's R&D prototype *Chadoq2* [55].

The quantum register defines the feasible domain for the spatial coordinates of the atoms. In this configuration, atoms can be positioned within a circular region of radius $L = 50\ \mu m$, which implies a maximum pairwise distance of $2L = 100\ \mu m$. Additionally, experimental constraints impose a minimum separation of $D_{\text{min}} = 4\ \mu m$ between atoms to ensure stable trapping in optical tweezers.

These geometric limitations translate into bounds on the pairwise Euclidean distance $d_{ij} = ||\vec{p}_i - \vec{p}_j||_2$:

$$d_{ij} \leq 2L \qquad \forall i, j \in \mathcal{V}^2,\ i < j \tag{2.1}$$

$$d_{ij} \geq D_{\text{min}} \qquad \forall i, j \in \mathcal{V}^2,\ i < j \tag{2.2}$$

Beyond these geometric bounds, the embedding must reproduce the interaction structure defined by the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. A necessary condition is that adjacent vertices are placed closer than non-adjacent ones, which can be expressed as

$$\max_{(i,j)\in\mathcal{E}} d_{ij} < \min_{(h,k)\in\bar{\mathcal{E}}} d_{hk}. \tag{2.3}$$

However, this condition is not sufficient to guarantee physical realizability. In neutral-atom systems, interactions are mediated by the Rydberg blockade, whose effective range is characterized by the blockade radius r_b [56]. This radius depends on the van der Waals interaction coefficient C_6 and the Rabi frequency Ω of the driving laser, according to

$$r_b = \sqrt[6]{\frac{C_6}{\hbar\Omega}}. \tag{2.4}$$

At the same time, the duration of the excitation pulse required to generate entanglement is given by

$$t = \frac{\pi}{\sqrt{2}\Omega}, \tag{2.5}$$

which must remain within the coherence time $\tilde{t}$ of the device [57, 45]. This constraint imposes a lower bound on Ω, and consequently an upper bound on the achievable blockade radius. By combining Eqs. (2.4) and (2.5), the maximum effective blockade radius $\tilde{r_b}$ can be expressed as

$$\tilde{r_b} = \sqrt[6]{\frac{\sqrt{2}|C_6|\tilde{t}}{\hbar\pi}}. \tag{2.6}$$

Taking these physical limitations into account, the embedding constraints can be formulated directly in terms of $\tilde{r_b}$:

$$d_{ij} \leq \tilde{r_b} \qquad \forall (i,j) \in \mathcal{E} \tag{2.7}$$

$$d_{hk} > \tilde{r_b} \qquad \forall (h,k) \in \bar{\mathcal{E}} \tag{2.8}$$

Together, these conditions define a constrained UDG embedding problem. Solving it amounts to determining spatial coordinates for the atoms such that the induced geometric graph reproduces the desired interaction pattern while satisfying the physical constraints of the quantum processor.

2.2.2 Representable Graph Classes and Unit-Disk Abstractions

Current neutral-atom quantum processors operate within a two-dimensional geometry which, combined with the constraints introduced in Eqs. (2.1)–(2.8), limits the class of

graphs that can be represented as feasible embeddings. As a consequence, not every graph admits a realization as a constrained unit-disk embedding under the physical constraints imposed by the hardware. It is therefore important to identify structural properties that characterize representable graph classes.

Two connectivity-related properties provide useful necessary conditions for embedding feasibility: the size of the maximum clique and the maximum degree of the graph $\mathcal{G}$. These bounds arise from the geometric limitations of the embedding space together with the minimum distance constraints imposed by the hardware.

The underlying geometric argument is based on *Thue's Theorem*, which states that the regular hexagonal packing achieves the densest circle packing in the plane [58]. Accordingly, the densest feasible arrangement of atoms that respects the minimal separation constraint corresponds to a triangular lattice. In the considered experimental setting, the minimum separation between atoms is $D_{\text{min}} = 4\ \mu m$, which determines the lattice spacing. Moreover, the interaction range is limited by the maximum achievable blockade radius. Given a coherence time $\tilde{t} = 3\ \mu s$, the corresponding upper bound is approximately $\tilde{r_b} \approx 10.26\ \mu m$. Together, these parameters define the geometric constraints governing feasible embeddings.

Property 1 (Maximum clique property)**.** *Consider the constrained unit-disk graph problem with parameters $D_{\text{min}} = 4\ \mu m$, $D_{adj} = 10.26\ \mu m$. Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be the graph to be embedded and let M denote the size of its largest clique K_M. A necessary condition for the existence of a feasible embedding is*

$$M \leq 7.$$

This bound follows from the fact that all vertices of a clique must lie within a disk of radius $D_{\text{adj}}/2$ to ensure pairwise interactions. Under the minimal separation constraint D_{min}, at most seven atoms can be placed within this region.

Property 2 (Maximum degree property)**.** *Under the same setting of 1, let deg_{max} denote the maximum degree of $\mathcal{G}$. A necessary condition for embedding feasibility is*

$$deg_{\text{max}} \leq 18.$$

This bound is obtained by considering the densest arrangement of neighbors around a single vertex. All adjacent vertices must lie within a disk of radius D_{adj}, and the minimal separation constraint limits their number to 18.

Figure 2.3 illustrates these geometric limits. The reference configuration is given by hexagonal packing with spacing $D_{\text{min}} = 4\ \mu m$.

It is important to note that these conditions are necessary but not sufficient. In particular, even if $\deg_{\text{max}} \leq 18$, the specific connectivity pattern among neighboring vertices may still prevent a valid embedding from existing. Figure 2.3 provides examples of feasible configurations that satisfy these bounds.

These results provide practical guidelines for identifying graph instances that are compatible with two-dimensional neutral-atom registers under realistic experimental constraints.

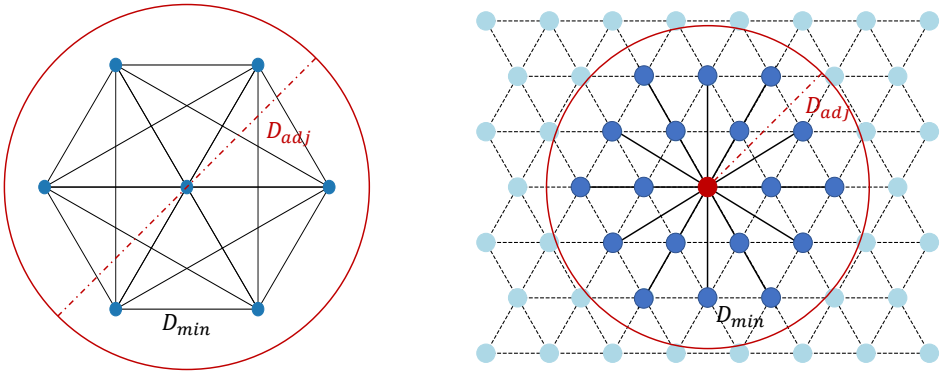


Figure 2.3: Geometric constraints on feasible embeddings: the maximum clique K_7 within the interaction radius (left), and the maximum number of neighbors under hexagonal packing with $D_{\text{min}} = 4\ \mu m$ and $D_{\text{adj}} \approx 10.26\ \mu m$ (right).

2.3 The Constrained Unit-Disk Graph Embedding as an Optimization Problem

The unit-disk property required for neutral-atom QPUs to represent computational problems, together with the hardware-induced constraints, naturally defines the *constrained unit-disk graph embedding problem.*

In this section, we introduce a mathematical programming formulation of this problem, which represents the most direct approach for tackling it using classical optimization solvers. Although this formulation is conceptually straightforward, it also highlights the intrinsic complexity of the embedding task. In particular, it allows us to better understand the limitations of exact methods when applied to this class of problems. At the same time, this formulation provides a useful benchmark against which alternative approaches, such as the neural-based embedding method introduced later in Section 2.4, can be evaluated in terms of feasibility, scalability, and solution quality.

2.3.1 Mathematical Programming Formulation

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ denote the undirected graph to be embedded, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ the set of edges. The number of vertices is $n = |\mathcal{V}|$, and vertices are indexed starting from 0. Let $\mathcal{P}$ denote the set of all unordered vertex pairs, with cardinality $|\mathcal{P}| = \frac{n(n-1)}{2}$. The matrix $A \in \mathbb{R}^{n \times n}$ represents the adjacency matrix of $\mathcal{G}$.

The feasibility domain of a constrained UDG instance is defined by three positive parameters:

- D_{min}: minimum allowed distance between any pair of vertices,
- D_{adj}: maximum distance between adjacent vertices,

- L: maximum radius of the circular (or spherical) region containing the embedding.

The embedding dimensionality is denoted by $N \in \{2,3\}$. Each vertex $i \in \mathcal{V}$ is associated with an N-dimensional vector $\vec{p}_i \in \mathbb{R}^N$ representing the spatial coordinates of the corresponding qubit.

Binary variables δ_{ij} are introduced for each pair $\{i,j\} \in \mathcal{P}$ in order to represent violations of the distance constraints:

$$\delta_{ij} := \begin{cases} 1 & \text{if the pairwise distance is infeasible} \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

The feasibility conditions depend on the adjacency relation between vertices. Adjacent vertices must lie within the interval $[D_{\text{min}}, D_{\text{adj}}]$, whereas non-adjacent vertices must be separated by a distance greater than D_{adj}. A small constant ϵ is introduced to avoid strict inequalities in the formulation.

The complete programming model is therefore defined as follows:

$$\begin{aligned}
\min_{\vec{p},\,\delta,\,d} \quad & (2L - D_{\text{min}} + \iota) \sum_{\{i,j\}\in\mathcal{P}} \delta_{ij} + d_{\text{adj}} - d_{\overline{\text{adj}}} && && \text{(2.10a)} \\
\text{s.t.} \quad & ||\vec{p}_i - \vec{p}_j||_2^2 \leq D_{\text{adj}}^2 + (8L^2 - D_{\text{adj}}^2)\delta_{ij} && (i,j) \in \mathcal{E}, && \text{(2.10b)} \\
& ||\vec{p}_i - \vec{p}_j||_2^2 \geq (1 - \delta_{ij}) D_{\text{min}}^2 && (i,j) \in \mathcal{E}, && \text{(2.10c)} \\
& ||\vec{p}_i - \vec{p}_j||_2^2 \leq d_{\text{adj}}^2 && (i,j) \in \mathcal{E}, && \text{(2.10d)} \\
& ||\vec{p}_i - \vec{p}_j||_2^2 \leq 4L^2 + 4L^2\delta_{ij} && (i,j) \notin \mathcal{E}, && \text{(2.10e)} \\
& ||\vec{p}_i - \vec{p}_j||_2^2 \geq (1 - \delta_{ij})(D_{\text{adj}} + \epsilon)^2 && (i,j) \notin \mathcal{E}, && \text{(2.10f)} \\
& ||\vec{p}_i - \vec{p}_j||_2^2 \geq d_{\overline{\text{adj}}}^2 && (i,j) \notin \mathcal{E}, && \text{(2.10g)} \\
& \vec{p} \in [-L, +L]^{n \times N}, && && \text{(2.10h)} \\
& d_{\text{adj}} \in [D_{\text{min}}, D_{\text{adj}}], && && \text{(2.10i)} \\
& d_{\overline{\text{adj}}} \in [D_{\text{adj}} + \epsilon, 2L], && && \text{(2.10j)} \\
& \delta \in \{0,1\}^{|\mathcal{P}|} && && \text{(2.10k)}
\end{aligned}$$

The proposed model introduces binary variables to explicitly enforce the feasibility of pairwise distances between vertices. In principle, a formulation relying solely on continuous variables could be designed. However, such models remain non-convex and are particularly sensitive to numerical inaccuracies when solved with nonlinear optimization solvers. Since the feasibility constraints correspond to strict physical limits of the hardware, approximate feasibility cannot be tolerated. Binary variables therefore allow constraint violations to be explicitly identified and penalized within the objective function, ensuring that the resulting solutions correspond to physically realizable embeddings.

The objective function penalizes infeasible pairwise distances through the binary variables δ_{ij} while simultaneously maximizing the *adjacency gap* $d_{\overline{\text{adj}}} - d_{\text{adj}}$. This encourages adjacent vertices to be positioned as close as possible and non-adjacent vertices to be placed as far apart as possible, thus improving the robustness of the resulting embedding.

Constraint (2.10b) activates the δ_{ij} variables, and its corresponding penalty in the objective (2.10a) whenever two adjacent vertices are further than D_{adj}. Constraints (2.10c) ensure that the minimum distance D_{min} is respected for feasible embeddings. Constraint (2.10d), together with the minimization of the objective function, makes the variable d_{adj} assume the value of the largest distance among adjacent vertices. Constraint (2.10e) restricts vertices to lie within the feasible region of the register. Constraint (2.10f) enforces separation between non-adjacent vertices beyond D_{adj}. Constraint (2.10g), together with the objective, ensures that $d_{\overline{\text{adj}}}$ corresponds to the minimum distance among non-adjacent pairs.

Finally, constraints (2.10h)–(2.10k) define the domains of the optimization variables $\vec{p}$, d_{adj}, $d_{\overline{\text{adj}}}$, and δ.

2.3.2 Computational Complexity and Hardness

The constrained UDG embedding problem is computationally challenging both from a theoretical and a practical perspective. Its difficulty is closely related to the complexity of geometric graph realization problems studied in graph theory and combinatorial optimization.

In particular, the *recognition problem* for unit-disk graphs, namely, determining whether a given graph admits a geometric realization, in which vertices correspond to points in the plane and edges connect pairs of vertices whose distance is below a fixed threshold, has been proven to be NP-hard [59]. This result directly implies that deciding the feasibility of a unit-disk embedding is computationally intractable in the general case. Further complexity results reinforce this conclusion. In [60], the authors analyze structural properties and approximation limits of unit-disk and d-quasi unit-disk graphs [61], showing that even approximate solutions remain difficult to obtain. Moreover, the problem does not become substantially easier when restricted to specific graph classes. In particular, Bhore et al. [62] show that the recognition problem remains NP-hard even for outerplanar graphs [63] and trees [64].

Despite this intrinsic hardness, approximation techniques have been explored in related domains such as wireless communication and sensor networks. In these settings, virtual coordinate systems are used to approximate geometric embeddings [65, 66, 67, 68]. However, these approaches typically allow approximate realizations and therefore cannot be directly applied in the quantum computing context considered in this work, where embedding constraints correspond to strict physical limitations of the hardware.

A natural approach to tackle the embedding problem is to formulate it as a non-convex optimization problem and solve it using classical optimization solvers. Several formulations of this type have been proposed in the literature [69]. Depending on the modeling choices, the problem can be addressed using different solver classes. For instance, mixed-integer quadratically constrained programming formulations can be solved using tools such as *Gurobi* [70], while nonlinear programming formulations can be handled by solvers such as *Ipopt* [71].

In this work, the proposed formulation was implemented using the *Pyomo*[1] Python library. Preliminary experiments indicated that *Ipopt* provides more stable behavior for the considered non-convex formulation, and it is therefore adopted as the reference classical solver for benchmarking purposes.

Nevertheless, the scalability of classical optimization approaches remains severely limited. Due to the NP-hard nature of the problem, the computational effort required to find feasible or optimal embeddings grows rapidly with the number of vertices n, corresponding to the number of qubits. In practice, this necessitates the introduction of strict time limits, often leading to incomplete or suboptimal solutions.

These limitations motivate the development of alternative embedding strategies that trade exact optimality for computational efficiency, while still producing high-quality feasible embeddings. The neural-based approaches introduced in the following sections are designed precisely to address this challenge.

2.4 Neural Optimization for Embedding under Hardware Constraints

The embedding problem introduced in the previous section can be formulated as a non-convex optimization problem and addressed using classical solvers. However, as discussed earlier, the computational complexity of the problem and the large number of geometric constraints make exact optimization approaches difficult to scale. Even for moderately sized graphs, classical solvers may require significant computational time to retrieve feasible embeddings, and in many cases they fail to find a solution within practical time limits.

These limitations motivate the exploration of alternative optimization paradigms. In particular, NNs (Neural Networks) have emerged as a promising tool for solving complex optimization problems. Instead of explicitly exploring the combinatorial search space, neural models learn parametric transformations that progressively refine an initial configuration toward a feasible or optimal solution. In this perspective, NNs act as differentiable optimizers capable of approximating the solution of otherwise intractable problems.

[1] https://pyomo.readthedocs.io

Several studies have investigated the use of neural architectures for solving optimization tasks. In [72], NNs are used to solve binary MILPs by designing architectures in which *ReLU* activation functions represent binary variables, while continuous variables correspond to the outputs of neural units. The approach is applied to feature visualization and adversarial machine learning problems. A related framework, *OptNet*, was proposed by Amos and Kolter [73], where quadratic programs are embedded directly within neural NN layers, enabling end-to-end learning of optimization tasks such as solving Sudoku puzzles. Neural optimization techniques have also been applied to more domain-specific problems. Chandrasekhar and Suresh proposed a neural formulation for topology optimization, where the weights and biases of a NN parametrize the density function of the optimized structure [74]. This approach was later extended to multi-material topology optimization problems [75]. Reinforcement learning techniques have also been successfully applied to combinatorial optimization tasks, such as the Capacitated Vehicle Routing Problem, where a trained policy can generate solutions for previously unseen instances [76]. Graph-based optimization problems have also been addressed using neural architectures. In particular, convolutional and graph NNs have been proposed for solving problems such as Vertex Cover and Maximum Independent Set [77, 78]. These approaches exploit the structural properties of graphs to guide the search for feasible or high-quality solutions.

Inspired by these developments, we propose a neural-enhanced optimization framework tailored to the constrained UDG embedding problem. The proposed approach processes one graph instance $\mathcal{G}(\mathcal{V}, \mathcal{E})$ at a time and aims to compute spatial coordinates for the qubits that satisfy the hardware-induced constraints described in the previous sections.

As illustrated in Fig. 2.4, the framework consists of two main stages. In the **preprocessing phase**, an initial configuration of qubit coordinates $\vec{p}_i^{\,0}$, $\forall i \in \mathcal{V}$, is generated from the input graph. This configuration does not necessarily correspond to a feasible embedding, but provides a starting point for the optimization process.

In the subsequent **learning phase**, a neural model iteratively transforms the initial configuration into a final set of coordinates $\vec{p}_i^{\,f}$, $\forall i \in \mathcal{V}$, corresponding to a feasible embedding of the graph in the quantum register.

2.4.1 Initialization of Vertex Coordinates

Since the proposed optimization framework learns spatial transformations of graph embeddings, an initial set of coordinates $\vec{p}_i^{\,0}$, $\forall i \in \mathcal{V}$, must be provided. These coordinates are generated during a preprocessing phase whose goal is to produce a suitable starting configuration for the learning process.

Providing an informed initialization significantly improves the convergence of the neural optimization procedure. In particular, starting from coordinates that partially satisfy the geometric constraints allows the subsequent NN to reach feasible embeddings in fewer training iterations compared to random initialization. Two strategies

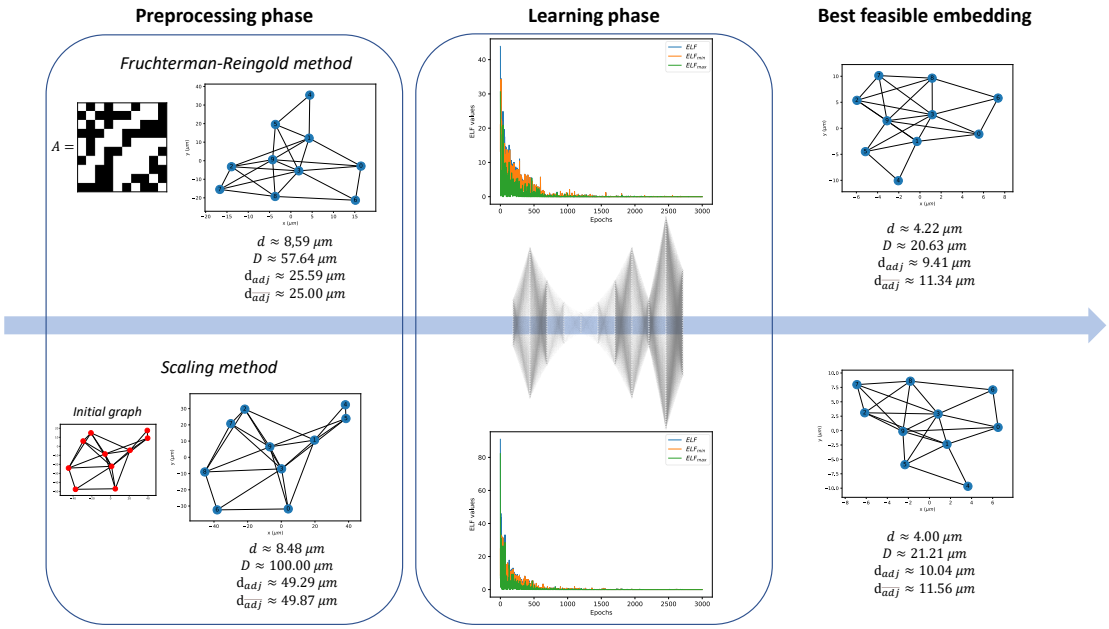


Figure 2.4: Neural optimization framework for constrained graph embedding on neutral-atom registers.

were investigated to compute the initial vertex coordinates.

Scaling method. Some computational problems are associated with predefined vertex coordinates $\vec{c}_i$, $\forall i \in \mathcal{V}$. These coordinates can be rescaled to match the domain of the quantum register, producing the initial positions $\vec{p}^0$. In particular, the scaling ensures that all vertices lie within a circular region of radius L. When a three-dimensional embedding is considered ($N = 3$), all z-coordinates are initially set to zero.

Fruchterman–Reingold method. When no initial coordinates are available, a force-directed layout algorithm can be used to generate an initial embedding. The Fruchterman–Reingold algorithm [79] models attractive and repulsive forces between vertex pairs according to the graph adjacency structure. In the implementation provided by the *NetworkX*[2] library, repulsive forces act between all vertex pairs with magnitude $k^2/\|\vec{p}_i - \vec{p}_j\|_2^2$, while attractive forces act only on adjacent vertices with magnitude $\|\vec{p}_i - \vec{p}_j\|_2/k$.

The parameter k represents the equilibrium distance at which these forces balance [80]. In this work, k is set to 7 μm, which lies within the feasible interval $[D_{\min}, D_{\text{adj}}]$. The algorithm is iterative and typically converges rapidly; therefore, the number of iterations was limited to 1000. Although this approach does not guarantee satisfaction of all embedding constraints, it provides a meaningful initialization even in the absence

[2]https://networkx.org

of prior geometric information.

Traditional optimization approaches could also be used to compute vertex coordinates by directly solving the corresponding non-convex quadratic programming problem. However, as discussed in Section 2.3.2, the unit-disk embedding problem is NP-hard, and classical solvers often fail to produce any solutions as the graph size or connectivity increases.

Overall, these approaches provide reasonable initial placements, which improve over random initialization but do not guarantee a feasible unit-disk embedding. The subsequent learning phase of the proposed framework therefore aims to transform these initial configurations into embeddings that fully satisfy the hardware constraints.

2.4.2 Neural Architecture Design

The proposed solution relies on a neural architecture, referred to as the DEN (Distance Encoder Network) [36], illustrated in Fig. 2.5. The model is composed of two main components: a trainable autoencoder that learns spatial transformations of the vertex coordinates, and a fixed-weight module that computes pairwise distances between vertices. The objective of the network is to transform an initial configuration of coordinates $\vec{p}^0$ into a feasible embedding $\vec{p}^f$ that satisfies the geometric constraints while maximizing the adjacency gap.

Trainable Autoencoder. The first component of the architecture is a symmetric autoencoder that maps the input coordinates (x_i, y_i) to transformed coordinates (x'_i, y'_i) in the case $N = 2$. Since the embedding domain is bounded, the output layer (*CoordL*) employs the activation function

$$F_a(t) = L \tanh(t),$$

which ensures that the generated coordinates lie within the square domain $(-L, L)^2$, enclosing the circular quantum register.

Both the input and output layers contain $N \times n$ nodes. The hidden layers follow a symmetric structure with sizes $64, 36, 18, 9, 18, 36, 64$, using ReLU activation functions. Dropout layers with probability p_{drop} are included for regularization. Fully connected layers are adopted instead of convolutional ones, as the ordering of vertices in the input vector does not encode any spatial locality.

Fixed-weight Distance Encoder. The second component of the architecture computes pairwise distances between the transformed coordinates. This module has fixed weights and explicitly encodes the geometric structure of the embedding.

The computation is carried out through two sparse layers.

The first layer, denoted as *DiffL*, computes coordinate differences for all vertex pairs. The weights of this layer are fixed to values in $\{-1, 0, +1\}$, so that the pre-activation outputs u are given by

$$u_{i(n-1)-\binom{i}{2}+j-i-1} = \overrightarrow{p_i^f}^{(x)} - \overrightarrow{p_j^f}^{(x)}, \quad \forall \{i,j\} \in \mathcal{P}, \tag{2.11}$$

$$u_{(n-1)(\frac{n}{2}+i)-\binom{i}{2}+j-i-1} = \overrightarrow{p_i^f}^{(y)} - \overrightarrow{p_j^f}^{(y)}, \quad \forall \{i,j\} \in \mathcal{P}. \tag{2.12}$$

A square activation function is then applied to obtain squared coordinate differences.

The second sparse layer, denoted as *DistL*, aggregates these values to compute squared Euclidean distances. The weights of this layer are fixed to $\{0, +1\}$, yielding the outputs v corresponding to squared pairwise distances:

$$v_k = u_k^2 + u^2_{\binom{n}{2}+k}, \qquad k \in \left\{1, \dots, \binom{n}{2}\right\}.$$

In an earlier version of the architecture [35], a final square-root activation was applied so that the network output directly represented distances d_{ij}, as shown in Fig. 2.5. In the final formulation, this activation is removed and the model outputs squared distances d_{ij}^2 [36]. This modification improves numerical stability and accelerates convergence during training.

2.4.3 Embedding Loss Function and Constraint Penalization

Once the pairwise distances between vertices are computed by the DEN architecture, the optimization objective can be formulated through a loss function that penalizes violations of the embedding constraints. Differently from standard machine learning tasks, the training procedure does not aim at prediction over a dataset. Instead, each training run processes a single graph instance and learns a spatial transformation that maps an initial configuration of coordinates into a feasible unit-disk embedding.

Initial Penalty-based Formulation. In an initial formulation, the loss function was defined directly on the pairwise distances d_{ij}. Since the embedding must satisfy four classes of constraints (Eqs. (2.1), (2.2), (2.7), and (2.8)), the loss function included four corresponding penalty terms.

The first contribution penalizes pairs of qubits whose distance exceeds the maximum allowed distance:

$$loss_1(d) = \operatorname*{avg}_{i<j} \left|\max(2L, d_{ij}) - 2L\right|.$$

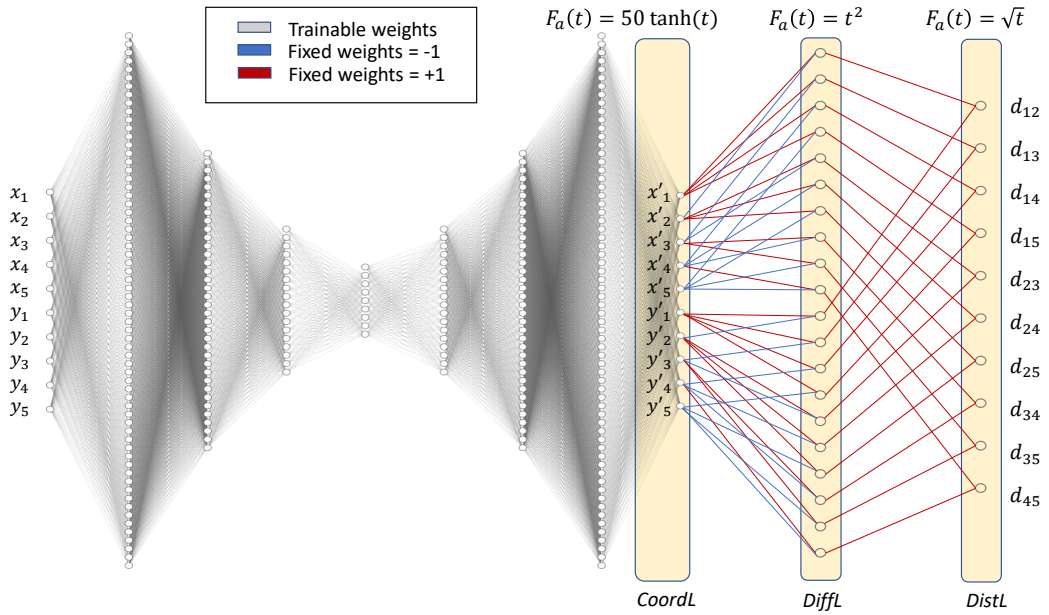


Figure 2.5: Architecture of the Distance Encoder Network. An autoencoder transforms input coordinates into new ones; a fixed-weight distance encoder then computes pairwise distances used by the loss function. Example shown for a 5-vertex graph.

The second contribution enforces the minimum separation constraint:

$$loss_2(d) = \underset{i<j}{\text{avg}} \left|\min(D_{\min}, d_{ij}) - D_{\min}\right|.$$

The third term penalizes adjacent vertices whose distance exceeds the interaction threshold:

$$loss_3(d) = \underset{i<j}{\text{avg}}\, A_{ij} \left|\max(D_{\text{adj}}, d_{ij}) - D_{\text{adj}}\right|.$$

The fourth contribution enforces separation between non-adjacent vertices:

$$loss_4(d) = \underset{i<j}{\text{avg}}\, (1 - A_{ij}) \left|\min(D_{\text{adj}} + \epsilon, d_{ij}) - (D_{\text{adj}} + \epsilon)\right|.$$

The overall loss is obtained by summing all contributions:

$$loss(d) = loss_1(d) + loss_2(d) + loss_3(d) + loss_4(d).$$

Although this formulation directly reflects the embedding constraints, it was later replaced by a more stable formulation operating on squared distances.

Embedding Loss Function (ELF). In the final formulation, the loss operates on the squared distances produced by the DEN model:

$$v = \|\overrightarrow{p_i}^f - \overrightarrow{p_j}^f\|_2^2, \qquad \forall\{i,j\} \in \mathcal{P}.$$

The embedding constraints are separated into lower-bound and upper-bound inequalities. The lower-bound constraints (Eqs. (2.10c), (2.10f)) are handled through ELF_{min}, while the upper-bound constraints (Eqs. (2.10b), (2.10e)) are handled through ELF_{max}.

Both components are implemented using a *margin ranking* formulation:

$$\text{MarginRanking}(v, v^t, m) = \text{avg}\left(\max(0, -m(v - v^t))\right),$$

where m determines the inequality direction and v^t represents the target squared distances.

For the lower-bound constraints, we set $m = \mathbb{I}_{|\mathcal{P}|}$ and define

$$v^t_{i(n-1)-\binom{i}{2}+j-i-1} = \begin{cases} D^2_{\text{min}} & A_{ij} = 1, \\ (D_{\text{adj}} + \alpha)^2 & A_{ij} = 0, \end{cases} \qquad \forall \{i, j\} \in \mathcal{P}.$$

For the upper-bound constraints, we set $m = -\mathbb{I}_{|\mathcal{P}|}$ and define

$$v^t_{i(n-1)-\binom{i}{2}+j-i-1} = \begin{cases} D^2_{\text{adj}} & A_{ij} = 1, \\ 4L^2 & A_{ij} = 0, \end{cases} \qquad \forall \{i, j\} \in \mathcal{P}.$$

The final embedding loss function is therefore given by

$$ELF(v) = ELF_{\text{min}}(v) + ELF_{\text{max}}(v).$$

Training Procedure. The loss function is minimized using the *AdamW* optimizer [81] with a tuned learning rate lr. For each graph instance, training is performed for a maximum of 3000 epochs. During training, feasible embeddings are continuously monitored and the best solution found so far is updated according to the adjacency gap criterion, favoring configurations that maximize the separation between adjacent and non-adjacent vertex pairs. At the end of the optimization process, the best feasible embedding identified during training is returned.

2.4.4 Adapting the Embedding to Real Quantum Processors

As introduced in the previous sections, a neural-network-based heuristic was developed to compute UDG embeddings for neutral-atom quantum architectures. The proposed methodology assumes that atoms can be freely positioned within the quantum register, provided that the geometric constraints induced by the Rydberg blockade are satisfied. This setting, referred to as the *free-space assumption*, was used to design the DEN and the associated embedding loss function.

When transitioning from this idealized model to real quantum hardware, additional constraints must be taken into account. In particular, practical implementations require adapting the embedding procedure to the specific characteristics of available devices.

In this work, we consider two neutral-atom QPUs: Orion Alpha by PASQAL and Aquila by QuEra. These platforms impose different restrictions on atom placement and, consequently, require different adaptations of the embedding strategy. The modifications required for each device are described below.

Orion Alpha: discrete trap lattice

The Orion Alpha device does not allow arbitrary atom placement. Instead, atoms must occupy predefined optical traps arranged in a triangular lattice with side length $D_{\text{min}} = 5\ \mu m$. The register contains 61 traps arranged so that the lattice can be inscribed within a disk of radius $L = 41/2\ \mu m$. Since the DEN model produces continuous coordinates, the resulting embedding must be mapped onto this discrete set of positions.

While updating the parameters D_{min} and L is straightforward within the DEN framework, the discrete nature of the lattice introduces a significant challenge. Incorporating integrality constraints directly into the loss function would make the optimization non-differentiable. Alternative approaches based on MIP (Mixed-Integer Program) were also explored using solvers such as *Gurobi* [70], *Ipopt* [71], and *GLPK* [82]. However, these formulations require a large number of constraints and were unable to produce feasible embeddings within a time limit of 5 hours, even for relatively small graphs with 7–10 vertices.

To address this limitation, a nearest-neighbor projection procedure that maps the continuous DEN solution onto the available trap positions was introduced. Vertices are processed in decreasing order of degree, so that highly connected vertices are assigned trap locations first, reducing the risk of infeasible assignments in later stages. The resulting algorithm is reported in Algorithm 1.

Although this projection step does not guarantee that the unit-disk representation is perfectly preserved, it consistently produces hardware-feasible embeddings and represents an effective compromise between solution quality and computational efficiency.

Algorithm 1 Embedding algorithm on Orion Alpha

$P \leftarrow$ DEN solution for $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
$T \leftarrow$ trap coordinates in the triangular lattice
$\mathcal{V}_s \leftarrow sort(\mathcal{V})$ ▷ sort vertices by decreasing degree
for $v \in \mathcal{V}_s$ **do**
  $p \leftarrow P(v)$
  $p' \leftarrow$ nearest neighbor of p in T
  $T \leftarrow T \setminus p'$
  $P(v) \leftarrow p'$
end for

Aquila: row-spacing constraints

In this setting, the free-space assumption can still be maintained. However, the Aquila device [29] introduces two important differences with respect to the reference configuration used in the design of the DEN model.

First, the Aquila register has a rectangular geometry of size $75\ \mu m \times 76\ \mu m$, instead of the circular domain considered in the original formulation. This difference can be incorporated by adapting the activation function of the coordinate layer so that the generated coordinates lie within the rectangular region. In practice, this corresponds to setting $L = 75/2$ and removing the circular-domain constraint, since each coordinate is already constrained within the register dimensions (up to a translation to positive coordinate values).

Second, the atom loading procedure imposes an additional row-spacing constraint: atoms must either lie on the same row or have a vertical separation of at least $D_{\text{row}} = 4\ \mu m$. To enforce this requirement, the DEN architecture is extended by introducing an additional component in the loss function. The modified architecture, shown in Fig. 2.6, augments the output layer so that it computes both squared pairwise distances and squared vertical distances $(y'_i - y'_j)^2$. The first $\binom{n}{2}$ outputs are used to evaluate the standard embedding constraints, while the remaining $\binom{n}{2}$ outputs capture the row separations.

To incorporate this information into the optimization process, a new loss component ELF_{row} is introduced (Fig. 2.7). This term penalizes vertex pairs whose vertical distance lies in the interval $(0, D_{\text{row}})$, while no penalty is applied when vertices lie on the same row or are separated by at least D_{row}.

The overall embedding loss function becomes

$$ELF = ELF_{\text{min}} + ELF_{\text{max}} + ELF_{\text{row}}.$$

When this loss reaches zero, the resulting coordinates correspond to an embedding that satisfies both the unit-disk constraints and the hardware-specific row-spacing requirement of the Aquila processor.

2.4.5 Beyond Two-Dimensional Embeddings

Although current neutral-atom quantum processors typically operate with atoms arranged on a two-dimensional register, extending the embedding methodology to three-dimensional configurations can increase the class of graphs that can be represented. Moreover, future quantum hardware may support fully three-dimensional atomic registers. For this reason, this methodolgy can be extended to generate feasible embeddings in three-dimensional space. The extension to three dimensions does not require modifications to the loss function, since the embedding constraints depend only on pairwise distances between qubits. Instead, the changes concern the neural architecture responsible for generating and processing the coordinates

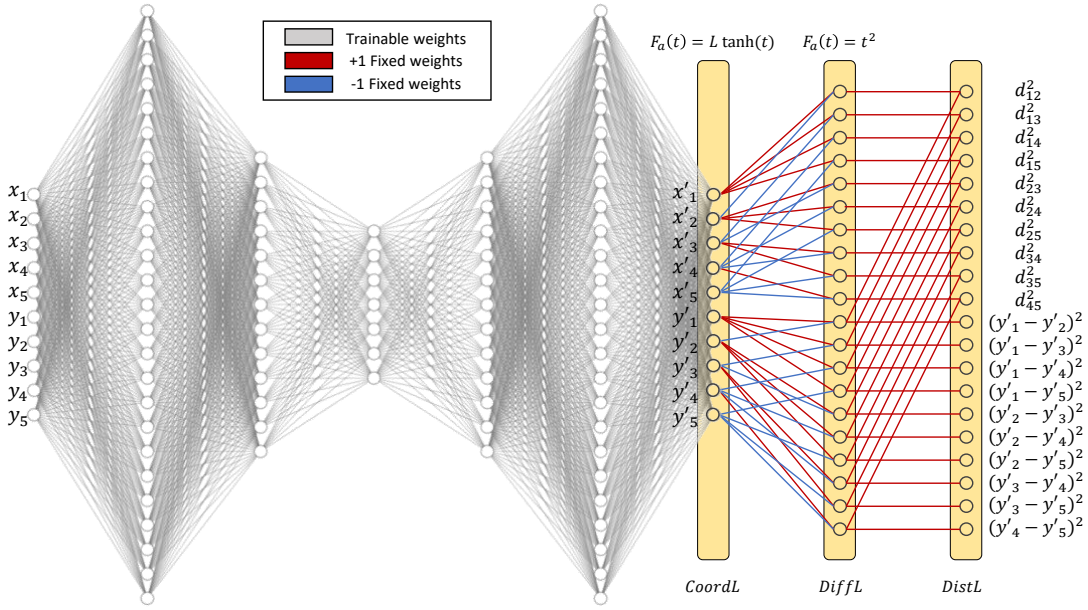


Figure 2.6: Modified DEN architecture for embedding on the Aquila QPU. The model computes squared pairwise and vertical distances to enforce the row-spacing constraint. Example for a 5-vertex graph.

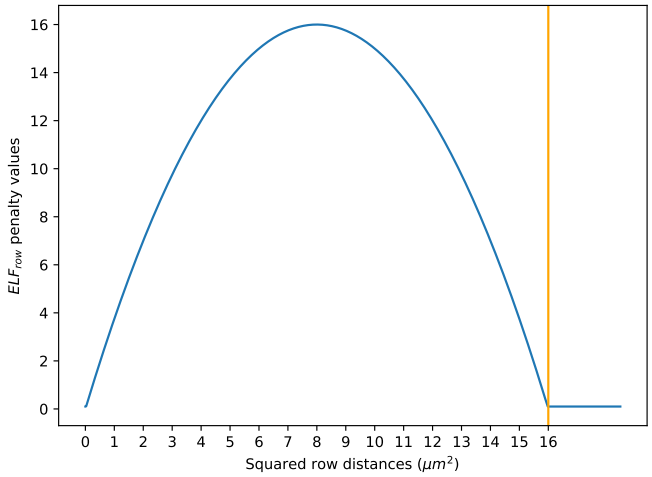


Figure 2.7: Row-spacing penalty function $\mathrm{ELF}_{\mathrm{row}}$: zero when vertices share the same row or when their vertical separation exceeds $4\ \mu m$.

In particular, three modifications are required.

- *Extension of the coordinate layers.* The input layer and the *CoordL* layer must be extended to represent the z coordinates of the atoms. Consequently, their size increases from $2n$ to $3n$ nodes, accounting for the coordinates (x_i, y_i, z_i) and

(x'_i, y'_i, z'_i) for all $i \in \mathcal{V}$. When no initial z_i values are available, they can be initialized to zero.

- *Extension of the difference layer.* The *DiffL* layer must also account for pairwise differences along the z axis. Its size therefore increases from $2\binom{n}{2}$ to $3\binom{n}{2}$ nodes. The weights of the sparse fully connected layer are modified so that the last $\binom{n}{2}$ nodes compute

$$u_{(n-1)(n+i-1)-\binom{i-1}{2}+j-i} = \overrightarrow{p_i^f}^{(z)} - \overrightarrow{p_j^f}^{(z)} \qquad i, j \in \mathcal{P}. \tag{2.13}$$

- *Modification of the distance computation.* The *DistL* layer maintains the same size, since it still outputs the $\binom{n}{2}$ pairwise distances. However, the computation must now include the contribution of the squared z-coordinate differences. Therefore, the values of the v nodes become

$$v_k = u_k^2 + u^2_{\binom{n}{2}+k} + u^2_{2\binom{n}{2}+k}, \qquad k \in \left\{1, \dots, \binom{n}{2}\right\}. \tag{2.14}$$

With these modifications, the proposed architecture can generate embeddings in three-dimensional space while preserving the same constraint-handling mechanism used in the two-dimensional case.

2.5 Empirical Evaluation

This section reports on the effectiveness of the proposed neural-enhanced optimization framework for solving the constrained unit-disk graph embedding problem. The evaluation focuses on both the capability of the method to retrieve feasible embeddings and the quality of the resulting spatial configurations.

Different experimental settings are considered. First, a synthetic benchmark dataset of randomly generated graphs in order to compare the proposed method with classical optimization solvers under controlled conditions. Second, we evaluate the embedding framework on application-driven datasets arising from practical quantum computing use cases, whose subsequent solutions methods and algorithms are coved in Chapter 3, for the optimization uses case, and in Chapter 4 for the machine learning applications. For the cases, in which we ran on the neutral atoms QPU, for the feasibility of the embedding it was considered also the additional constraint set described in Section 2.4.4

2.5.1 Benchmark Datasets and Experimental Setup

This section describes the datasets and experimental configurations adopted to evaluate the proposed embedding framework. The analysis considers both controlled synthetic benchmarks and application-driven scenarios, enabling a comprehensive assessment of feasibility, scalability, and generalization capabilities.

Artificial Dataset

To evaluate the proposed neural-enhanced optimization framework, its performance is compared with a classical nonlinear solver, namely *Ipopt* [71, 83], applied to the programming model (2.10a)–(2.10k).

A synthetic benchmark dataset composed of 200 graph instances was generated, including 20 graphs for each value of $n \in \{10,20,\dots,100\}$. The generation process follows the feasibility conditions discussed in Section 2.2.2, ensuring that structural properties compatible with unit-disk embeddings are preserved.

Graph instances were obtained by sampling vertex coordinates $\vec{c}_i$, $\forall i \in \mathcal{V}$, within a square domain of side l, and defining edges according to a unit-disk rule: two vertices are connected if their distance is below a threshold d. No explicit constraints were imposed on minimum distances or on the relationship between d and l, preventing trivial solutions obtained through simple scaling.

The parameters l and d were progressively increased with n in order to generate graphs of increasing density. The dataset generation required approximately 2.3 minutes.

To ensure representability, the following structural conditions were enforced:

- estimated maximum clique size $M \leq 7$;
- maximum vertex degree $\deg_{\max} \leq 18$;
- single connected component.

The maximum clique size was estimated using the *NetworkX* heuristic algorithm [84].

Experimental Configuration. The feasible domain parameters were set according to the *Chadoq2* prototype:

$$D_{\min} = 4\ \mu m, \qquad D_{\text{adj}} \approx 10.26\ \mu m, \qquad L = 50\ \mu m.$$

The parameter ι was set to 1 for the programming model, while $\epsilon = 0.1$ was used for both the ELF (Embedding Loss Function) and the programming model. Each DEN model was trained for a maximum of $E = 3000$ epochs.

Hyperparameters were selected through grid search over

$$lr \in \{10^{-2},10^{-3},10^{-4}\}, \qquad p_{\text{drop}} \in \{0.3,0.5,0.7\}.$$

Combined with the two initialization strategies described in Section 2.4.1, scaling and Fruchterman–Reingold (FR), this resulted in 18 independent trials per graph instance.

All experiments were executed on an IBM Power9 cluster with 32 CPU cores and 256 GB of RAM per node, equipped with NVIDIA Volta V100 GPUs. Due to the single-instance nature of the training (mini-batch size equal to 1), GPU acceleration did not provide significant benefits. Instead, experiments were parallelized across CPU cores, with each trial using 8 cores.

Application-Driven Datasets

Beyond synthetic benchmarks, the embedding framework was evaluated on application-driven datasets arising from practical quantum computing scenarios. These datasets allow assessing the method in realistic settings where the embedding constitutes a pre-processing step for hybrid quantum-classical algorithms.

Cellular Antenna Networks. The *antennas dataset* originates from the vertex coloring optimization research activity [38, 39, 40] and represents a cellular network in the city of Turin obtained from the OpenCellID database.[3]

Conflicts between antennas arise when their distance is below a threshold $D_c = 140$m, naturally defining a UDG. The original dataset contains up to 90 antennas, requiring the embedding of graphs with up to 87 vertices.

Since the graph can be decomposed into connected components, embeddings are computed independently for each component. After decomposition, the dataset includes:

- 9 isolated vertices;
- 6 K_2, 4 K_3, 1 K_4, and 2 K_5 cliques;
- four non-trivial components with 7, 7, 10, and 19 vertices.

Only non-trivial components are considered, as isolated vertices and cliques admit trivial embeddings and optimization solutions. Since spatial coordinates are available, initialization is performed via scaling. The DEN model is then used to transform this initial (generally infeasible) configuration into a feasible embedding under the free-space assumption. The resulting coordinates are finally projected onto the Orion Alpha lattice using Algorithm 1.

[3] https://opencellid.org/

Protein Graphs. For quantum machine learning experiments, the PROTEINS dataset [85, 86] was used. Only graphs with $n \leq 256$ vertices were considered, matching the maximum number of atoms available on the Aquila QPU.

Three subsets were defined:

- PROTEINS_12: $n \leq 12$
- PROTEINS_16: $n \leq 16$
- PROTEINS_256: $n \leq 256$

Smaller subsets were used for classical emulations of the quantum system and quantum algorithm tuning, while the largest subset was used for experiments on the QPU. Initialization is performed using the Fruchterman–Reingold method, with $lr = 10^{-3}$ and $p_{\text{drop}} = 0.5$. Further details on the hybrid quantum machine learning method are provided in Section 4.2.

Additional QUBO Formulations

In addition to graph-based datasets, the embedding framework was evaluated on QUBO formulations derived from different application domains. This analysis serves two complementary purposes. First, it demonstrates that the proposed method is not restricted to explicitly geometric graphs, but can be applied to interaction graphs derived from general optimization problems. Second, it assesses the capability of the embedding procedure to handle structures that do not naturally satisfy the unit-disk property.

The relevance of these formulations is closely tied to the constraints imposed by neutral-atom quantum hardware. In this setting, an optimization problem must first be expressed in QUBO form, i.e., through a matrix $Q \in \mathbb{R}^{n \times n}$, where n corresponds to the number of binary variables and each variable is mapped to a physical qubit.

However, the hardware introduces specific restrictions on the structure of Q. Due to the use of globally applied laser pulses, when targeting *Chadoq2* prototype, all qubits are driven with the same Rabi frequency [28], which implies that the diagonal elements of Q must be uniform. Moreover, pairwise interactions between variables are mediated by the Rydberg blockade mechanism [87], whose strength scales as C_6/d_{ij}^6. As a consequence, the off-diagonal elements of Q must have consistent sign and their magnitude is directly determined by the geometric arrangement of the qubits. Under these conditions, a compatible QUBO formulation can be mapped to the effective Hamiltonian implemented by the hardware (1.1).

While feasible QUBO formulations can be constructed for several problems (e.g., vertex coloring [33]), the coupling coefficients are ultimately constrained by the inter-qubit distances, making the embedding step essential.

One representative example is the protein folding problem [88], where hydrophobic interactions are encoded through binary variables δ_{ij}. The corresponding QUBO formulation is given by

$$\min_{\delta_{ij}} -\sum_{i>j} \delta_{ij} + P \sum_{ij,kh \in S} \delta_{ij}\delta_{kh}, \qquad \delta_{ij} \in \{0,1\},$$

with

$$S = \{ij, kh \mid i \leq f < j,\ f \neq \tfrac{i+j-1}{2},\ f = \tfrac{k+h-1}{2} \in \mathbb{N}\}.$$

In this formulation, the objective promotes configurations that maximize the number of hydrophobic interactions, corresponding to favorable protein folding structures. Additional QUBO instances include one-hot encoded vertex coloring formulations [89]. In this setting, each vertex is expanded into a set of binary variables encoding the possible color assignments, with additional constraints enforcing both exclusivity (one color per vertex) and adjacency conditions. This transformation produces interaction graphs with increased size and structured connectivity patterns, which can significantly complicate the embedding task.

2.5.2 Results Analysis

The empirical results obtained with the proposed DEN-based embedding framework are evaluated using a set of geometric indicators that quantify both feasibility and embedding quality.

Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with n vertices, let

$$d_{\text{adj}} = \max_{(i,j)\in\mathcal{E}} d_{ij}, \qquad d_{\overline{\text{adj}}} = \min_{(i,j)\notin\mathcal{E}} d_{ij}.$$

A feasible embedding must satisfy

$$d_{\text{adj}} \leq D_{\text{adj}} \qquad \text{and} \qquad d_{\text{adj}} < d_{\overline{\text{adj}}}.$$

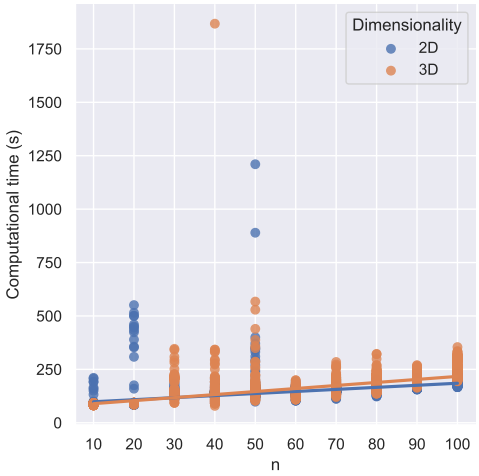
The difference $d_{\overline{\text{adj}}} - d_{\text{adj}}$, referred to as the *adjacency gap*, measures how well the embedding separates interacting and non-interacting qubit pairs. Larger values of this quantity correspond to more robust embeddings, as they reduce the risk of unintended interactions due to positioning uncertainties and noise effect.

Using these metrics, the analysis is organized according to the role of the datasets in the overall study. First, synthetic benchmarks are used to assess scalability and to compare the proposed approach against a classical optimization baseline. Then, application-driven datasets are considered to evaluate the practical adaptation of the method to real neutral-atom QPUs. Finally, additional QUBO instances are discussed to highlight the generality of the embedding strategy beyond the specific use cases addressed in the subsequent chapters.

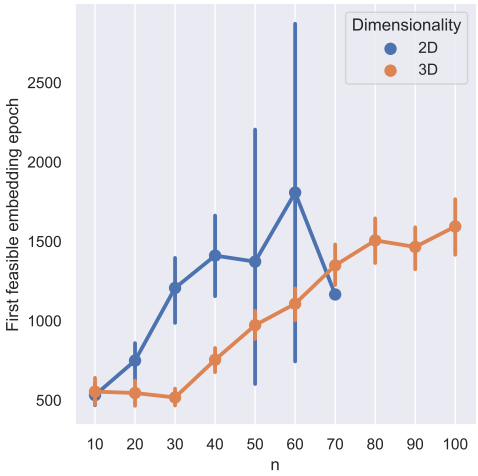
Synthetic Benchmark Results

The synthetic benchmark provides a controlled setting for comparing the proposed framework with the classical solver *Ipopt*, while also assessing the effect of graph size,

embedding dimensionality, and initialization strategy on the performance of the DEN model.



(a) Learning phase computational time vs. graph size $n = |\mathcal{V}|$



(b) Epoch at which the first feasible embedding is found during training

Figure 2.8: Scaling of computational time and first feasibility epoch with graph size and embedding dimensionality for the DEN model.

Figure 2.8a reports the variability in computational time across the $3600 = 200 \times 18$ trials, corresponding to 200 graph instances and 18 combinations of hyperparameters and initialization methods, for both embedding dimensionalities $N = 2$ and $N = 3$. For each pair (n, N), an average computational time $T_{n,N}$ is computed. To compare with the classical baseline, a multi-start *Ipopt*-based optimization was performed using 18 starting points and a maximum walltime for each run equal to $T_{n,N}$, selected according to the graph size and the targeted embedding dimensionality.

Figure 2.8b shows the epoch at which the first feasible embedding was found for each trial. As expected, increasing the dimensionality of the embedding space makes the problem easier to solve. On average, feasible embeddings are found earlier when $N = 3$, and successful solutions are obtained for larger graph sizes than in the two-dimensional case. By contrast, when $N = 2$, increasing graph size makes the problem substantially harder, and for $n = 70$ only one graph was successfully embedded within the 3000-epoch budget.

The advantage of the proposed approach is even more evident in Fig. 2.9, which reports the percentage of feasible embeddings found by each optimization strategy. For the DEN framework, the results obtained with the two initialization methods are shown separately. To make the comparison more favorable to the classical baseline, *Ipopt* was also allowed to exploit up to $10T_{n,N}$ computational time, which in some cases exceeded half an hour. Nevertheless, this additional time did not produce feasible solutions for the graph instances that were not already solved within the original walltime.

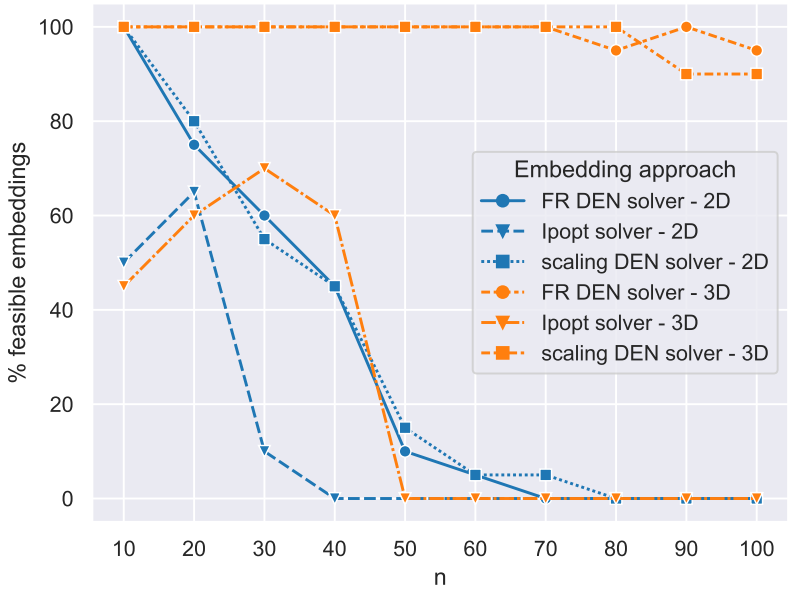


Figure 2.9: Feasibility rate across the synthetic graph dataset for $N = 2$ and $N = 3$: the DEN model outperforms *Ipopt*, while Fruchterman–Reingold and scaling initializations perform comparably.

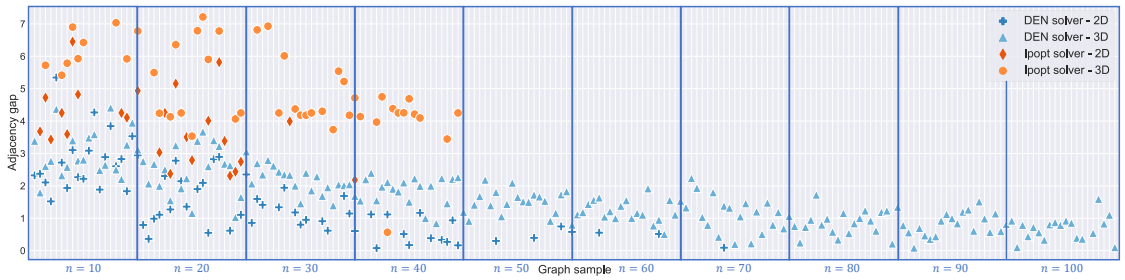


Figure 2.10: Largest adjacency-gap values obtained by each solver on the synthetic dataset, grouped by graph size n (blue rectangles).

Figure 2.10 reports the results related to adjacency-gap maximization. The DEN-based optimization clearly outperforms the classical approach when feasibility is the primary objective. However, whenever *Ipopt* succeeds in finding a feasible embedding, the resulting solution often exhibits a larger adjacency gap. This observation suggests that future work should investigate more refined strategies for optimizing this quantity within the ELF formulation.

A final observation concerns hyperparameter selection. Within the explored search space, no specific combination of learning rate lr and dropout probability p_{drop} consistently yielded a higher success rate. The retrieval of a feasible embedding therefore

appears to depend only weakly on these hyperparameters. For the subsequent experiments, the values $lr = 10^{-3}$ and $p_{\text{drop}} = 0.5$ were adopted.

Embeddings for the Orion Alpha QPU

The first application-driven results concern vertex coloring instances mapped to the Orion Alpha QPU. In this setting, the original graph is derived from a cellular antenna network, and the DEN model is first used to compute feasible embeddings under the free-space assumption. These embeddings are then remapped onto the discrete triangular lattice of optical traps available on the hardware.

The antenna network is decomposed into connected components, and only the non-trivial ones require the intervention of the DEN. The corresponding results are summarized in Table 2.1, where d_{adj} and $d_{\overline{\text{adj}}}$ denote the free-space values, while d^l_{adj} and $d^l_{\overline{\text{adj}}}$ denote the corresponding quantities after remapping to the lattice through Algorithm 1.

Table 2.1: Antenna-dataset embeddings. All distances are in μm.

Graph	d_{adj}	$d_{\overline{\text{adj}}}$	d^l_{adj}	$d^l_{\overline{\text{adj}}}$	$\deg_{\max}$	M
$7G_3cc$	10.2	16.3	10.0	13.2	5	4
$7G_13cc$	10.2	12.2	10.0	10.0	5	4
$10G_4cc$	10.1	15.7	10.0	10.0	4	4
$19G_1cc$	10.2	13.3	10.0	10.0	9	6

Under the free-space assumption, the DEN model found feasible embeddings for all four non-trivial graphs. As expected, the nearest-neighbor remapping procedure generally deteriorates the quality of the embedding, since the difference $d^l_{\overline{\text{adj}}} - d^l_{\text{adj}}$ is consistently smaller than the original gap $d_{\overline{\text{adj}}} - d_{\text{adj}}$. In some cases, the remapped solution yields $d^l_{\overline{\text{adj}}} - d^l_{\text{adj}} = 0$, indicating that the hardware-induced discretization may collapse the separation between adjacent and non-adjacent pairs.

Despite this degradation, the subsequent vertex coloring algorithm [39] proved robust enough to solve the problem even for these non-ideal unit-disk configurations.

This robustness can be attributed to the small number of edges that are not perfectly represented after remapping. Figure 2.11 illustrates the lattice embedding of graph $19G_1cc$. In this case, considering a unit-disk radius $r_b = 10.0\ \mu m$ would introduce only one additional edge, shown in orange, between vertices 13 and 22.

Embeddings for the Aquila QPU

The second set of application-driven results concerns graph-classification experiments carried out on the Aquila QPU using the PROTEINS dataset. In this case, the embedding

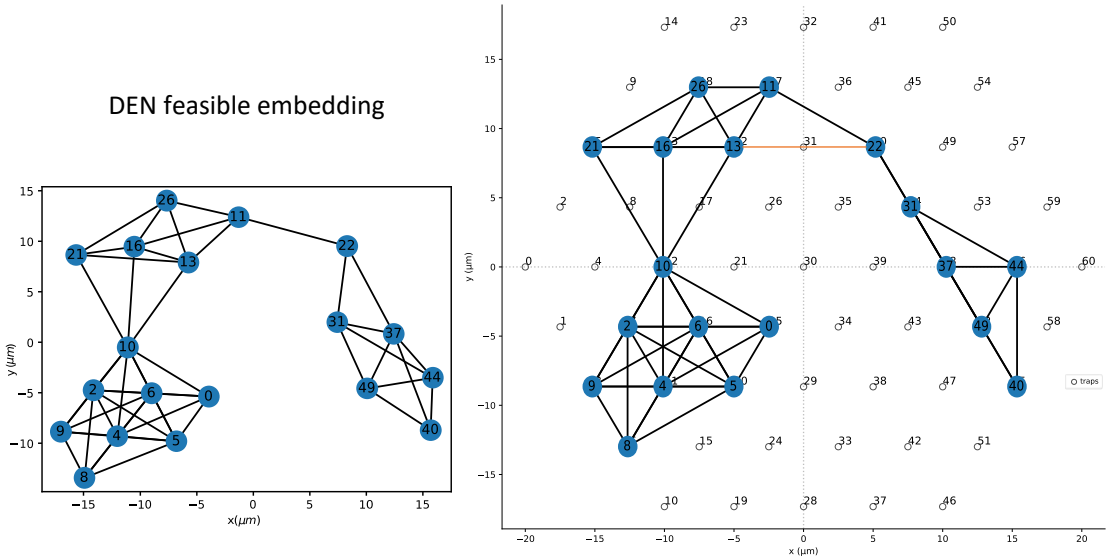


Figure 2.11: Embedding of sample $19G_1cc$ onto the Orion Alpha register.

method was adapted to account for the rectangular geometry of the register and for the row-spacing constraint introduced through the additional term ELF_{row}.

For the smaller subsets of the dataset, the proposed method achieved encouraging success rates. In the PROTEINS_12 subset, feasible embeddings were found for 158 out of 207 graph samples, corresponding to approximately 76%. When the PROTEINS_16 subset is also considered, the total becomes 210 feasible embeddings out of 307 samples, corresponding to approximately 68%.

Figure 2.12 shows that the probability of obtaining a feasible embedding decreases as the number of vertices increases. This behavior is expected, since larger graphs must be embedded within a finite rectangular register while satisfying both the unit-disk constraints and the row-spacing requirement.

A similar trend is visible in Fig. 2.13, which reports the distributions of the adjacency gap. As the graph size increases, it becomes increasingly difficult to find embeddings that clearly separate adjacent and non-adjacent pairs.

When the experiments are extended to the PROTEINS_256 dataset, feasibility becomes significantly harder to achieve. The total number of feasible embeddings rises to 279, but this corresponds to only 34% of the overall dataset. Although this lower success rate still proved useful for the downstream classification task, it also highlights the limitations of the current approach when targeting graphs with a number of vertices close to the maximum number of atoms supported by the QPU.

This difficulty is consistent with the geometry of the problem, since placing 256 vertices within a $75\ \mu m \times 76\ \mu m$ rectangle while satisfying the minimum-separation and

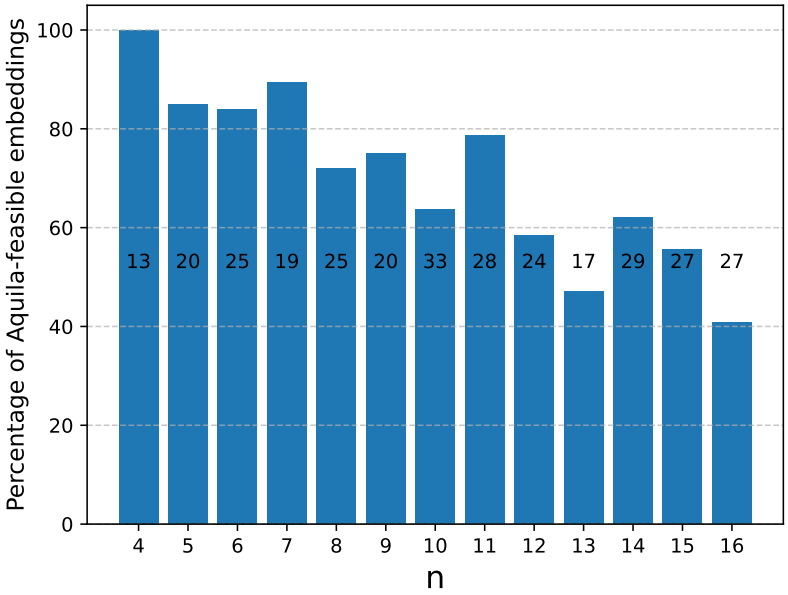


Figure 2.12: Aquila-feasibility rate of the DEN model, grouped by number of vertices n. Bin labels indicate the total number of graph samples..

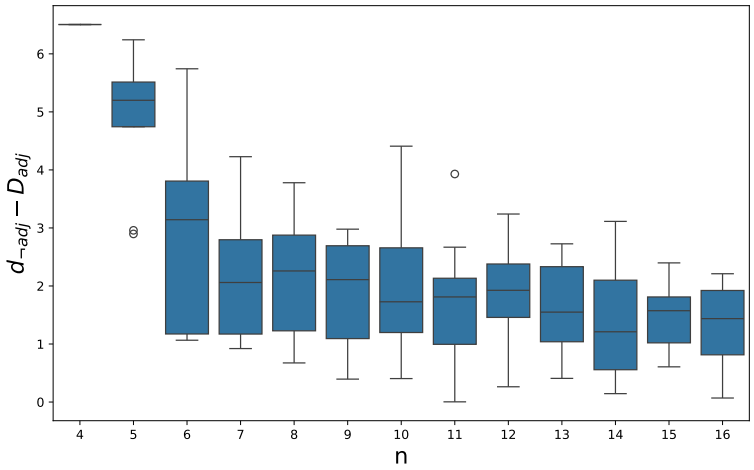


Figure 2.13: Gap $d_{\overline{\text{adj}}} - d_{\text{adj}}$ distributions grouped by n for the PROTEINS_12 and PROTEINS_16 datasets.

row-spacing constraints makes an exact representation of a general graph extremely demanding.

To illustrate the complexity of the embedding task, Fig. 2.14 reports a feasible Aquila

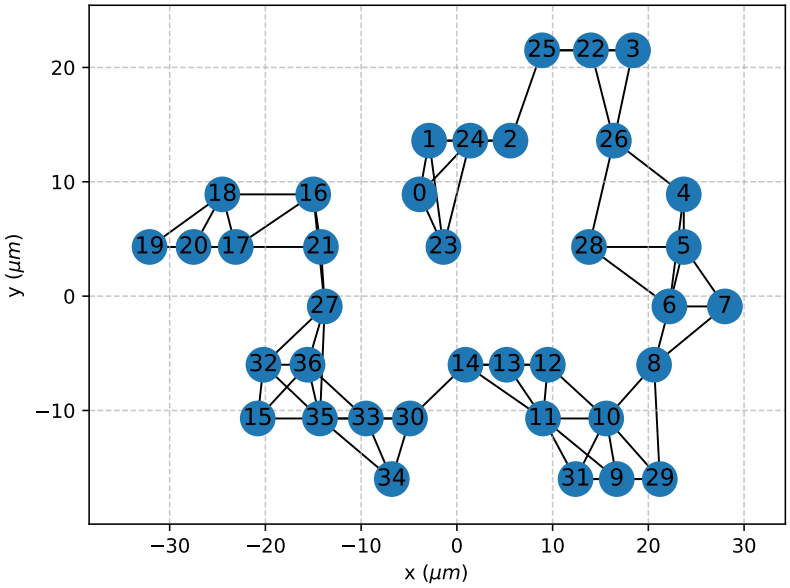


Figure 2.14: DEN embedding of sample 115 from the PROTEINS_256 dataset.

embedding for sample 115 in the PROTEINS_256 dataset. This graph contains 37 vertices and is characterized by $\deg_{\max} = 7$ and $M = 4$. Feasibility was achieved with $d_{\text{adj}} = 10.0\ \mu m$ and $d_{\overline{\text{adj}}} = 10.4\ \mu m$.

It is worth noting that, even when the success rate appears modest for larger graphs, state-of-the-art optimization solvers achieved success rates close to zero even on smaller instances, despite walltimes on the order of 5 to 10 hours. By contrast, the proposed heuristic usually produces feasible embeddings in less than 5 minutes.

Additional QUBO Embedding Experiments

Beyond the experiments directly tied to Orion Alpha and Aquila, the proposed framework was also evaluated on additional QUBO instances in order to assess its flexibility across different graph structures and problem formulations. These experiments make it possible to analyze the behavior of the embedding procedure on interaction graphs that are not directly derived from the application-driven datasets, while still remaining relevant for neutral-atom quantum computing.

To summarize the embedding characteristics reported in this section, the superscripts 0 and f denote quantities measured before and after optimization, respectively:

- $d^0_{\min}$, $d^f_{\min}$: minimum pairwise distance;
- $d^0_{\max}$, $d^f_{\max}$: maximum pairwise distance;

- d^0_{adj}, d^f_{adj}: maximum distance among adjacent pairs;
- $d^0_{\overline{\text{adj}}}$, $d^f_{\overline{\text{adj}}}$: minimum distance among non-adjacent pairs;
- E_c: number of epochs required to obtain a feasible embedding;
- I_{sol}: whether the classical *Ipopt* solver finds a feasible solution.

The first set of experiments concerns instances of the QUBO protein-folding problem. Feasible embeddings were obtained for all considered instances. The graphs associated with these formulations are denoted by $\mathcal{G}_{aa_ha}$, where aa is the number of amino acids in the chain and ha the number of hydrophobic ones. The considered instances are:

- $\mathcal{G}_{12_6}$, with hydrophobic amino acids at positions 1, 2, 3, 5, 11, and 12;
- $\mathcal{G}_{17_7}$, with hydrophobic amino acids at positions 1, 2, 5, 6, 10, 12, and 17;
- $\mathcal{G}_{22_8}$, with hydrophobic amino acids at positions 1, 3, 5, 6, 9, 10, 11, and 17.

In these experiments, the initial coordinates were generated using the Fruchterman–Reingold method with equilibrium distance $k = 4\,\mu m$. As shown in Table 2.3, the initial layouts do not satisfy the unit-disk property, since $d^0_{\text{adj}} > d^0_{\overline{\text{adj}}}$ and the maximum distance among adjacent vertices is often larger than $D_{\text{adj}} = 10.26\,\mu m$.

The structural properties of the graphs and the *Ipopt* performance are reported in Table 2.2, while the geometric embedding statistics before and after optimization are summarized in Table 2.3.

Table 2.2: Structural properties and optimization performance for the QUBO protein-folding instances.

$\mathcal{G}$	n	$\deg_{\max}$	M	E_c	I_{sol}
$\mathcal{G}_{12_6}$	5	4	3	187	Yes
$\mathcal{G}_{17_7}$	10	9	4	492	No
$\mathcal{G}_{22_8}$	9	7	4	404	Yes

To further illustrate the behavior of the optimization process, Fig. 2.15 reports a representative example corresponding to the instance $\mathcal{G}_{17_7}$.

The evolution of the loss function in Fig. 2.15c) highlights the relative importance of its different components. Most of the contribution arises from the adjacency-related terms, with $loss_3$ and $loss_4$ accounting on average for 37% and 31% of the total loss, respectively. This confirms that enforcing correct separation between adjacent and non-adjacent vertices is the most critical aspect of the embedding optimization.

Table 2.3: Embedding distance statistics before (^{0}) and after (^{f}) optimization for the DEN model. All distances are in μm.

$\mathcal{G}$	d^0_{min}	d^f_{min}	d^0_{max}	d^f_{max}	d^0_{adj}	d^f_{adj}	$d^0_{\overline{\text{adj}}}$	$d^f_{\overline{\text{adj}}}$
$\mathcal{G}_{12_6}$	27.344	4.000	74.611	11.324	55.524	9.853	27.344	11.138
$\mathcal{G}_{17_7}$	11.601	4.056	64.463	20.237	32.719	10.255	22.277	10.606
$\mathcal{G}_{22_8}$	9.438	4.104	79.640	23.329	27.957	10.217	27.215	10.514

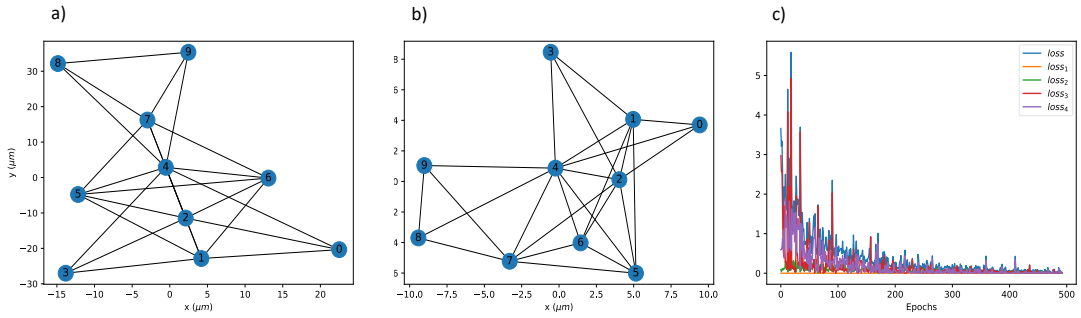


Figure 2.15: Embedding of the QUBO protein-folding instance $\mathcal{G}_{17_7}$: Fruchterman–Reingold initialization (a), feasible 2D embedding (b), and loss function over epochs (c).

Three-dimensional Embeddings

The last set of experiments concerns cases in which the restrictions of the two-dimensional domain prevent a feasible embedding, while feasibility can be recovered by exploiting the third dimension.

A first example arises from a more complex representation of the antenna-network problem obtained by increasing the conflict distance to $D_c = 250\ m$. In this setting, the associated graph becomes a single connected component with 87 vertices. The corresponding embedding problem therefore requires positioning 87 qubits inside a spherical domain of radius $50\ \mu m$.

The increased dimensionality allows the model to handle a more complex graph characterized by $\deg_{\text{max}} = 14$ and $M = 10$. The effect of the DEN transformation on the coordinates is substantial: the minimum distance between qubits increases from $d^0_{\text{min}} = 0.001\ \mu m$ to $d^f_{\text{min}} = 4.007\ \mu m$, while the maximum distance among adjacent qubits is reduced from $d^0_{\text{adj}} = 13.552\ \mu m$ to $d^f_{\text{adj}} = 10.134\ \mu m$ within 2120 epochs. The unit-disk property is preserved, with $d^f_{\overline{\text{adj}}} = 10.415\ \mu m$, and the maximum-distance constraint remains well satisfied since $d^f_{\text{max}} = 53.300\ \mu m$. The *Ipopt* solver did not provide a feasible solution within comparable computational times.

A second example comes from a QUBO vertex coloring problem. Starting from

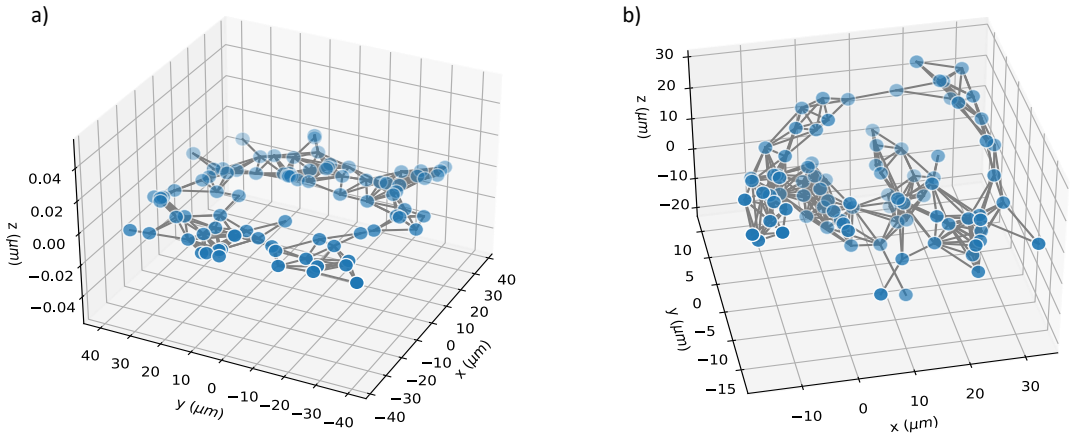


Figure 2.16: Embedding of the full MIS antenna problem in 3D: initial positions with $z = 0$ (a) and resulting feasible 3D embedding (b).

the red graph shown in Fig. 2.17, a one-hot encoded QUBO formulation for a three-color vertex coloring problem was constructed. The resulting graph $\mathcal{G}_{GC}$ contains 21 vertices, with $\deg_{\max} = 6$ and $M = 3$, the latter corresponding to the exclusivity constraint among the three colors associated with each original vertex.

The effect of the DEN transformation can be summarized as follows:

$$
\begin{aligned}
d^0_{\min} &= 8.948\ \mu m & \longrightarrow \quad & d^f_{\min} = 4.183\ \mu m \\
d^0_{\max} &= 67.980\ \mu m & \longrightarrow \quad & d^f_{\max} = 28.545\ \mu m \\
d^0_{\text{adj}} &= 19.217\ \mu m & \longrightarrow \quad & d^f_{\text{adj}} = 10.242\ \mu m \\
d^0_{\overline{\text{adj}}} &= 9.137\ \mu m & \longrightarrow \quad & d^f_{\overline{\text{adj}}} = 10.265\ \mu m
\end{aligned}
$$

The feasible embedding was obtained within 767 epochs, whereas *Ipopt* did not reach feasibility within the available walltime.

2.6 Limitations, Practical Adaptations, and Outlook

The results discussed throughout this chapter show that the proposed embedding methodology significantly reduces the gap between the abstract graph formulation of a computational problem and its physical realization on neutral-atom quantum hardware. By learning spatial transformations that satisfy unit-disk constraints and hardware-specific geometric requirements, the proposed DEN model enables the representation of non-trivial graph instances on current quantum processors. In this sense, the method provides a concrete bridge between problem-level formulations and the effective Hamiltonian implemented by the machine.

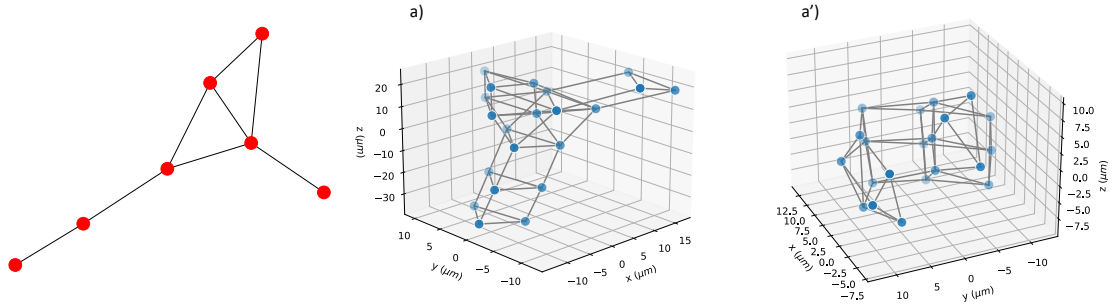


Figure 2.17: 3D QUBO embedding for vertex coloring on $\mathcal{G}_{GC}$: original graph (left), initial qubit configuration (center), and feasible embedding after optimization (right).

At the same time, the proposed approach inherits some limitations. First, its effectiveness remains tied to the geometric constraints of the target hardware, including register shape, minimum atom separation, admissible interaction radius, and device-specific loading constraints. Although the framework can be adapted to different processors, as shown in the case of Orion Alpha and Aquila, each new platform requires an explicit reformulation of the embedding constraints and, in some cases, architectural modifications of the neural model or of the loss function. Second, while the method is considerably more effective than classical exact optimization on the considered instances, the success rate decreases as the graph size and structural complexity increase. This behavior is particularly evident when the number of vertices approaches the maximum number of atoms supported by the QPU.

A further limitation concerns the optimization objective itself. The current framework prioritizes the retrieval of feasible embeddings, while the maximization of the adjacency gap remains only partially optimized. As observed in the experiments, feasible solutions can often be obtained even when the separation between adjacent and non-adjacent pairs is small. This is sufficient for some applications, but larger adjacency gaps would improve robustness against positional uncertainty and hardware noise. A more systematic treatment of this aspect, including improved stopping criteria and a deeper convergence analysis, remains an important direction for future work.

Despite these limitations, the proposed approach offers several advantages. It does not rely on fixed graph templates or on chain-based minor embeddings, and it avoids the overhead of introducing auxiliary qubits solely to emulate unavailable interactions. Instead, it directly exploits the geometric flexibility of neutral-atom platforms while still respecting the actual hardware constraints. As a consequence, it provides a practical tool for testing graph-based problem encodings on machines with a limited number of qubits. Moreover, the methodology is naturally extensible to higher-dimensional embeddings. While this chapter focused primarily on two-dimensional configurations and discussed an extension to the three-dimensional case, the underlying distance-based construction can in principle be generalized to arbitrary dimensions.

Future developments will follow several directions. From a theoretical perspective, a deeper characterization of the graph classes that admit feasible constrained unit-disk embeddings would improve the understanding of which problem instances are naturally compatible with neutral-atom hardware. From an algorithmic perspective, a more systematic comparison with classical nonlinear solvers and heuristic embedding strategies would provide a clearer picture of the scalability regime in which neural optimization becomes most advantageous. From a practical perspective, further experiments on real quantum hardware, will help quantify the tolerance of the embedding procedure to imperfections in atom placement and to deviations in the effective Rydberg blockade radius.

The experiments on Orion Alpha and Aquila already show that hardware adaptation is not a secondary implementation detail, but a central component of the embedding problem itself. On Orion Alpha, the need to remap continuous free-space solutions onto a discrete triangular lattice highlights the tension between geometric flexibility and hardware calibration requirements. On Aquila, the introduction of row-spacing constraints demonstrates that even when free-space placement is retained, the geometry of the loading procedure may substantially affect the set of feasible embeddings. These observations suggest that future embedding methodologies should increasingly be designed together with the operational characteristics of the target processor, rather than treating hardware adaptation as a purely final-stage correction.

2.6.1 Integration into Hybrid Quantum-Classical Pipelines

A major motivation for developing hardware-aware embedding strategies is their role within hybrid quantum-classical workflows. In such pipelines, the embedding stage acts as an interface between the classical description of the problem and the quantum routine executed on the processor. This role is particularly important for neutral-atom platforms, where the geometry of the register directly determines the interaction graph available to the quantum algorithm.

Within a hybrid optimization workflow, the embedding procedure can be viewed as a compilation stage that precedes the execution of the quantum routine. A graph instance is first transformed into a hardware-compatible spatial arrangement of qubits, after which the corresponding Hamiltonian can be realized on the device and integrated into a variational, iterative, or measurement-driven optimization loop. In this setting, the quality of the embedding has a direct impact on the efficiency of the overall pipeline, since it determines whether the intended problem structure is faithfully represented and how robustly the resulting quantum evolution reflects that structure.

A similar role emerges in quantum machine learning settings. When graph-based data must be encoded into a neutral-atom register, the embedding stage determines whether the topology of the input sample can be transferred to the hardware in a way that preserves the relevant relational information. The experiments on protein graphs

illustrate that this step is not merely a geometric preprocessing operation, but a necessary condition for enabling the subsequent learning task on the QPU.

More broadly, the results of this chapter support the view that embedding should be regarded as an integral component of hybrid quantum-classical algorithm design. Rather than being treated as an isolated technical subproblem, it should be incorporated into the full computational workflow, together with the choice of problem formulation, the structure of the quantum routine, and the characteristics of the target hardware. Under this perspective, hardware-aware embedding is not only a constraint-satisfaction problem, but also a design principle for making hybrid quantum algorithms executable on real neutral-atom devices.

Chapter 3

Hybrid Quantum Optimization on Neutral Atom Platforms

The embedding methodology introduced in the previous chapter enables the mapping of combinatorial optimization problems onto the geometric constraints of neutral-atom quantum hardware. Building upon this foundation, this chapter focuses on the solution of such problems through hybrid quantum-classical optimization strategies.

Combinatorial optimization problems are characterized by exponentially large search spaces, which make them challenging for classical algorithms as the problem size increases. Quantum computing offers new opportunities to address these challenges by exploiting coherent quantum dynamics to explore complex energy landscapes. Among the available quantum computing platforms, neutral-atom analog devices are particularly promising due to their scalability and their intrinsic ability to encode interaction constraints directly at the hardware level [28].

In these systems, qubits are realized by individual atoms trapped in optical tweezers and manipulated via laser pulses. The computational basis is typically defined by a ground state and a highly excited Rydberg state. A key physical mechanism is the Rydberg blockade [56], which prevents the simultaneous excitation of atoms within a certain distance. As discussed in the previous chapter, this effect induces an interaction pattern that depends on the spatial arrangement of atoms and can be exploited to enforce constraints between variables. This hardware naturally supports the solution of problems such as the MIS (Maximum Independent Set), where the objective is to select the largest subset of non-adjacent vertices in a graph. When the problem graph admits a UD (unit-disk) representation, the Rydberg blockade directly enforces the independence constraint, enabling a direct mapping between the problem structure and the physical interactions [90, 46]. However, many relevant optimization problems do not exhibit this structure and therefore cannot be directly encoded.

In these cases, the problem must be reformulated into an equivalent energy minimization form, such as a QUBO (Quadratic Unconstrained Binary Optimization) model [33]. This reformulation introduces constraints on the interaction graph, which must be

embedded into the physical geometry of the quantum register. Once a feasible embedding is obtained, the resulting Hamiltonian can be explored using quantum evolution techniques, such as adiabatic protocols or variational algorithms.

While QUBO formulations are natively suited to quantum optimization, many real-world problems are naturally expressed as MILP (Mixed-Integer Linear Program) [91] or ILP (Integer Linear Program) formulations. These can be addressed within the same hybrid framework by combining classical reformulation techniques with quantum subroutines, enabling the treatment of more general problems such as VC (Vertex Coloring) [92, 39]. In this context, hybrid quantum-classical approaches play a crucial role, combining quantum subroutines with classical optimization techniques to overcome current hardware limitations.

The rest of this chapter presents hybrid optimization frameworks and their application to two practically motivated problem classes, together with their implementation on neutral-atom quantum processors.

3.1 Hybrid Quantum-Classical Optimization

The embedding framework introduced in the previous chapter enables the representation of combinatorial optimization problems on neutral-atom QPUs (Quantum Processing Units). However, practical limitations of current devices, such as the finite number of qubits and restricted interaction patterns, prevent the direct solution of large or highly connected problem instances. As a consequence, purely quantum approaches are often insufficient, and an effective integration with classical optimization techniques becomes necessary.

A natural way to address these limitations is through *problem decomposition*. Instead of solving a complex problem as a whole, the idea is to partition it into smaller and more manageable components. These subproblems can then be solved using the most suitable computational resource, combining the strengths of classical and quantum methods. In this hybrid setting, the classical component is responsible for coordinating the global optimization process, enforcing feasibility, and handling global constraints. The quantum component, on the other hand, is used selectively to tackle computationally demanding substructures that can be efficiently mapped to the hardware. Typical examples include problems such as MIS or specific QUBO instances, which naturally match the interaction model induced by the Rydberg blockade.

This paradigm has been explored in several recent works, where quantum routines are embedded within classical optimization frameworks to accelerate the solution of hard subproblems [93, 92]. In these approaches, the quantum processor acts as a specialized solver for specific combinatorial tasks, while the classical algorithm maintains control over the global optimization process. More generally, decomposition techniques

have a long tradition in classical optimization. Methods such as Benders decomposition [94, 95, 96, 97] and CG (Column Generation) [98] split large problems into a coordinating master problem and a set of subproblems, significantly improving scalability. These principles naturally extend to hybrid quantum-classical settings, where some of the subproblems can be delegated to quantum hardware. More advanced strategies combine different decomposition paradigms, such as tree-based search and problem partitioning, leading to integrated frameworks like BP (Branch-and-Price), which have proven effective in a wide range of applications [99, 100, 101, 102]. Recent works have started to explore the integration of quantum algorithms within these schemes, for example by employing variational methods such as QAOA (Quantum Approximate Optimization Algorithm) [103] or quantum annealing techniques to solve selected subproblems [104, 105].

The approach adopted in this work follows this general paradigm. In particular, neutral-atom QPUs are used to solve subproblems that can be naturally expressed in terms of MIS-type constraints, leveraging the Rydberg blockade mechanism. This design choice is consistent with the embedding framework introduced in Chapter 2 and enables an efficient use of the available quantum resources.

3.1.1 Analog Quantum Evolution under Rydberg Blockade

Within the hybrid framework introduced above, the quantum processor is responsible for solving subproblems that can be naturally mapped onto the physical interactions of the hardware. Understanding how these problems are encoded and solved at the quantum level therefore requires a closer look at the underlying dynamics of neutral-atom QPUs.

Neutral-atom QPUs encode qubits using individual atoms trapped in optical tweezers and manipulated through laser pulses. Each atom represents a two-level system, typically defined by a ground state ($|g\rangle$) and a highly excited Rydberg state ($|r\rangle$). A key mechanism governing the system is the Rydberg blockade [56], which prevents the simultaneous excitation of atoms within a certain distance, effectively inducing constraints between qubits. As discussed in Chapter 2, these interactions depend on the spatial arrangement of the atoms and therefore on the embedding of the problem graph into the quantum register. The system dynamics are described by a time-dependent Hamiltonian $\mathcal{H}(t)$ (see (1.1)), parameterized by the Rabi frequency $\Omega(t)$, the detuning $\delta(t)$, and the interaction terms V_{ij}. By shaping these parameters over time, the quantum system explores the energy landscape associated with the encoded optimization problem and generates candidate solutions that can be integrated within the classical optimization loop.

Problem Encoding via Register Design

The encoding of an optimization problem onto the QPU is achieved through the design of the atomic register, where qubits are spatially arranged so that their interactions reproduce the structure of the problem. The construction of such configurations relies on the embedding procedures introduced in Chapter 2. In particular, the DEN (Distance Encoder Network) model is used to generate atom positions that are compatible with the geometric constraints of the hardware while approximating the target interaction graph.

When an exact unit-disk representation is not available, approximate embeddings are employed. Although these configurations remain physically feasible, they may not perfectly preserve the original adjacency structure, leading to deviations between the intended and the implemented interactions. Within the hybrid framework, these discrepancies influence the quality of the quantum-generated solutions. As a result, the effectiveness of the overall optimization process depends not only on the quantum evolution, but also on how accurately the embedding captures the structure of the underlying problem.

Problem Solving via Pulse Shaping

Once the problem has been encoded through the register design, the quantum processor is used to explore the solution space by evolving the system under a sequence of laser pulses. This stage constitutes the core quantum routine within the hybrid framework, where candidate solutions are generated by exploiting the dynamics of the underlying Hamiltonian.

The evolution is governed by time-dependent control parameters, in particular the Rabi frequency $\Omega(t)$ and the detuning $\delta(t)$, whose temporal profiles define the pulse sequence. This process, commonly referred to as pulse shaping, determines how the system navigates the energy landscape associated with the encoded problem.

When the embedding accurately captures the structure of the target interaction graph, adiabatic evolution can be employed. In this approach, the system is initialized in the ground state of a simple Hamiltonian and gradually evolved toward the problem Hamiltonian by smoothly varying the control parameters. This corresponds to the QAA (Quantum Adiabatic Algorithm) [106], which is particularly effective for problems such as MIS on unit-disk graphs, where the Rydberg blockade naturally enforces independence constraints. The resulting pulse schedules are smooth and predefined, making this approach well suited for analog neutral-atom hardware.

When the embedding is approximate or when greater flexibility is required, variational quantum algorithms can be adopted. In particular, the QAOA [6] alternates between the application of a cost Hamiltonian and a mixer Hamiltonian, with parameters optimized through a classical feedback loop. This allows the algorithm to compensate for imperfections in the embedding or hardware, at the cost of increased computational overhead due to repeated QPU evaluations.

These two approaches offer complementary advantages. Adiabatic evolution provides a hardware-efficient solution for problems that admit a suitable geometric embedding, while variational methods extend the applicability of the framework to more general interaction structures.

3.2 Application I: PCI Optimization in Wireless Networks

The hybrid quantum-classical approach introduced in the previous sections can be applied to a combinatorial optimization problem arising in wireless communication networks, namely the PCI (Principal Cell Identifier) assignment problem. This application provides a relevant test case to evaluate the effectiveness of the proposed methodology on instances characterized by practical constraints arising from a real-world scenario.

Modern wireless networks are becoming increasingly dense due to the rapid growth in the number of connected devices. As a consequence, ensuring reliable connectivity and efficient resource allocation has become a challenging task. Current cellular standards, such as LTE/LTE-A, rely on advanced communication techniques to mitigate interference and support large numbers of users [107, 108]. Within this context, the assignment of PCIs plays a crucial role. Each cell in the network must be assigned an identifier that supports efficient mobility management of user equipment, particularly during handover procedures between neighboring cells [109, 108]. Improper assignments may lead to interference phenomena such as collisions and confusions, which degrade the overall quality of service [110, 111].

From an optimization perspective, the PCI assignment problem can be modeled as a vertex coloring problem, where vertices represent cells and edges encode interference relationships. This problem is known to be NP-hard [109], and has traditionally been addressed using heuristic and metaheuristic methods [112, 113, 114].

In the context of quantum optimization, a common approach is to reformulate vertex coloring problems as QUBO models [5]. However, such formulations typically rely on one-hot encodings, which introduce additional binary variables for each color. This significantly increases the number of required qubits and results in dense interaction graphs that are difficult to embed on current quantum hardware.

In this work, a different strategy is adopted. Instead of relying on QUBO formulations, the PCI problem is decomposed into subproblems that can be expressed in terms of MIS-like structures. This choice is motivated by the physical properties of neutral-atom QPUs, where the Rydberg blockade mechanism naturally enforces independence constraints between nearby qubits. As a result, MIS problems can be directly mapped onto the hardware without the need for auxiliary variables or penalty terms.

Following the hybrid paradigm introduced earlier, the PCI problem is therefore partitioned into smaller subproblems that are compatible with the quantum processor. The quantum layer is used to solve these MIS-type subproblems, while the classical layer

coordinates the overall optimization process, ensuring consistency and handling global constraints. This approach enables the treatment of large-scale instances despite the limitations of current NISQ (Noisy Intermediate-Scale Quantum) devices.

The following sections detail optimization strategies, from basic to more advanced methods, used to solve the PCI-derived vertex coloring problem.

3.2.1 Solving the Maximum Independent Set with Neutral-Atom QPUs

The central computational primitive exploited in the hybrid methods presented in this chapter is the ability of neutral-atom QPUs to naturally represent and solve MIS and MWIS (Maximum-Weighted Independent Set) problems. Before introducing more advanced optimization frameworks, it is therefore useful to describe how these problems can be directly addressed on analog quantum hardware.

Neutral-atom QPUs encode qubits using individual atoms trapped in optical tweezers, each representing a two-level system with ground state $|g\rangle = |0\rangle$ and Rydberg state $|r\rangle = |1\rangle$. When atoms are excited to the Rydberg state, they interact through van der Waals forces whose strength decays with the inter-atomic distance as

$$V_{ij} \propto \|\vec{p}_i - \vec{p}_j\|^6, \tag{3.1}$$

as described by the Hamiltonian in (1.1). A key feature of this interaction is the *Rydberg blockade* [56], which prevents two atoms within a certain distance from being simultaneously excited. By arranging atoms such that vertices connected by an edge lie within a blockade radius r_b, the physical system encodes a UDG (Unit-Disk Graph). In this representation, edges correspond to pairs of atoms that cannot both be in the state $|1\rangle$. As a consequence, any valid excitation pattern corresponds to an independent set of the graph. The ground state of the system therefore encodes a MIS, or more generally a MWIS when vertex-dependent weights are introduced.

Quantum Adiabatic Algorithm

The QAA exploits adiabatic quantum evolution to drive the system toward low-energy configurations of the problem Hamiltonian. The algorithm proceeds in two stages. First, the system is prepared in the ground state of a simple driving Hamiltonian, which is easy to construct and whose ground state is known analytically. On neutral-atom platforms, this corresponds to initializing all atoms in their electronic ground state under strong Rabi driving and large negative detuning, i.e., $\Omega(t_0) > 0$ and $\delta(t_0) \ll 0$. Second, the laser parameters $\Omega(t)$ and $\delta(t)$ are smoothly varied so that the Hamiltonian $\mathcal{H}(t)$ interpolates between the driving and the cost Hamiltonian:

$$\Omega(t_0) > 0, \quad \delta(t_0) \ll 0, \tag{3.2}$$

$$\Omega(t_f) \to 0, \quad \delta(t_f) \gg 0. \tag{3.3}$$

At the end of the evolution, the Hamiltonian is dominated by the detuning term, which encodes the MIS cost function. According to the adiabatic theorem [115], provided the evolution is sufficiently slow relative to the inverse spectral gap, the system remains close to its instantaneous ground state throughout, and the final measurement yields a low-energy solution with high probability.

The choice of pulse parameters is not arbitrary but is tied to the geometric embedding of the problem graph. Let

$$d_{\text{adj}} = \max_{(i,j)\in\mathcal{E}} d_{ij}, \tag{3.4}$$

$$d_{\overline{\text{adj}}} = \min_{(i,j)\notin\mathcal{E}} d_{ij}, \tag{3.5}$$

denote the maximum distance among adjacent vertex pairs and the minimum distance among non-adjacent ones, respectively. A good embedding satisfies $d_{\text{adj}} < d_{\overline{\text{adj}}}$, ensuring that adjacent atoms interact more strongly than non-adjacent ones. From these two quantities, an effective blockade radius is defined as their geometric mean:

$$r_b = \sqrt{d_{\text{adj}} \cdot d_{\overline{\text{adj}}}}, \tag{3.6}$$

which represents a threshold separating interacting from non-interacting atom pairs. The corresponding Rabi frequency is then set to match the interaction strength at this radius:

$$\Omega_b = \frac{C_6}{r_b^6}, \tag{3.7}$$

where C_6 is the van der Waals coefficient characterizing the Rydberg interaction. The actual Rabi frequency used during the pulse is capped at the hardware maximum:

$$\Omega = \min(\Omega_b, \Omega_{\text{max}}), \tag{3.8}$$

ensuring compatibility with the physical constraints of the device.

This configuration promotes the generation of valid independent sets during the evolution, since atoms within the blockade radius suppress each other's excitation. In the case of exact unit-disk embeddings, the independence constraint is strictly enforced by the geometry. When approximate embeddings are used, the effective interaction graph may differ slightly from the target one; however, the resulting measurement outcomes remain meaningful candidates for the optimization process and can be post-processed classically to recover feasible solutions.

Quantum Approximate Optimization Algorithm

While the QAA relies on a fixed evolution schedule, the QAOA provides a variational alternative that is more robust to hardware noise and imperfect embeddings. In this

approach, the system evolves under alternating applications of a cost Hamiltonian $\mathcal{H}_C$ and a mixer Hamiltonian $\mathcal{H}_M$. The resulting state after p layers is given by

$$|\psi(\gamma,\beta)\rangle = \prod_{l=1}^{p} e^{-i\beta_l \mathcal{H}_M} e^{-i\gamma_l \mathcal{H}_C} |s\rangle \,, \tag{3.9}$$

where $|s\rangle$ is a suitable initial state, and $\gamma = (\gamma_1, \dots, \gamma_p)$ and $\beta = (\beta_1, \dots, \beta_p)$ are variational parameters optimized through a classical feedback loop [6]. The cost Hamiltonian encodes the MIS objective, while the mixer Hamiltonian promotes exploration of the solution space. The parameters are iteratively updated to minimize the expected energy

$$\langle \mathcal{H}_C \rangle = \langle \psi(\gamma,\beta)| \mathcal{H}_C |\psi(\gamma,\beta)\rangle \,. \tag{3.10}$$

In the neutral-atom analog implementation considered here, the variational parameters γ and β correspond to the durations of piecewise-constant pulse segments applied to the Rabi frequency $\Omega(t)$. The amplitude of Ω is fixed according to the embedding geometry, as described in the previous section, and applied globally to all atoms. The detuning is set to $\delta = 0$ throughout the evolution, so that the optimization acts solely through the timing of the drive pulses. This parametrization reduces the search space while remaining compatible with the control capabilities of neutral-atom hardware.

This flexibility comes at the cost of increased computational overhead, as multiple QPU evaluations are required during parameter optimization.

Remark: Both QAA and QAOA enable the solution of MIS-type problems on neutral-atom hardware by leveraging the Rydberg blockade to enforce independence constraints directly at the physical level. In the hybrid framework considered in this work, the quantum processor is primarily used as a sampler of high-quality independent sets, which are then integrated within a classical optimization loop. The following subsections describe how this capability can be exploited to address more complex combinatorial problems, starting from VC.

3.2.2 Iterative MIS-Based Coloring

A first approach to solving VC problems, without introducing additional binary variables for each possible color assignment [33], leverages the natural ability of neutral-atom QPUs to efficiently solve MIS instances.

The key observation is that any MIS corresponds to a valid color class. Since all vertices in an MIS are pairwise non-adjacent, they can be assigned the same color without violating feasibility. Based on this property, a coloring can be constructed iteratively by repeatedly extracting independent sets from the graph. At each iteration, an MIS is computed on the current graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The vertices belonging to the selected MIS are assigned a new color and removed from the graph, together with their incident

edges. This results in a reduced graph

$$\mathcal{G}' = (\mathcal{V} \setminus \text{MIS}, \mathcal{E}'),$$

on which the procedure is repeated until all vertices have been colored.

The complete procedure is reported in Algorithm 2. This strategy, referred to as *Greedy-it-MIS*, selects at each step the first feasible MIS returned by the solver, without evaluating its impact on the overall number of colors. As a consequence, the method is greedy in nature: it makes locally optimal choices without considering their global implications. While Greedy-it-MIS always produces a feasible solution to the VC problem, it does not explicitly aim at minimizing the number of colors. The quality of the solution therefore depends on the sequence of MIS selections, which may lead to suboptimal colorings.

Algorithm 2 Greedy-it-MIS for VC problems

Ensure: C is a feasible coloring for graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  $k \leftarrow 0$ ▷ number of colors used
  while $|\mathcal{V}| > 0$ **do**
    if $|\mathcal{E}| > 0$ **then**
      $MIS \leftarrow MIS_solver(\mathcal{G})$
      $\mathcal{G} \leftarrow (\mathcal{V} \setminus MIS, \mathcal{E}')$
    else
      $MIS \leftarrow \mathcal{V}$
      $\mathcal{G} \leftarrow \emptyset$
    end if
    $k \leftarrow k + 1$
    $C_k \leftarrow MIS$
  end while

This iterative decomposition highlights how VC problems can be reformulated as a sequence of MIS subproblems, making them particularly suitable for solution on neutral-atom quantum hardware. However, the greedy nature of the approach motivates the introduction of more advanced strategies capable of improving solution quality, which are discussed in the following sections.

3.2.3 The BBQ-MIS algorithm for Vertex Coloring

To improve the quality of the solutions to the VC problem, a hybrid algorithm, named BBQ-mIS [39], has been introduced. The method combines the natural representation of MIS problems on neutral-atom QPUs with a BB (Branch-and-Bound) framework. This approach exploits the probabilistic nature of quantum measurements, using the distribution of sampled bitstrings to explore the VC solution space.

In this setting, the graph structure is implicitly defined by the interactions between qubits, where each qubit represents a vertex. A coloring is constructed by iteratively assigning one color to a maximal independent set of vertices, while a BB strategy is used to guide the search toward solutions with a reduced number of colors.

Definition 2 (Maximal independent set)**.** *A mIS (maximal Independent Set) of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an independent set that is not properly contained in any other independent set of $\mathcal{G}$. In other words, adding any additional vertex to a maximal independent set would violate its independence.*

The proposed approach is particularly motivated by current hardware limitations, most notably the restricted number of available qubits. Direct mappings of large optimization problems onto quantum hardware are often infeasible, making problem decomposition essential. The BBQ-mIS algorithm addresses this challenge by partitioning the original problem into smaller subproblems that can be processed within the constraints of the quantum device, while maintaining a global optimization perspective through the classical control layer.

Compared to direct QUBO formulations, this approach avoids the introduction of auxiliary variables (e.g., one-hot encodings for colors), resulting in a significantly more efficient use of qubits. Instead, the method leverages the native capability of neutral-atom systems to encode independence constraints, allowing MIS subproblems to be solved directly at the hardware level.

Finally, the BB structure provides a systematic exploration of the solution space through three key operations: *branching*, *bounding*, and *pruning*. Each node of the search tree corresponds to a partial coloring, and nodes can be explored independently, making the approach naturally suited for parallel and distributed implementations.

Rational for MIS-Based Branch-and-Bound Exploration

A key aspect motivating the proposed approach lies in the probabilistic nature of quantum computation. When solving a MIS instance on a neutral-atom QPU, multiple measurements are required in order to obtain candidate solutions. These repeated measurements produce a distribution over bitstrings, which can be interpreted as a histogram of independent sets. In a simple strategy such as *Greedy-it-MIS*, the most frequently observed solution is selected as the MIS. However, the remaining samples in the distribution also contain valuable information, as they correspond to alternative independent sets that are obtained at no additional computational cost. This is particularly relevant since MIS solutions are not necessarily unique. Even when an optimal coloring contains a maximum independent set, there is no guarantee that it coincides with the most frequently sampled one. Furthermore, optimal VC solutions may not rely exclusively on maximum independent sets, but may instead include smaller independent sets.

Exploring all possible independent sets obtained from the quantum measurements would quickly become computationally intractable. Therefore, it is necessary to restrict

the search space while still preserving the ability to identify high-quality solutions. A key theoretical result supporting this reduction is given by Theorem 1 [116].

Theorem 1 (Optimal vertex coloring)**.** *Every graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ admits an optimal coloring in which at least one color class is a maximal independent set.*

Proof. Let $C = \{C_1, \dots, C_k\}$ be an optimal coloring of $\mathcal{G}$. Let I be a maximal independent set such that $C_1 \subseteq I$. Then the coloring

$$C' = \{I,\, C_2 \setminus I,\, \dots,\, C_k \setminus I\}$$

is still feasible and uses at most k colors, thus preserving optimality.

This result implies that the search for optimal VC solutions can be restricted to mISs without loss of optimality. This observation provides the foundation for the BB strategy adopted in the proposed approach.

Branch-and-Bound Framework

Building on the observations of the previous subsection, the exploration of maximal independent sets can be structured within a BB framework. The objective is to minimize the number of colors used in a feasible VC solution, that is, to determine the chromatic number $\chi(\mathcal{G})$ or a tight approximation thereof.

The resulting algorithm, referred to as BBQ-mIS, can be interpreted as an outer optimization layer that leverages the solutions generated by the quantum MIS solver. Instead of selecting a single independent set at each step, as in greedy approaches, the algorithm explores multiple candidate mISs obtained from the quantum measurement distribution, organizing them into a search tree.

The overall structure of the BB procedure is illustrated in Fig. 3.1. The BB tree is rooted at the original graph $\mathcal{G}$, and each node represents a partial coloring obtained by iteratively assigning colors to maximal independent sets and removing them from the graph.

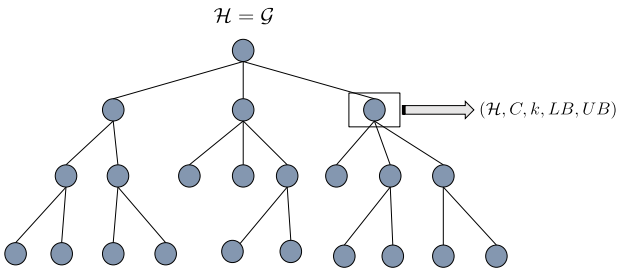
More precisely, each node is associated with a state described by the tuple

$$(\mathcal{H}, C, k, LB, UB),$$

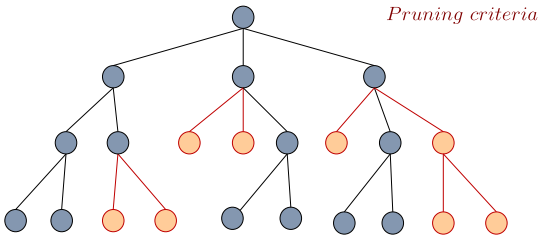
where:

- $\mathcal{H} = (\mathcal{U}, \mathcal{I})$ is the induced subgraph obtained after removing the vertices assigned in previous steps;
- C is a feasible (possibly partial) coloring, where colors correspond to previously selected mISs;
- $k = |C|$ is the number of colors used so far;

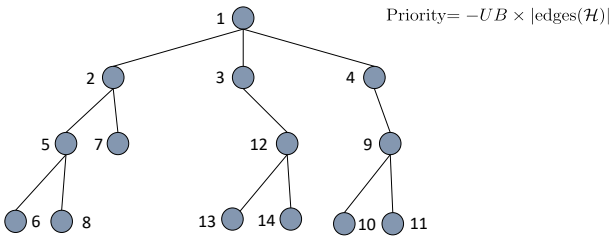
- LB is a lower bound on the minimum number of colors achievable from the current node;
- UB is an upper bound derived from feasible solutions.



(a) Node structure: induced subgraph $\mathcal{H}$, partial coloring C, color count k, and bounds LB and UB.



(b) Pruning eliminates non-improving, infeasible, or redundant subproblems to reduce the number of nodes explored.



(c) Node expansion is guided by a priority score favoring the most promising subproblems.

Figure 3.1: Branch-and-Bound tree structure, pruning strategy, and exploration policy underlying the BBQ-mIS algorithm.

As shown in Fig. 3.1a, each node encodes both the combinatorial structure of the remaining subproblem and the current state of the solution. From a given node, branching is performed by selecting one of the candidate maximal independent sets and generating a child node corresponding to the induced subgraph obtained after its removal.

The exploration of the tree is guided by pruning and node-selection strategies. In particular, pruning rules (Fig. 3.1b) discard nodes that cannot lead to improved solutions. At the same time, a priority score is assigned to each node to guide the exploration toward the most promising regions of the search space (Fig. 3.1c).

Branching

At each node of the BB tree, the MIS solver is invoked to generate a collection of maximal independent sets. Each maximal independent set $\tilde{S}$ defines a branching decision, corresponding to assigning one color to all vertices in $\tilde{S}$.

Given a node associated with graph $\mathcal{H} = (\mathcal{U}, \mathcal{I})$, branching on $\tilde{S}$ produces a child node with graph

$$\mathcal{H}' = (\mathcal{U} \setminus \tilde{S},\ \mathcal{I} \setminus \{(i,j) \mid i \in \tilde{S} \text{ or } j \in \tilde{S}\}),$$

and an updated coloring obtained by adding $\tilde{S}$ as a new color class. This process reduces the size of the remaining problem while preserving feasibility.

Bounding

At each node of the BB tree, both a LB (lower bound) and an UB (Upper Bound) on the chromatic number are computed to guide the exploration. Let the node be at depth k, corresponding to the number of colors already assigned, and let $\mathcal{H} = (\mathcal{U}, \mathcal{I})$ be the induced subgraph associated with the node.

Lower Bound (LB). A valid lower bound on the chromatic number $\chi(\mathcal{G})$ is given by

$$LB = k + LB(\mathcal{H}), \tag{3.11}$$

where $LB(\mathcal{H})$ is a lower bound on $\chi(\mathcal{H})$.

To obtain a tight estimate, multiple spectral bounds are evaluated:

- **Hoffman bound** [117]:

$$LB_H(\mathcal{H}) = 1 - \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}, \tag{3.12}$$

- **Inertial (Elphick–Wocjan) bound** [118]:

$$LB_{EW}(\mathcal{H}) = 1 + \max\left(\frac{n^+}{n^-},\ \frac{n^-}{n^+}\right), \tag{3.13}$$

- **Edwards–Elphick bound** [119]:

$$LB_{EE}(\mathcal{H}) = \frac{|\mathcal{U}|}{|\mathcal{U}| - \bar{d}}, \tag{3.14}$$

where A is the adjacency matrix of $\mathcal{H}$, $\lambda_{\max}$ and $\lambda_{\min}$ are its largest and smallest eigenvalues, n^+ and n^- denote the number of positive and negative eigenvalues, and $\bar{d}$ is the average degree.

The combined lower bound is defined as

$$LB(\mathcal{H}) = \max\left(\lfloor LB_H \rfloor,\ \lfloor LB_{EW} \rfloor,\ \lfloor LB_{EE} \rfloor\right). \tag{3.15}$$

Upper Bound (UB). An upper bound is derived from the current partial coloring and the structure of the remaining subgraph. In particular, a worst-case extension of the current solution is obtained by assigning distinct colors to the remaining vertices in $\mathcal{H}$, leading to

$$UB = d + \deg_{\max}(\mathcal{H}) + 1, \tag{3.16}$$

where $\deg_{\max}(\mathcal{H})$ denotes the maximum degree of $\mathcal{H}$.

The bounds LB and UB are used to guide the exploration of the BB tree and to prune nodes that cannot lead to improved solutions.

Pruning

The exploration of the BB tree is significantly reduced through pruning rules that eliminate nodes unlikely to improve the current best solution.

- *Non-improving pruning*: a node is discarded if its lower bound is greater than or equal to the best known coloring solution;
- *Unfeasibility pruning*: candidate sets that are not independent or not maximal are discarded, ensuring consistency with Theorem 1;
- *Redundancy pruning*: nodes corresponding to identical subgraphs are not revisited. To detect duplicates, each graph $\mathcal{H} = (\mathcal{U}, \mathcal{I})$ is associated with a fingerprint

$$fp(\mathcal{H}) = \sum_{u \in \mathcal{U}} 2^u,$$

and new nodes are generated only if their fingerprint has not been previously encountered.

Node Selection

The order in which nodes are explored has a significant impact on performance. To balance depth-first and breadth-first exploration, each node is assigned a priority score defined as

$$\text{score}(\mathcal{H}) = -UB \times |\mathcal{J}|,$$

where $|\mathcal{J}|$ is the number of edges in the current subgraph. This criterion favors nodes that simultaneously exhibit low connectivity and promising upper bounds.

At early stages, the exploration tends to proceed in depth, quickly identifying feasible solutions comparable to those obtained by greedy strategies. As the search progresses, the priority mechanism shifts the focus toward more promising regions of the search space.

Stopping Criterion

In principle, the algorithm terminates when all branches of the BB tree have been explored. However, the number of maximal independent sets grows rapidly with the graph size [120], making exhaustive exploration impractical.

To ensure tractability, a stopping criterion is introduced by limiting the maximum number of explored nodes. In the experiments, this limit is set to 50, and the best feasible coloring found within this budget is returned as the final solution.

3.2.4 Comparison between Greedy-it-MIS and BBQ-mIS

The performance of *Greedy-it-MIS* and BBQ-mIS is evaluated within the hybrid quantum-classical framework described in the previous sections. The MIS solver is implemented through a quantum emulation using the *Pulser* library[1], adopting the QAOA [121, 69, 6] to optimize the pulse parameters. The input graph $\mathcal{G}$ is embedded into a Rydberg atom register, where the Rabi frequency Ω is set according to the blockade radius r_b to reproduce the graph connectivity, while detuning and pulse shapes follow the specifications of the *Pasqal Chadoq2* prototype [55]. The variational parameters correspond to pulse durations, optimized via a classical Nelder–Mead solver, and multiple measurements are performed to obtain candidate mISs.

Due to the cost of simulating quantum dynamics, this approach is limited to graphs with up to 15 vertices. For larger instances, the MIS solver is replaced by classical heuristics while preserving the same algorithmic structure. The dataset consists of 120 unit-disk graphs with $n \in \{10, \dots, 15\}$ vertices, with embeddings generated using the DEN model introduced in Chapter 2. All experiments are executed on an IBM Power9

[1] https://pulser.readthedocs.io

cluster with MPI-based parallelization, requiring approximately 8000 core-hours. Optimal solutions are computed using the *Gurobi* solver [70].

The results are summarized in Fig. 3.2. The BBQ-mIS algorithm consistently matches the optimal solutions for all instances, while *Greedy-it-MIS* produces suboptimal colorings in 38 out of 120 cases, with a gap of up to four additional colors. These results highlight the benefit of exploring multiple mISs instead of relying on a single greedy selection.

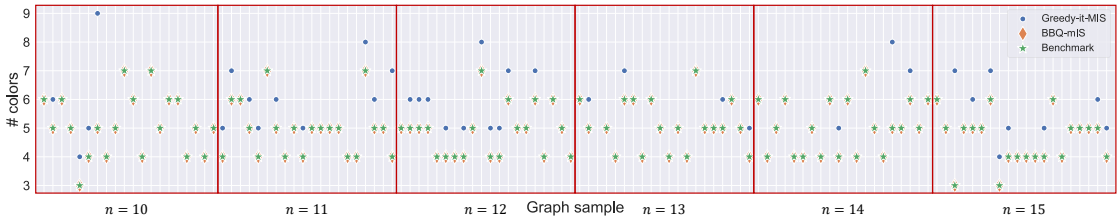


Figure 3.2: VC performance of *Greedy-it-MIS* and BBQ-mIS on all graph instances, with *Gurobi* as the optimal reference and *Pulser* as the MIS solver.

From a computational perspective, the size of the BB tree remains limited: in all instances, BBQ-mIS converges within at most 20 nodes, well below the imposed limit of 50. This behavior is due to the relatively small number of mISs in the considered graphs. The estimated runtime is dominated by quantum sampling: assuming a worst-case exploration of 50 nodes, approximately 255000 samples are required, corresponding to about 14 hours at a 5 Hz sampling rate. In practice, the number of explored nodes ranges between 8 and 20, resulting in execution times between 2 and 6 hours.

To assess scalability, additional experiments are performed on larger graphs by replacing the quantum solver with a classical MIS heuristic. The results, reported in Table 3.1, show that BBQ-mIS maintains competitive performance and consistently outperforms *Greedy-it-MIS*, while remaining close to the optimal solutions.

Vertices	Greedy-itMIS	BBQ-mIS	Optimum
25	4 colors	3 colors	3 colors
35	5 colors	5 colors	5 colors
45	8 colors	6 colors	6 colors
55	7 colors	7 colors	7 colors
65	8 colors	8 colors	7 colors
75	10 colors	9 colors	8 colors
85	10 colors	9 colors	8 colors
95	11 colors	9 colors	8 colors

Table 3.1: Results on larger instances using classical MIS heuristics.

Overall, BBQ-mIS provides a significant improvement over *Greedy-it-MIS*, achieving optimal or near-optimal solutions while maintaining a manageable computational

cost. The integration of quantum sampling within a BB framework enables an effective exploration of the solution space, suggesting good scalability prospects for larger quantum devices.

3.2.5 Quantum-Classical Branch-and-Price for Vertex Coloring

The methods introduced so far, namely Greedy-it-MIS and BBQ-mIS, exploit the ability of neutral-atom QPUs to efficiently generate mISs in order to construct feasible solutions to the VC problem. While these approaches demonstrate the effectiveness of combining quantum sampling with classical optimization, they also exhibit important limitations.

On the one hand, Greedy-it-MIS relies on a single sequence of independent sets and therefore does not explicitly target the minimization of the number of colors. On the other hand, BBQ-mIS improves solution quality by exploring multiple mISs within a BB framework, but this comes at the cost of an increasingly expensive tree exploration, which limits scalability on larger or more complex graph instances.

Moreover, both approaches rely on quantum routines based on the QAOA, which, while flexible, introduce a significant computational overhead due to the need for classical optimization of variational parameters. This limitation becomes particularly relevant when the quantum solver is invoked repeatedly within an outer optimization loop.

To overcome these limitations, we introduce a more advanced hybrid framework, referred to as QCBP (Quantum-Classical Branch-and-Price), which builds upon the previous methods while addressing both their algorithmic and quantum-related bottlenecks.

From the optimization perspective, the key idea is to move from a purely tree-based exploration, as in BBQ-mIS, to a decomposition-based approach inspired by BP. In this formulation, the VC problem is expressed as a set-partitioning ILP [122], where each column corresponds to an IS (Independent Set) of the graph. Instead of explicitly enumerating all possible independent sets, which is computationally prohibitive, the algorithm dynamically generates them through a CG procedure.

From the quantum perspective, we replace the variational QAOA with an analog evolution based on the QAA. This choice is motivated by the intrinsic compatibility between neutral-atom hardware and adiabatic dynamics, as discussed in the previous section. The use of QAA eliminates the need for iterative parameter optimization and enables a more efficient sampling of low-energy configurations, significantly reducing the quantum overhead.

An additional advantage of this approach is its improved robustness with respect to non-unit-disk embeddings. While previous methods rely on high-quality unit-disk representations to ensure accurate correspondence between the problem graph and the physical interactions, the QCBP framework leverages quantum sampling as a heuristic within a broader optimization scheme. As a result, even when the embedding deviates from the ideal unit-disk structure, the algorithm can still exploit the generated independent sets to progressively improve the solution.

The resulting framework combines the strengths of the previously introduced methods. The structured exploration provided by BP improves scalability and optimality, while the use of QAA-based quantum routines ensures efficient generation of high-quality independent sets. In this way, QCBP achieves a better trade-off between solution quality and computational cost, particularly on general graph instances where both greedy and tree-based approaches become less effective.

In the following, we detail the formulation of the VC problem within the BP framework and describe how quantum routines are integrated into the column generation process.

Formal Foundations of the Branch-and-Price Framework

To formalize the methodology introduced above, we briefly recall the structure of BP algorithms [123, 124, 125, 126], which combine CG with BB to efficiently solve large-scale ILPs.

We consider a generic binary ILP of the form

$$P: \quad \min \quad c^\top x \tag{3.17}$$
$$\text{s.t.} \quad Ax \geq b, \tag{3.18}$$
$$x \in \{0,1\}^n, \tag{3.19}$$

where $x \in \{0,1\}^n$ is the vector of decision variables, $A \in \mathbb{R}^{m \times n}$ is the constraint matrix, $b \in \mathbb{R}^m$ the right-hand side, and $c \in \mathbb{R}^n$ the cost vector.

Solving P directly is typically intractable when n is large. Therefore, BP relies on the linear relaxation

$$P_{LR}: \quad \min \quad c^\top x \tag{3.20}$$
$$\text{s.t.} \quad Ax \geq b, \tag{3.21}$$
$$x \geq 0, \tag{3.22}$$

which provides a lower bound on the optimal objective value.

Due to the potentially huge number of variables, CG is used to solve P_{LR} implicitly. Instead of considering all variables, a restricted subset is maintained, defining the RMP (Reduced Master Problem) at iteration k:

$$\text{RMP}^{(k)}: \quad \min \quad (c^{(k)})^\top x^{(k)} \tag{3.23}$$
$$\text{s.t.} \quad A^{(k)} x^{(k)} \geq b, \tag{3.24}$$
$$x^{(k)} \geq 0, \tag{3.25}$$

where $A^{(k)}$ contains only the currently selected columns.

After solving the RMP, the corresponding dual solution $y^{(k)} \in \mathbb{R}^m$ is used to guide the generation of new variables through the PSP (Pricinig Sub-Problem). The reduced

cost associated with a candidate column A_j is given by

$$\bar{c}_j = c_j - A_j^\top y^{(k)}. \tag{3.26}$$

The PSP seeks columns with negative reduced cost, i.e.,

$$\bar{c}_j < 0,$$

which can potentially improve the current solution. If such columns are found, they are added to the RMP and the process is repeated. Otherwise, optimality for the linear relaxation is reached.

Since the solution obtained through CG is generally fractional, a BB procedure is required to recover integrality. At each node of the search tree, the problem is refined through:

- **Branching:** a fractional decision is selected and the problem is partitioned into subproblems by enforcing complementary constraints (e.g., fixing a variable or imposing structural conditions).
- **Bounding:** each subproblem is solved via CG, providing a valid LB. Nodes whose bound exceeds the best known feasible solution (UB) are pruned.

In parallel, primal heuristics are used to construct feasible integer solutions from the current set of variables, improving the UB and accelerating convergence.

The BP procedure recursively explores the resulting search tree until all nodes are either solved or pruned, yielding an optimal or near-optimal integer solution.

This general framework provides the foundation for the hybrid approach developed in this chapter, where quantum routines are integrated within the PSP to efficiently generate promising variables for the VC problem.

Vertex Coloring Formulations

Building on the BP framework introduced in the previous subsection, we now specialize the formulation to the VC problem, highlighting the models that underpin the proposed hybrid approach. We consider an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. The objective is to partition the vertex set into the minimum number of ISs, corresponding to the chromatic number $\chi(\mathcal{G})$.

Compact ILP Formulation. A standard formulation introduces binary variables x_{ic} indicating whether vertex $i \in \mathcal{V}$ is assigned color c, and variables y_c indicating whether color c is used. Given an upper bound m on $\chi(\mathcal{G})$, the problem reads:

$$\min \quad \sum_{c=1}^{m} y_c \tag{3.27a}$$

$$\text{s.t.} \quad \sum_{c=1}^{m} x_{ic} = 1 \quad \forall i \in \mathcal{V}, \tag{3.27b}$$

$$x_{ic} + x_{jc} \leq y_c \quad \forall (i,j) \in \mathcal{E},\ \forall c, \tag{3.27c}$$

$$x \in \{0,1\}^{n \times m}, \tag{3.27d}$$

$$y \in \{0,1\}^{m} \tag{3.27e}$$

This formulation is widely used due to its simplicity, but it is known to have a weak linear relaxation, leading to poor lower bounds within a BP framework. Moreover, its direct encoding requires $n \times m$ binary variables, which is not suitable for neutral-atom QPUs with limited qubit resources.

Set-partitioning Formulation. To better exploit decomposition techniques, we adopt a set-partitioning formulation in which each variable corresponds to an IS. Let $\mathcal{S}$ denote the set of all independent sets of $\mathcal{G}$, and introduce binary variables λ_S indicating whether $S \in \mathcal{S}$ is selected. The formulation reads:

$$\min \quad \sum_{S \in \mathcal{S}} \lambda_S \tag{3.28a}$$

$$\text{s.t.} \quad \sum_{S \ni i} \lambda_S = 1 \quad \forall i \in \mathcal{V}, \tag{3.28b}$$

$$\lambda_S \in \{0,1\} \quad \forall S \in \mathcal{S} \tag{3.28c}$$

Constraint (3.28b) ensures that each vertex is covered exactly once, while the independence of each S guarantees a valid coloring.

Due to the exponential size of $\mathcal{S}$, this formulation is solved using CG. The RMP is defined over a restricted subset $\mathcal{S}' \subset \mathcal{S}$, and its linear relaxation (denoted as VCPF-R) is obtained by replacing (3.28c) with non-negativity constraints:

$$\min \quad \sum_{S \in \mathcal{S}'} \lambda_S \tag{3.29a}$$

$$\text{s.t.} \quad \sum_{S \ni i} \lambda_S = 1 \quad \forall i \in \mathcal{V}, \tag{3.29b}$$

$$\lambda_S \geq 0 \quad \forall S \in \mathcal{S}' \tag{3.29c}$$

Dual Formulation and Pricing. Solving the relaxed problem yields dual variables π_i associated with constraints (3.29b). The corresponding dual formulation is:

$$\max \quad \sum_{i \in \mathcal{V}} \pi_i \tag{3.30a}$$

$$\text{s.t.} \quad \sum_{i \in S} \pi_i \leq 1 \quad \forall S \in \mathcal{S}', \tag{3.30b}$$

$$\pi_i \in \mathbb{R} \quad \forall i \in \mathcal{V} \tag{3.30c}$$

These dual variables define the PSP, which consists of identifying an independent set with maximum reduced cost. This corresponds to a MWIS problem:

$$\max_{S \in \mathcal{S}} \sum_{i \in S} \pi_i. \tag{3.31}$$

This observation is central to the hybrid approach developed in this chapter. As discussed previously, MWIS problems can be efficiently addressed on neutral-atom QPUs through analog quantum evolution. Therefore, the PSP provides a natural interface between the classical BP framework and the quantum routines.

Hybrid Column Generation with Quantum Pricing

Building upon the set-partitioning formulation introduced above, the proposed QCBP framework integrates a HCG (Hybrid Column Generation) procedure within the BB scheme. The goal of this phase is to efficiently approximate the solution of the relaxed problem VCPF-R while generating only a limited subset of relevant independent sets, as illustrated in Fig. 3.3.

At each iteration, the algorithm solves the RMP defined over a restricted set of columns $\mathcal{S}' \subseteq \mathcal{S}$. The initial set $\mathcal{S}'$ is typically composed of simple feasible independent sets, such as singletons. Once the RMP is solved, the corresponding dual solution $\pi^* = (\pi_i^*)_{i \in \mathcal{V}}$ is used to guide the PSP.

As discussed in the previous subsection, the PSP consists of identifying a column violating the dual constraint (3.30b), that is, an independent set $S^* \in \mathcal{S} \setminus \mathcal{S}'$ such that

$$\sum_{i \in S^*} \pi_i^* > 1. \tag{3.32}$$

Equivalently, the PSP can be formulated as a MWIS problem, where the vertex weights are given by the dual variables:

$$\max \left\{ \sum_{i \in S} \pi_i^* \ : \ S \subseteq \mathcal{V},\ S \text{ is an independent set} \right\}.$$

This problem can be expressed as the following ILP:

$$\min \quad \sum_{i=1}^{n} \pi_i^* z_i \tag{3.33a}$$

$$\text{s.t.} \quad z_i + z_j \leq 1 \quad \forall (i,j) \in \mathcal{E}, \tag{3.33b}$$

$$z \in \{0,1\}^n \tag{3.33c}$$

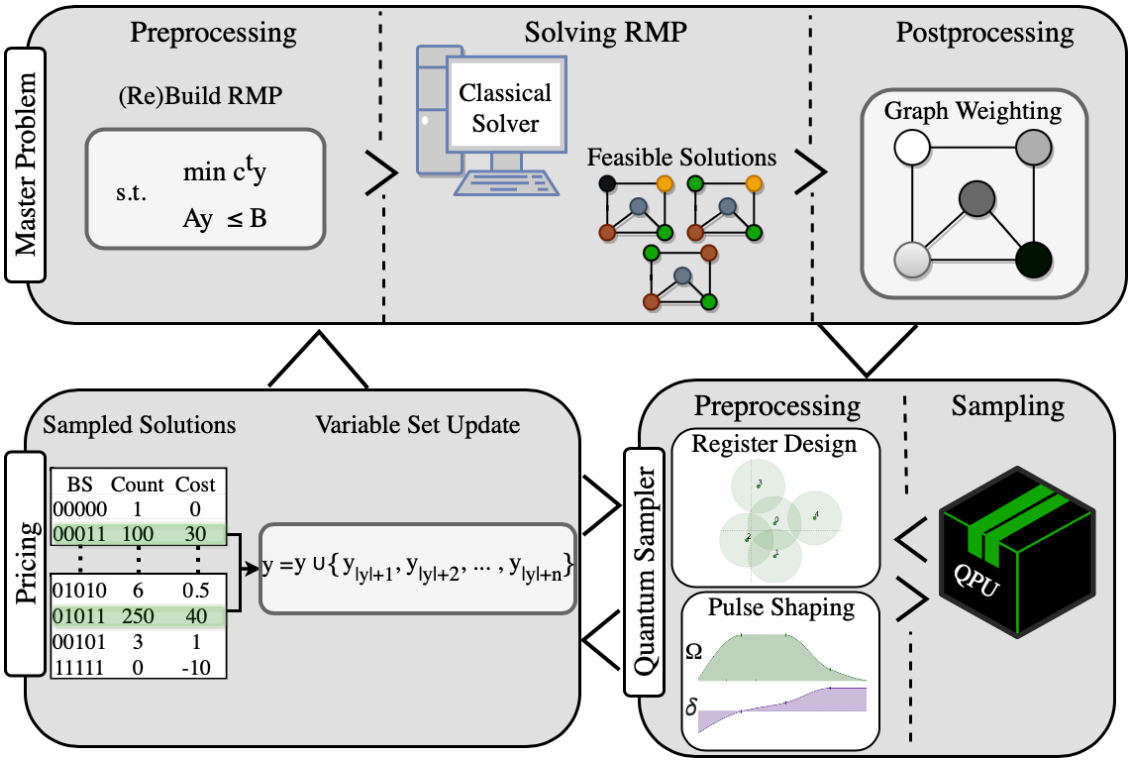


Figure 3.3: Workflow of the hybrid classical-quantum column generation procedure [92].

The MWIS problem is NP-hard and represents the computational bottleneck of the CG procedure. In the proposed framework, it is addressed using neutral-atom QPUs, leveraging the mapping between independent sets and Rydberg blockade configurations discussed previously. In contrast to approaches based on QAOA, we employ the QAA, which avoids variational parameter optimization and enables efficient sampling of low-energy configurations. Repeated measurements naturally produce multiple candidate independent sets, allowing several negative reduced-cost columns to be identified within a single iteration and accelerating convergence.

This hybrid strategy follows the approach proposed in [92], see Fig. 3.3, adapted here by integrating QAA within a full BP framework. Due to the approximate nature of both embedding and quantum evolution, the quantum routine may fail to identify all improving columns, particularly in later iterations or for non-unit-disk graphs. To ensure correctness, if no improving column is found, the MWIS is solved exactly using a classical ILP solver (i.e., *GLPK* [82]), guaranteeing proper termination of the CG procedure.

The HCG procedure is embedded within the BB framework, where generated columns are shared across nodes, improving both lower bounds and feasible solutions. Overall, this integration combines the scalability of CG, the optimality guarantees of BB, and the sampling capabilities of neutral-atom quantum devices, enabling a more effective

solution strategy for VC problems.

Primal Heuristic

At every node of the BP tree, a primal heuristic is applied to quickly derive a feasible integer solution from the pool of columns, i.e. the independent sets produced throughout the CG phase. Since optimality is enforced by the BB procedure, solving the RMP to integrality at every node is unnecessary. Instead, we adopt a fast greedy strategy that builds a valid coloring directly from the current column pool.

Given the subgraph $\mathcal{H} = (\mathcal{U}, \mathcal{J})$ associated with a BB node, the heuristic operates as follows. The vertices are first ordered by decreasing degree, prioritizing those that are more constrained. At each iteration, the highest-degree vertex v is selected, and among the available independent sets, the largest set containing v is chosen to define a new color class. All vertices belonging to this set are then removed from $\mathcal{H}$, and the procedure is repeated until all vertices are assigned to a color.

This greedy construction yields a feasible coloring that provides a local upper bound $\mathrm{UB}_{\mathrm{loc}}$ on the chromatic number at the current node. Importantly, the heuristic relies exclusively on the independent sets already generated through the HCG procedure, avoiding additional calls to either quantum or classical optimization routines. As a result, it introduces negligible computational overhead while significantly improving the effectiveness of the bounding and pruning mechanisms within the BB tree.

The Overall Hybrid QCBP Algorithm

The overall workflow of the proposed QCBP algorithm is summarized in Fig. 3.4. The procedure begins from an initial RMP and iteratively applies HCG, where quantum routines solve the PSP and generate promising independent sets. Once no improving columns are identified, the algorithm transitions to the BB phase: a primal heuristic supplies feasible solutions and upper bounds, while branching decisions progressively refine the search space. Each node in the search tree corresponds to fixing a color class associated with a maximal independent set, thereby reducing the problem size and guiding exploration toward integer optimality.

The QCBP framework is designed by combining and extending the most effective elements of BBQ-mIS and HCG, with the dual objective of improving solution quality and reducing the number of required QPU calls. To motivate this design, it is instructive to first examine the limitations of each predecessor approach. The BBQ-mIS approach, while effective in exploring the solution space, incurs a high quantum overhead. Each BB node requires multiple executions of QAOA, including variational parameter optimization: in previous implementations, up to 50 optimization steps were performed, each requiring 200 samples, plus an additional 200 200 samples for final evaluation, yielding up to

$$200 \times 50 + 200 = 10{,}200$$

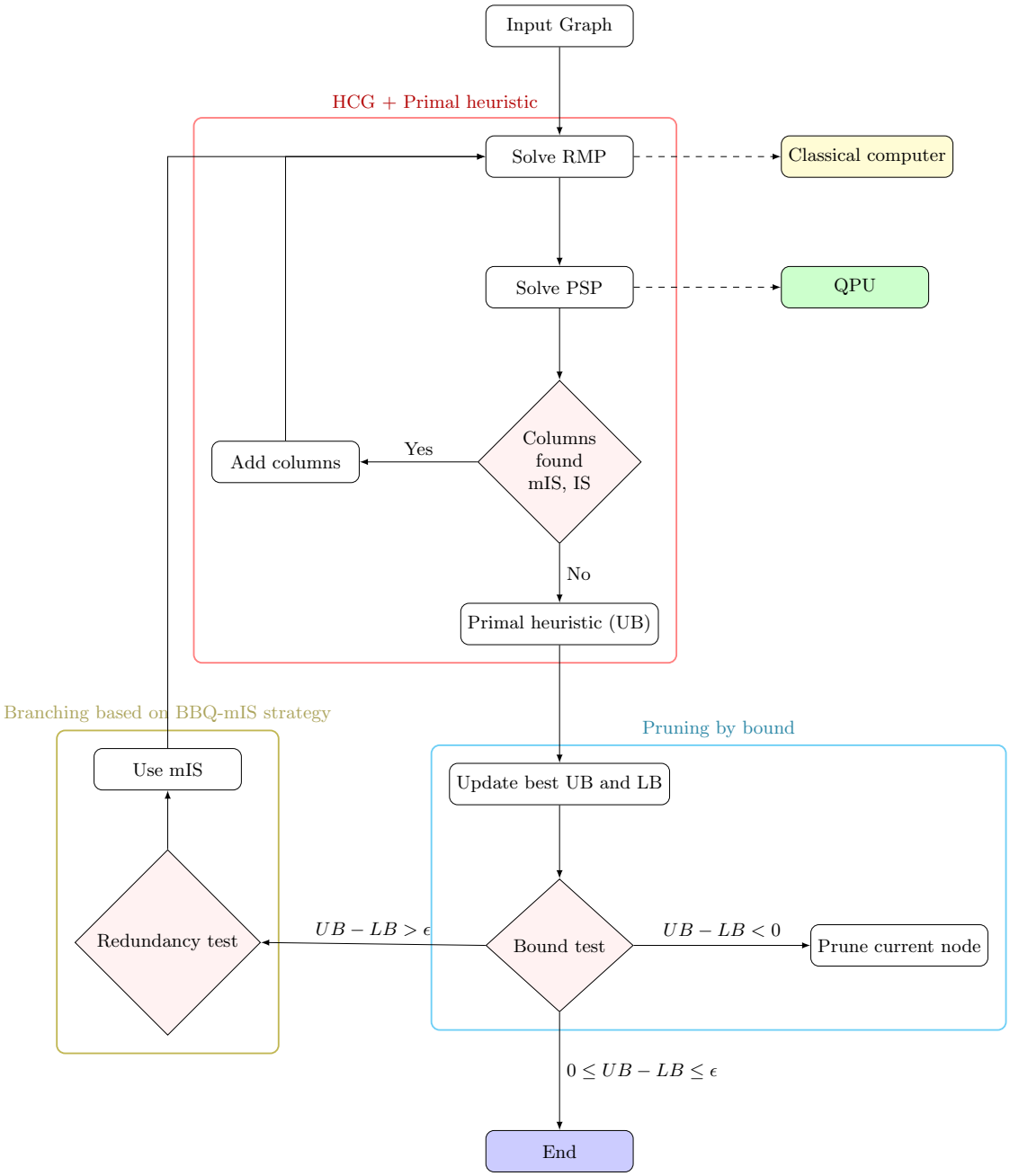


Figure 3.4: Hybrid Quantum-Classical Branch-and-Price Algorithm (QCBP).

QPU samples per node. Compounding this, relatively loose lower bounds result in larger BB trees and a correspondingly higher overall computational cost.

Hybrid CG, by contrast, avoids variational optimization entirely by relying on QAA, substantially reducing quantum resource requirements. Its main limitation, however,

lies in solution quality: in the absence of a global combinatorial search, colorings produced for non-unit-disk graphs can be suboptimal.

Minimizing QPU usage is a central design concern for both simulation and hardware contexts. In quantum emulation, cost grows rapidly with qubit count due to the numerical complexity of integrating the Schrödinger equation, while remaining relatively insensitive to the number of measurement shots. On real hardware, execution time is largely dictated by the sampling rate (typically on the order of 1 Hz), making the number of QPU calls the dominant bottleneck. Reducing quantum evaluations therefore translates directly into significant computational savings in both regimes.

The QCBP framework addresses these limitations through two key mechanisms. First, it derives tighter lower bounds from the relaxed RMP, improving pruning efficiency within the BB tree. Second, it eliminates the need for QAOA in the branching phase by directly reusing the maximal independent sets generated during quantum-assisted CG iterations. As a result, each QPU call contributes simultaneously to column generation and tree exploration, avoiding redundant quantum computations.

Operationally, at each CG iteration the algorithm performs: (i) solution of the relaxed RMP, (ii) solution of the MWIS pricing problem via QAA, and (iii) a classical fallback using an ILP solver (e.g., *GLPK* [82]) if no improving column is found. The quantum routine acts as a heuristic pricing oracle, while the classical solver is invoked only to certify pricing optimality. Since this fallback occurs exclusively in late iterations and on reduced subproblems, its computational overhead remains negligible. Once column generation terminates, the primal heuristic constructs a feasible coloring to establish an upper bound, while the relaxed RMP solution provides a valid lower bound. The BB phase then proceeds by branching on maximal independent sets, generating subproblems over reduced graphs. Feasibility is ensured by construction, and nodes are pruned using both RMP-based and classical bounds. The search continues until one of the following termination criteria is met: a predefined node limit is reached, no further branching is possible, or optimality is certified when upper and lower bounds coincide. By combining quantum-assisted column generation with efficient BB exploration, QCBP achieves a more favorable balance between solution quality and quantum resource usage than either predecessor. The following section provides a comprehensive experimental evaluation comparing all three approaches.

3.2.6 Comparison of Hybrid Quantum-Classical Methods for Vertex Coloring

In this section, we evaluate the performance of the proposed methods using both classical emulation and real quantum hardware, namely the Orion Alpha neutral-atom QPU. The analysis focuses on three key aspects: optimality, quantum resource efficiency (in terms of sampling cost), and scalability.

We first consider a benchmark dataset of 140 VC instances to assess the behavior under noiseless conditions, using classical emulation of the QAA and QAOA quantum routines in the corresponding QCBP/HCG and BBQ-mIS algorithms. The dataset includes 20 graphs for each value of the number of vertices $n \in \{10, 11, 12, 13, 14, 15, 16\}$. Among these instances, 78 admit a unit-disk representation fully compliant with the hardware constraints, while the remaining 62 correspond to non-unit-disk graphs, exhibiting slight deviations from the ideal geometric model. The inclusion of such instances is motivated by the need to assess the robustness of the proposed approach under realistic conditions, where perfect embeddings may not always be available.

In addition to the emulation study, we construct a second benchmark tailored to experiments on real quantum hardware. This dataset consists of a smaller set of UD graphs with increasing size: 3 instances with $n = 10$ vertices, 1 instance with $n = 20$, 2 instances with $n = 30$, and 1 instance with $n = 40$ vertices. Due to the exponential cost of classical simulation, instances with more than 10 vertices cannot be efficiently emulated, making hardware execution necessary. These experiments are used to evaluate the quality of the solutions obtained from the Orion Alpha QPU, by analyzing whether the sampled configurations correspond to high-quality candidate colorings.

Although a systematic noise sensitivity analysis falls outside the scope of the present work, the experimental design provides substantive indirect evidence regarding the robustness of the proposed methods. The deliberate inclusion of non-UD instances in the emulation dataset introduces controlled structural deviations from the ideal interaction model, enabling a principled assessment of algorithmic performance under imperfect graph embeddings — a practically relevant condition in near-term quantum computing deployments. Complementarily, the experiments conducted on the Orion Alpha QPU inherently subject the framework to realistic hardware noise, coherence limitations, and device imperfections characteristic of current NISQ processors. Taken together, these two experimental dimensions provide a rigorous, if indirect, characterization of the resilience of the QCBP framework under non-ideal conditions. A comprehensive noise-aware analysis, incorporating explicit error models and mitigation strategies, constitutes a natural and well-motivated direction for future investigation.

Results from Quantum Emulation

We first evaluate the optimality performance of QCBP in comparison with BBQ-mIS and HCG, using the exact ILP solver *GLPK* as reference for the optimum of each VC problem. The percentage of optimally solved instances is reported in Table 3.2. Overall, QCBP clearly outperforms the competing methods, failing in only 3 out of 140 instances. Moreover, it exhibits strong robustness with respect to non-unit-disk graphs: while the performance of BBQ-mIS drops from 92.3% to 72.6% and HCG degrades significantly to 40.3%, QCBP maintains a success rate above 96%.

To further quantify solution quality, Table 3.3 reports the average relative optimality gap, defined as $(c - \chi(\mathcal{G}))/\chi(\mathcal{G})$, where c denotes the computed coloring. QCBP

Table 3.2: Percentage of benchmark instances solved to optimality by each method (QCBP, BBQ-mIS, and HCG), aggregated across the full evaluation dataset.

Unit-Disk	BBQ-mIS	HCG	QCBP
True	0.923	0.897	0.987
False	0.726	0.403	0.968
Total	0.836	0.679	0.979

consistently achieves near-optimal solutions, with zero gap in most cases and a maximum deviation of only 0.033. In contrast, BBQ-mIS and HCG exhibit significantly larger gaps, reaching up to 0.083 and 0.208, respectively.

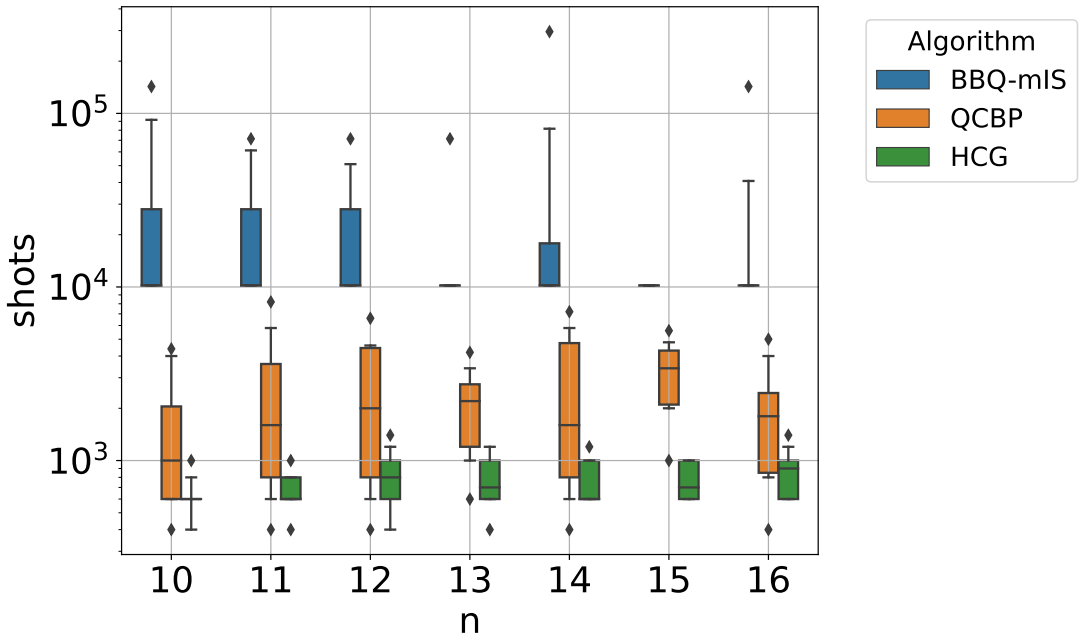
Table 3.3: Average relative optimality gap (%) achieved by each method (QCBP, BBQ-mIS, and HCG), disaggregated by graph size n. Lower values indicate solutions closer to the certified optimum.

	BBQ-mIS	HCG	QCBP
$n = 10$	0.033	0.050	0.0
$n = 11$	0.050	0.067	0.0
$n = 12$	0.083	0.117	0.0
$n = 13$	0.017	0.108	0.033
$n = 14$	0.075	0.092	0.0
$n = 15$	0.050	0.150	0.0
$n = 16$	0.083	0.208	0.017

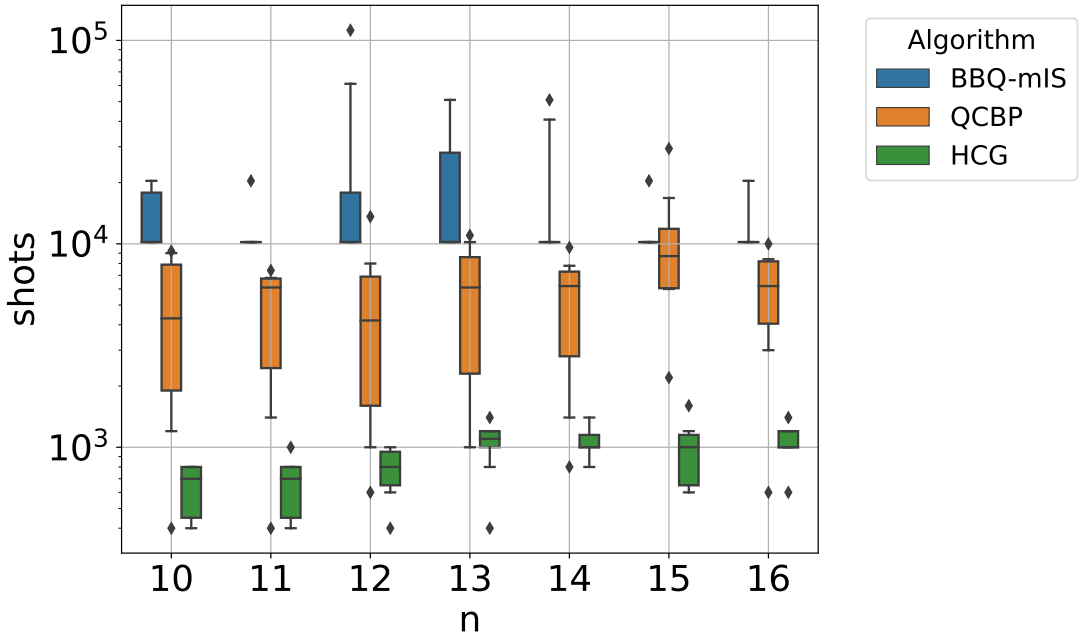
Beyond solution quality, we analyze quantum resource efficiency in terms of the number of required QPU samples (shots). The distributions are reported in Fig. 3.5, separately for unit-disk (Fig. 3.5a) and non-unit-disk (Fig. 3.5b) instances. Hybrid CG requires the fewest shots (typically 10^2–10^3), but at the cost of reduced accuracy. Conversely, BBQ-mIS achieves better solution quality but requires significantly more samples (10^4–10^5). The proposed QCBP approach strikes a favorable balance, achieving high accuracy with a substantially lower number of shots. This advantage is particularly evident on unit-disk instances, where the shot distribution of QCBP remains below 10^4, much closer to HCG than to BBQ-mIS. This reduction in sampling requirements is crucial for practical deployment on current quantum hardware.

We next analyze the behavior of the classical components of QCBP, focusing on the number of BB nodes and the use of the exact ILP solver within the HCG procedure. The distribution of generated, explored, and pruned nodes is reported in Fig. 3.6. A node is considered *explored* when the HCG routine is invoked. As shown in Fig. 3.6a–3.6c, BBQ-mIS generates and explores significantly more nodes due to weaker lower bounds, resulting in higher variability and larger search trees (up to 60 generated nodes and 9 explored nodes). In contrast, QCBP benefits from tighter bounds derived from the relaxed RMP, enabling more aggressive pruning (Fig. 3.6b) and a nearly constant number

of explored nodes, thus improving computational efficiency.

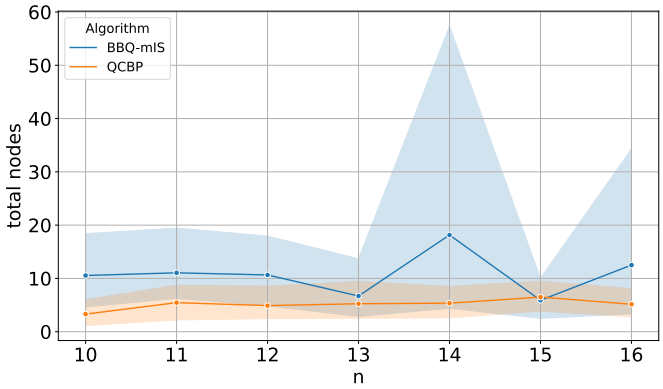


(a) QPU shot consumption per method on UD graph instances.

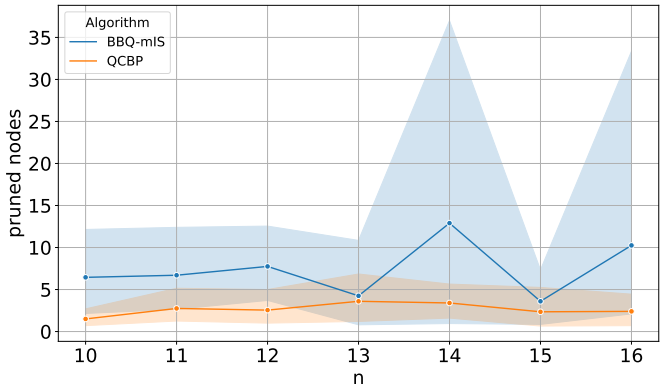


(b) QPU shot consumption per method on non-UD graph instances.

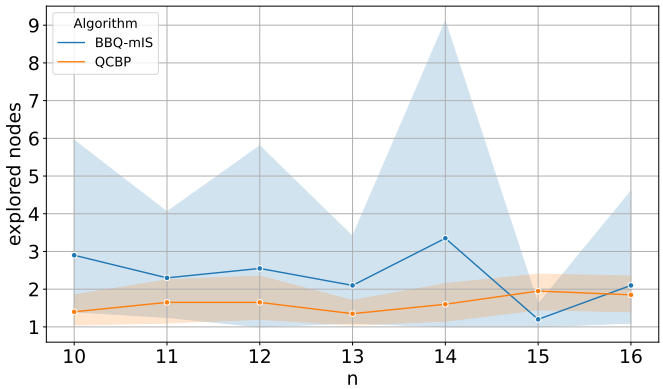
Figure 3.5: QPU shot consumption across methods, compared on UD and non-UD graph instances.



(a) Total generated BB nodes.



(b) Pruned BB nodes.



(c) Explored BB nodes.

Figure 3.6: Branch-and-Bound node distributions grouped by graph size n.

Finally, Table 3.4 reports the median number of calls to the exact ILP solver within the HCG procedure. These calls are required to guarantee correctness when the quantum routine fails to identify improving columns. As expected, the number of calls is higher for non-unit-disk instances, where the embedding deviates from the ideal interaction model. For unit-disk graphs, the number of calls remains limited (between 1 and

7), confirming the effectiveness of the quantum routine. For non-unit-disk graphs, the number increases (up to 21.5), highlighting the impact of imperfect embeddings. Nevertheless, these calls occur primarily in later iterations and on reduced subproblems, resulting in a limited overall computational overhead.

Table 3.4: Median number of ILP fallback calls in QCBP, disaggregated by graph size n and instance type (UD vs. non-UD).

Unit-Disk	$n = 10$	$n = 11$	$n = 12$	$n = 13$	$n = 14$	$n = 15$	$n = 16$
True	1.0	4.0	3.5	4.0	4.5	7.0	3.0
False	11.0	13.5	10.0	13.0	14.5	21.5	12.5

Evaluation on Real Neutral-Atom QPU

Given the promising results obtained under classical emulation, the proposed QCBP framework was further evaluated on the Orion Alpha neutral-atom QPU using the unit-disk benchmark instances introduced above. The purpose of this study is twofold. First, it aims to assess whether QCBP remains effective in the presence of hardware noise when solving the MWIS pricing problem within the HCG procedure. Second, it evaluates whether the independent sets sampled by the QPU are sufficiently structured to support not only column generation, but also the branching phase of the algorithm, where maximality becomes important. More generally, these experiments demonstrate that the entire workflow can operate in closed loop, alternating efficiently between direct QPU access and classical optimization steps.

From the perspective of the pricing step, the relevant outputs are independent sets with negative reduced cost, i.e.,

$$1 - \sum_{i \in \mathcal{V}} \pi_i^* z_i < 0.$$

Figure 3.7 reports the percentage of such useful pricing solutions identified through the QPU, relative to the total number of sampled bitstrings, grouped by the size n' of the MWIS subproblem. Although the original graph has size n, the pricing problem may involve a reduced graph of size $n' < n$, either because vertices associated with negative dual values are excluded or because branching in the BB phase has already removed part of the graph. As shown in Fig. 3.7, the QPU identifies a high percentage of improving columns, particularly for larger subproblems ($n' > 15$), where nearly 80% of the sampled bitstrings correspond to valid pricing solutions. This confirms that the quantum routine is particularly effective at exploring large unit-disk subproblems. For smaller instances ($n' \leq 5$), the proportion decreases below 40%, which is consistent with the reduced number of still-available improving columns in later iterations.

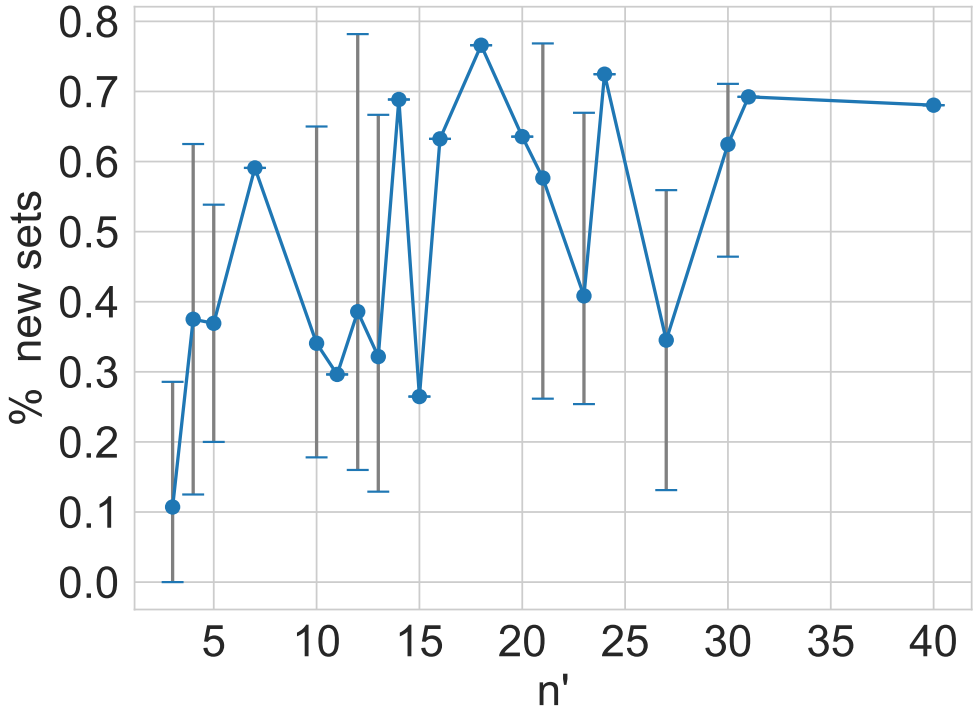


Figure 3.7: Proportion of improving columns generated by the Orion Alpha QPU as a function of PSP size n'.

A second relevant aspect concerns the branching phase, where maximal independent sets are particularly useful. Figure 3.8 shows the percentage of sampled solutions that correspond to maximal independent sets, again grouped by the subproblem size n'. As expected, the proportion decreases as n' increases, since maximality becomes harder to achieve in larger graphs. Nevertheless, the percentage remains substantial across all subproblem sizes, indicating that the QPU produces samples that are not only useful for pricing, but also informative for branching decisions.

A complete example of the QCBP workflow on a 10-vertex instance is reported in Figs. 3.9–3.10. The HCG procedure required four iterations, each aimed at identifying new independent sets with negative reduced cost $r_z < 0$ to be added to the subsequent RMPs.

In the first iteration (Fig. 3.9a), the pricing problem is defined over the full graph $\mathcal{G}$, with all vertex weights initialised to unity. The QPU executes the QAA sampler 200 times, producing bitstrings z of length $n' = n$. For each sample, the reduced cost is computed as $r_z = 1 - \sum_{i \in \mathcal{V}} z_i w_i$, where w denotes the vector of dual values from the current RMP solution. The majority of samples yield $r_z < 0$ and are therefore admitted as new columns; notably, the 16 most frequent bitstrings all correspond to

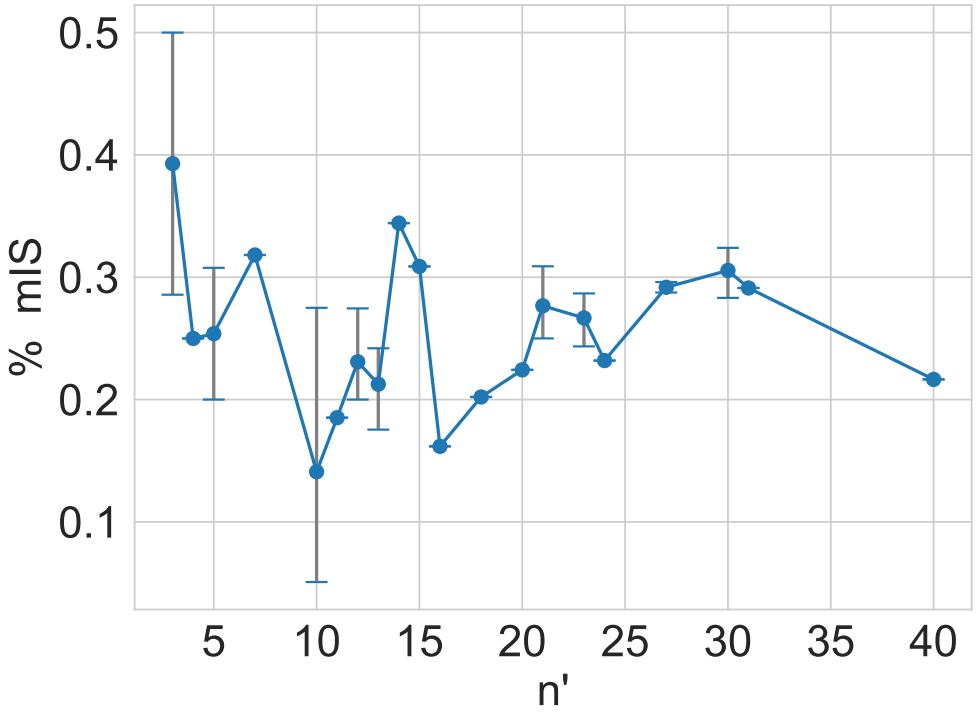
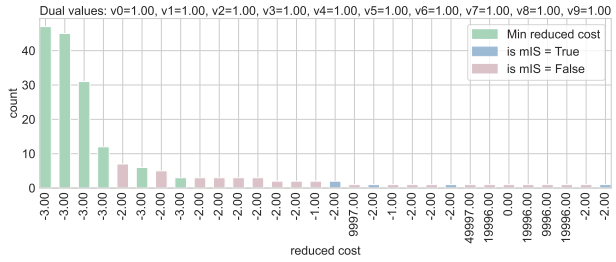


Figure 3.8: Percentage of maximal independent sets found through PSP solutions via QPU, grouped by PSP size n'.

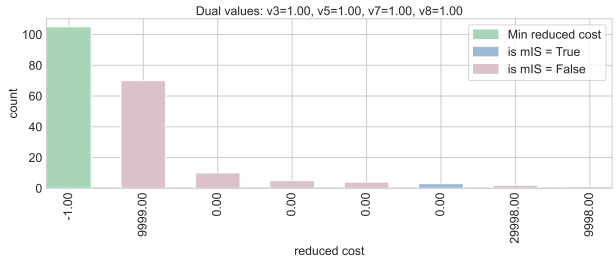
independent sets with negative reduced cost. Among these, the four lowest-cost solutions — $\{1,4,6,9\}$, $\{0,4,6,9\}$, $\{1,2,4,9\}$, and $\{1,2,4,8\}$ — are also maximal independent sets, highlighted in green in Fig. 3.9a. In total, 24 new columns are extracted from the 30 most relevant sampled bitstrings. As an illustration of the independence constraint enforcement, the 17^{th} most frequent bitstring $\{1,6,7,9\}$ is invalid, since vertices 7 and 9 share an edge (cf. Fig. 3.10); such violations are penalised with a coefficient of 10^4 in the objective.

In the second iteration (Fig. 3.9b), the pricing subproblem is restricted to a subgraph of $\mathcal{G}$, as several vertices have acquired negative dual values from the updated RMP solution and are consequently excluded. On this reduced instance, the QPU identifies the maximal independent set $\{3,7\}$ as the most frequently sampled valid solution, appearing in over 100 out of 200 draws; all remaining samples correspond to infeasible configurations and are discarded.

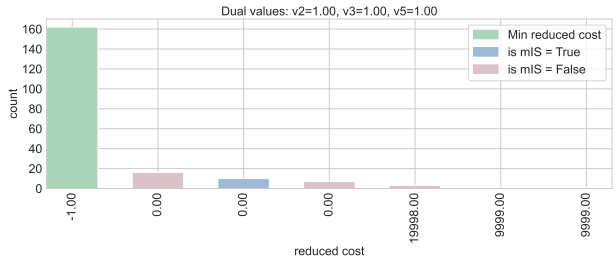
The third iteration (Fig. 3.9c) yields one further valid column, the independent set $\{2,5\}$, again on a reduced subgraph. By the fourth iteration (Fig. 3.9d), no improving columns remain: the minimum reduced cost among all sampled bitstrings is 0, and a confirmatory call to the exact ILP solver certifies termination of the pricing phase.



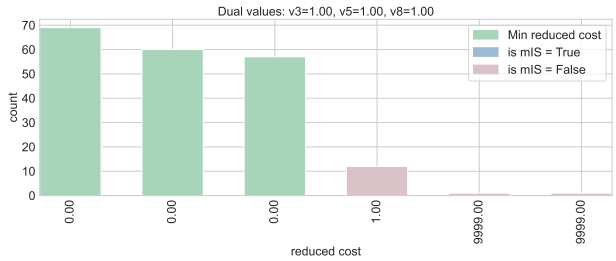
(a) Iteration 1: bitstring sampling on Orion Alpha QPU.



(b) Iteration 2: bitstring sampling on Orion Alpha QPU.



(c) Iteration 3: bitstring sampling on Orion Alpha QPU.



(d) Iteration 4: bitstring sampling on Orion Alpha QPU.

Figure 3.9: Sampled bitstring frequencies and associated reduced costs across four HCG iterations on the Orion Alpha QPU.

Once column generation is complete, the primal heuristic assembles a feasible coloring from the pool of columns discovered across all four iterations. For this instance, the optimal solution uses 3 colors: color class $1 = \{3,4,9\}$, color class $2 = \{0,5,6\}$, and color class $3 = \{1,2,7,8\}$, corresponding respectively to the 22^{nd}, 29^{th}, and 8^{th} most frequent solutions from the first iteration. The resulting 3-coloring, reported in Fig. 3.10, matches the size of the largest clique in $\mathcal{G}$, thereby certifying optimality by the clique lower bound.

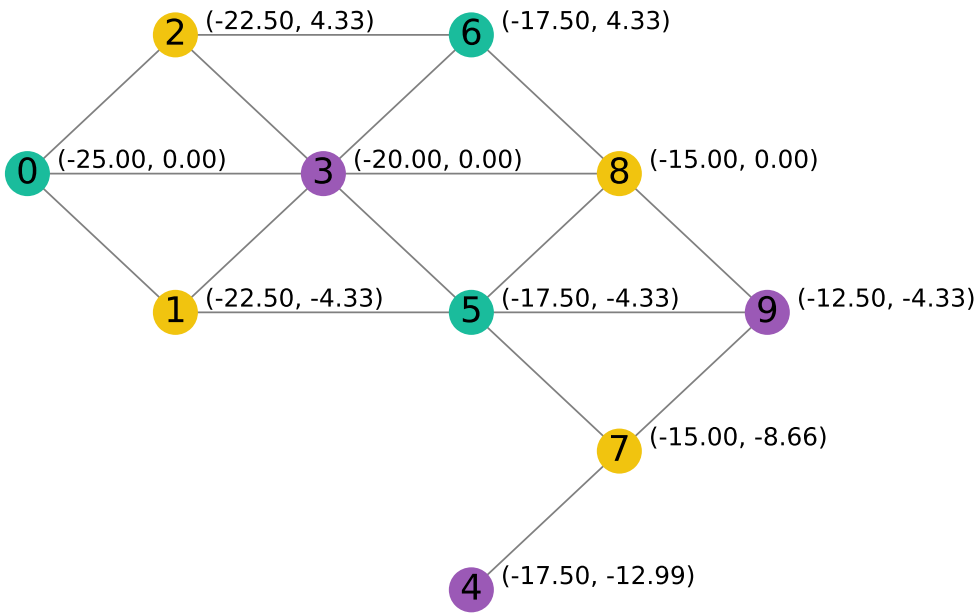


Figure 3.10: Final coloring of a 10-vertex VC instance solved with QCBP on the Orion Alpha QPU.

Overall, the experimental results demonstrate that the proposed QCBP framework effectively bridges classical optimization and neutral-atom quantum computation. In the emulation setting, the method consistently achieves near-optimal or optimal solutions while significantly reducing quantum resource requirements compared to previous approaches. These results are further supported by experiments on real hardware, where the QPU successfully identifies meaningful independent sets and integrates seamlessly within the hybrid workflow, despite the presence of noise and embedding imperfections.

Taken together, these findings confirm that combining structured decomposition techniques with quantum-assisted subroutines provides a practical and scalable pathway for tackling combinatorial optimization problems on near-term devices.

3.3 Application II: Space–Terrestrial Network Design

The hybrid optimization framework introduced in this chapter is now applied to a second use case: the design and optimization of STINs (Space-Terrestrial Integrated Networks). Although this application differs from vertex coloring in its problem structure and decomposition strategy, it relies on the same quantum computational primitive, namely the solution of independent set problems on neutral-atom QPUs. This continuity enables the direct reuse of the quantum routines developed in the previous sections

within a different application domain, providing a broader assessment of their generality and applicability.

Modern communication systems are increasingly required to deliver reliable connectivity across heterogeneous and geographically diverse environments. While terrestrial infrastructures offer high-performance communication in densely populated areas, their deployment remains limited in rural or remote regions due to economic and physical constraints [127, 128]. Satellite systems provide complementary wide-area coverage that is largely independent of ground conditions [129], and their integration with terrestrial components has given rise to the concept of STINs, where heterogeneous network elements cooperate to deliver continuous and scalable connectivity [130].

The design of such networks involves complex combinatorial decision problems, including resource allocation, link selection, and coordination across heterogeneous components. These problems are typically large-scale, rendering standard optimization techniques computationally intractable as problem size increases. It is precisely in this setting that hybrid quantum-classical methods offer a promising avenue, by offloading hard combinatorial subproblems to quantum hardware while retaining classical control over the overall deployment pipeline.

Within this context, the QRISTIN project (Quantum Routines In Space-Terrestrial Integrated Networks) investigates the application of such hybrid methods to STIN optimization challenges. The contributions of this thesis align directly with this objective: the embedding techniques and quantum solvers for MIS and MWIS problems introduced in Chapter 2 and the preceding sections are reused here as computational building blocks within a distinct optimization pipeline. However, whereas the vertex coloring application relied on the Orion Alpha QPU, the experiments in this section are conducted on the Aquila QPU by QuEra Computing [29], a neutral-atom device with a different native interaction model and connectivity constraints. This hardware difference necessitates targeted adaptations to the quantum routines, which are described in detail in the following sections. Crucially, the subproblems arising in STIN design admit independent set formulations, preserving structural compatibility with the quantum routines adopted throughout this work.

The following sections introduce the specific optimization problems considered in this application and describe how they are embedded within the hybrid methodology, with particular attention to the adaptations required to deploy the quantum routines developed in this thesis in this new operational context.

3.3.1 Combinatorial Structure of Satellite–Ground Integration

The optimization problems arising in STINs differ in formulation from the VC problem considered previously, yet they share a common combinatorial structure. In particular, several key tasks can be represented through graph models, where vertices correspond to network elements (e.g., satellites, gateways, or communication paths) and edges encode compatibility, resource-sharing, or interference constraints.

This abstraction is particularly relevant in the context of this thesis, as it enables the reuse of the same quantum computational primitive introduced in the previous sections, namely the solution of MIS/MWIS problems on neutral-atom QPUs. While the overall optimization framework differs from the VC case, the core quantum routine remains unchanged, highlighting its versatility across different application domains.

Within the QRISTIN project, these considerations are translated into a set of optimization problems that capture the main design and operational challenges of STINs. In this work, we focus on three representative problems, each associated with a different layer of the system and characterized by distinct decision variables and constraints. Despite these differences, all of them can be reformulated, either directly or through decomposition, in terms of independent set structures.

The first problem concerns the selection of satellites to extend coverage in areas not served by terrestrial infrastructure. The second addresses the assignment of satellites to ground gateways, ensuring balanced utilization of network resources. The third focuses on spectrum allocation, where interference between communication links must be avoided.

Together, these problems provide a coherent testbed to evaluate the applicability of hybrid quantum-classical approaches beyond the VC setting. In particular, they allow us to investigate how the same neutral-atom quantum routines can be embedded within different optimization pipelines, while preserving efficiency and scalability.

In the following, we describe each problem in detail and discuss how it can be mapped to combinatorial structures compatible with the quantum hardware.

Satellite Selection Problem

The SSP (Satellite Selection Problem) focuses on extending connectivity in regions not covered by terrestrial infrastructure. Given a service area $\mathcal{R}$, let $\mathcal{R}' \subset \mathcal{R}$ denote the uncovered portion. A set of candidate satellites $S = \{s_1, \dots, s_n\}$ is considered, where each satellite s_i is associated with a weight w_i representing the portion of $\mathcal{R}'$ it can cover over a given time window.

In this work, we focus on LEO (Low Earth Orbit) satellite constellations [131], motivated by their low latency and by the availability of realistic orbital datasets that enable accurate modeling. This choice is consistent with current and planned large-scale deployments, where LEO systems play a central role in extending connectivity.

Two satellites are considered conflictual if their coverage overlap exceeds a predefined threshold (e.g., $\epsilon = 0.90$), leading to a conflict graph $\mathcal{G}(S, E)$, where $(s_i, s_j) \in \mathcal{E}$ denotes excessive overlap. The objective is then to select a subset of satellites that maximizes coverage while avoiding redundancy.

This results in a MWIS problem, defined as:

$$\text{maximize} \quad \sum_{i \in S} w_i x_i, \tag{3.34}$$

$$\text{subject to} \quad \begin{cases} x_i + x_j \leq 1 & \forall (s_i, s_j) \in E, \\ x_i \in \{0,1\} & \forall s_i \in S. \end{cases} \tag{3.35}$$

This formulation directly matches the quantum computational primitive employed throughout this thesis. In particular, the correspondence between independent sets and feasible configurations enforced by the Rydberg blockade allows the problem to be naturally mapped onto neutral-atom QPUs, where candidate solutions can be efficiently sampled using the quantum routines introduced in the previous sections.

Variants of the SSP can also be considered depending on the application scenario. For instance, satellites may complement terrestrial base stations to improve coverage, or temporarily replace them in energy-aware configurations where parts of the ground infrastructure are deactivated. These extensions preserve the underlying combinatorial structure and remain compatible with the same MWIS-based quantum approach.

Gateway Selection Problem

The GSP (Gateway Selection Problem) addresses the assignment of satellites to terrestrial gateways. Each satellite must be connected to exactly one gateway, while ensuring a balanced load across gateways.

Let $G = \{g_1, \dots, g_m\}$ be the set of gateways and $L \subseteq S \times G$ the set of feasible links. Binary variables y_{ij} indicate whether satellite s_i is assigned to gateway g_j. The objective is to minimize the maximum load M:

$$\text{minimize} \quad M, \tag{3.36}$$

$$\text{subject to} \quad \sum_{j:(s_i,g_j)\in L} y_{ij} = 1 \qquad \forall s_i \in S, \tag{3.37}$$

$$\sum_{i:(s_i,g_j)\in L} y_{ij} \leq M \qquad \forall g_j \in G, \tag{3.38}$$

$$y_{ij} \in \{0,1\}, \quad M \in \{0, \dots, n\}. \tag{3.39}$$

Constraint (3.37) enforces that each satellite is assigned to exactly one gateway among the feasible links. Constraint (3.38) ensures that the number of satellites connected to any gateway does not exceed the maximum load M, which is minimized in the objective to promote balanced utilization of the infrastructure. Finally, constraint (3.39) defines the binary nature of the assignment variables and the domain of M.

Although the GSP does not directly reduce to a MIS or MWIS problem, it still exhibits a combinatorial structure driven by compatibility constraints between assignments. In particular, conflicts may arise due to limited gateway capacity or link feasibility, leading to subproblems where mutually incompatible assignments must be excluded. These substructures can be reformulated as independent set problems, enabling the integration of MIS-based quantum routines within a broader hybrid optimization workflow.

Spectrum Assignment Problem

The SAP (Spectrum Assignment Problem) concerns the allocation of frequency resources once the network topology has been established. Let P be the set of communication paths, and let $E \subseteq P \times P$ denote the set of pairs of paths that share physical resources and therefore cannot use the same frequency. The set of available frequencies is denoted by K.

The problem can then be formulated as a VC problem, where frequencies correspond to colors and interfering paths must be assigned different frequencies. The corresponding ILP formulation is:

$$\text{minimize} \quad \sum_{p \in P} \sum_{q \in K} c_{pq} x_{pq}, \tag{3.40}$$

$$\text{subject to} \quad \sum_{q \in K} x_{pq} = 1 \qquad \forall p \in P, \tag{3.41}$$

$$x_{pq} + x_{rq} \leq 1 \qquad \forall (p,r) \in E,\ \forall q \in K, \tag{3.42}$$

$$x_{pq} \in \{0,1\} \qquad \forall p \in P,\ \forall q \in K. \tag{3.43}$$

Constraint (3.41) ensures that each communication path is assigned exactly one frequency. Constraint (3.42) enforces interference avoidance by preventing adjacent paths from sharing the same frequency. Finally, constraint (3.43) defines the binary nature of the assignment variables.

This formulation establishes a direct connection with the VC framework discussed earlier in the chapter. As a result, the hybrid quantum-classical approaches developed for VC, and in particular those based on MIS/MWIS decompositions, can be directly adapted to address the SAP. In this sense, the SAP provides a natural extension of the previous application to a more complex network design setting.

Quantum-Compatible Structure

Despite their differences, the three problems share a common feature: they can be expressed, either directly or through decomposition, in terms of independent set structures. This is particularly evident for the SSP, which directly maps to MWIS, and for the SAP, which reduces to vertex coloring. The GSP, although structurally different, can

also be addressed within a hybrid framework where combinatorial subproblems exhibit compatibility constraints. This common structure enables the reuse of the quantum routines developed in this thesis.

To execute the quantum routines on the neutral-atom platform Aquila [29], which was the target QPU for the QRISTIN project, the combinatorial problems introduced above must be embedded into the physical qubit register. As discussed in the embedding chapter, this requires mapping the input graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ to a geometric configuration of atoms, where each vertex $i \in \mathcal{V}$ is associated with a position $\vec{p}_i \in \mathbb{R}^2$. Edges are encoded through the Rydberg blockade mechanism: atoms placed within a certain distance interact strongly and cannot be simultaneously excited. This induces a unit-disk graph representation of $\mathcal{G}$, allowing MIS/MWIS problems to be naturally mapped onto the hardware. The embedding must satisfy the geometric constraints of the Aquila register, summarized in Table 3.5[2].

Table 3.5: Geometric constraints of the Aquila QPU register.

Constraint	Requirement
Minimum vertical row spacing (D_{row})	$\geq 2\,\mu$m
Minimum inter-atom distance (D_{min})	$\geq 4\,\mu$m
Register dimensions ($L_x \times L_y$)	$76\,\mu\text{m} \times 128\,\mu\text{m}$

3.3.2 Solving MWIS with Aquila

The MWIS problem, central to the SSP, is solved using the QAA [121]. As discussed in the previous sections, neutral-atom systems provide a natural physical implementation of independent set constraints through the Rydberg blockade mechanism. In this setting, each vertex $i \in \mathcal{V}$ is associated with a qubit, whose state encodes whether the corresponding vertex is included in the solution.

The dynamics of the system are governed by the following Hamiltonian:

$$\begin{aligned}\mathcal{H}(t) = \frac{\hbar\Omega(t)}{2} \sum_{i=1}^{n} \left(|g_i\rangle\langle r_i| + |r_i\rangle\langle g_i|\right) \\ - \hbar \sum_{i=1}^{n} \delta_i(t)\, \hat{n}_i + \sum_{i<j} V_{ij},\end{aligned} \tag{3.44}$$

[2]The geometric constraints reported here reflect the specifications of the Aquila device at the time of the experiments. Hardware updates have modified parameters such as the register size and minimum row spacing. These changes, however, do not affect the embedding methodology, which remains valid under updated device configurations.

where the interaction term is defined as

$$V_{ij} = C_6 \frac{\hat{n}_i \hat{n}_j}{|\vec{p}_i - \vec{p}_j|^6}, \tag{3.45}$$

and

$$\hat{n}_i = |r_i\rangle\langle r_i|. \tag{3.46}$$

In this formulation, $\Omega(t)$ denotes the time-dependent Rabi frequency, which drives transitions between the ground state $|g_i\rangle$ and the Rydberg state $|r_i\rangle$. The operator $\hat{n}_i$ measures whether atom i is in the excited state. The interaction term V_{ij} captures the van der Waals interaction between atoms, which decays with the sixth power of the distance $|\vec{p}_i - \vec{p}_j|$.

The key element enabling the encoding of the MWIS problem is the detuning term $\delta_i(t)$, which is site-dependent. In contrast to a purely global detuning, the Aquila device allows the use of both a global component $\delta(t)$ and a local contribution $\delta_i^{\text{loc}}(t)$, such that

$$\delta_i(t) = \delta(t) + \delta_i^{\text{loc}}(t). \tag{3.47}$$

This decomposition plays a central role in the encoding of vertex weights. Specifically, the local detuning δ_i^{loc} is used to incorporate the weight w_i of vertex i into the Hamiltonian. By appropriately rescaling the weights, vertices with higher weight are energetically favored, increasing the likelihood that the corresponding qubits are found in the excited state $|r_i\rangle$ in low-energy configurations.

The interaction term V_{ij} enforces the independence constraint: if two atoms are placed within the blockade radius r_b, the energy penalty associated with their simultaneous excitation becomes large, effectively preventing both vertices from being selected. The blockade radius is chosen according to the embedding distances as

$$r_b = \sqrt{d_{\text{adj}} \cdot d_{\overline{\text{adj}}}}, \tag{3.48}$$

ensuring a consistent mapping between the geometric layout and the target graph structure [132].

The QAA proceeds by initializing the system in an easily preparable state, typically with all atoms in the ground state $|g\rangle^{\otimes n}$, and then gradually varying the control parameters $\Omega(t)$ and $\delta(t)$ according to a predefined schedule, as illustrated in Fig. 3.11. On Aquila, these parameters are implemented as piecewise-linear pulse sequences.

The pulse is designed so that, at the initial time, the detuning is negative and $\Omega(t) = 0$, ensuring that the ground state is energetically favored. During the evolution, $\Omega(t)$ is increased to promote superposition and exploration of the configuration space, while $\delta(t)$ is swept toward positive values. In the final stage, $\Omega(t)$ is brought back to zero and $\delta(t)$ becomes positive, so that the Hamiltonian is dominated by the detuning and interaction terms. In this regime, low-energy configurations correspond to feasible high-weight independent sets.

In parallel, the local detuning channel $\delta_i^{\text{loc}}(t)$ biases the dynamics toward vertices with larger weights. The combined effect of the global driving term, local detuning, and interaction term ensures a balance between exploration (via $\Omega(t)$), optimization (via δ_i^{loc}), and feasibility (via V_{ij} through the blockade mechanism).

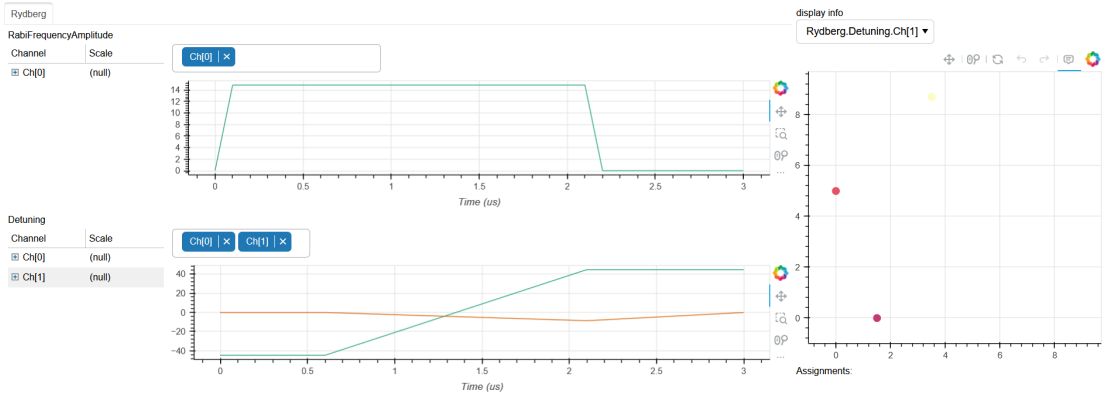


Figure 3.11: Adiabatic pulse schedule on Aquila for a MWIS instance on a 3-vertex graph. Channel *Ch[0]* broadcasts the global drive, comprising the Rabi frequency $\Omega(t)$ and the global detuning $\delta(t)$. Channel *Ch[1]* encodes the local detuning $\delta_i^{\text{loc}}(t)$, with site-specific amplitudes reflected by the color-coded qubit assignments.

At the end of the evolution, repeated measurements of the system yield bitstrings representing candidate solutions. Following the hardware convention, empty traps correspond to selected vertices (excited atoms), while occupied traps correspond to excluded vertices. Due to the probabilistic nature of the process and the presence of noise, some sampled configurations may violate independence constraints or be suboptimal.

To address this, a classical post-processing step is applied. Bitstrings corresponding to non-independent sets are discarded, and the remaining solutions are refined using a greedy augmentation procedure. In particular, additional vertices are iteratively added in descending order of weight whenever this does not violate independence. This hybrid quantum-classical workflow improves solution quality and mitigates the effects of imperfect embeddings and hardware noise.

For small problem instances ($|\mathcal{V}| \leq 15$), the full quantum evolution can be simulated classically. For larger instances, the algorithm is executed directly on the Aquila QPU, leveraging its capability to efficiently sample from the underlying quantum distribution.

3.3.3 Performance Evaluation

To evaluate the performance of the proposed quantum optimization framework based on QAA, a dataset of STIN instances was compiled, encompassing 165 remote regions defined as areas where terrestrial coverage remains limited ($\leq 60\%$ of the total territory). Terrestrial base stations and gateway locations were extracted from OpenStreetMap, while satellite orbital data were obtained from Celestrak, selecting the ONEWEB

constellation.

The dataset is generated through a simulation pipeline that combines orbital dynamics and geographic coverage. For each satellite, its trajectory is propagated over one full orbital period using the publicly available orbital elements. At each time step, the satellite footprint on Earth is computed, allowing us to determine which portions of the service area $\mathcal{R}$ are covered. By intersecting these footprints with the uncovered region $\mathcal{R}'$, we obtain time-dependent coverage information for each satellite.

From this temporal data, we derive static graph instances by aggregating coverage over the considered time window. In particular: *i)* each satellite is associated with a weight w_i, corresponding to the average fraction of $\mathcal{R}'$ it covers over the time horizon; *ii)* pairwise overlaps between satellites are evaluated over time, and an edge is introduced when the overlap exceeds a threshold (set to 90%) for the same region and time interval.

This procedure yields a conflict graph capturing both spatial and temporal redundancy between satellites, which forms the basis of the SSP. The same simulation outputs are then used to construct instances for the other STIN subproblems.

- **SSP instances:** graphs in which vertices denote satellites, vertex weights reflect the proportion of the uncovered area $\mathcal{R}'$ served within a given time window, and edges link pairs of satellites whose coverage regions overlap by more than 90%;
- **GSP instances:** bipartite graphs where vertices represent satellites and gateways, and edges denote feasible communication links;
- **SAP instances:** conflict graphs defined over communication paths, where edges represent shared physical resources.

For the SSP, which constitutes the core combinatorial component, three solution strategies were compared:

1. The quantum optimization approach based on QAA, with a fixed budget of 300 quantum measurements;
2. An exact ILP solver (GLPK [82]) implemented via `Pyomo` [133], with a time limit of 60 seconds, corresponding to the acquisition time of 300 quantum samples at 5 Hz;
3. A greedy heuristic, which iteratively selects the highest-weight satellite while preserving feasibility. This method represents a classical baseline and can be interpreted as the *no-quantum limit* [134].

Figure 3.12 reports the relative performance of QAA with respect to both classical baselines across all evaluated instances. The quantum approach consistently outperforms the greedy heuristic, with an average gain of 9.2%, while closely matching the

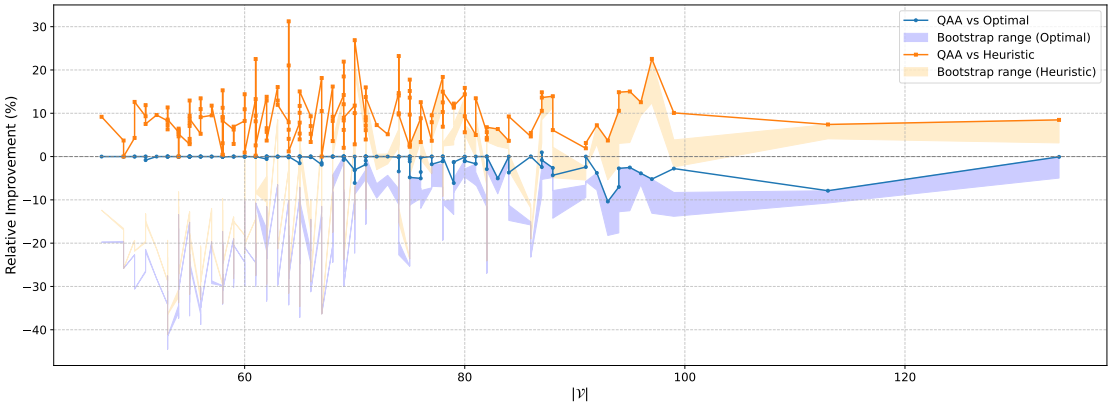


Figure 3.12: Per-instance performance gap of QAA relative to the greedy heuristic (orange) and the exact solver (blue), with bootstrap confidence regions obtained by subsampling 100 out of 300 measurements.

exact solver, deviating by only -0.7% on average. Importantly, the advantage over the heuristic becomes more pronounced as the graph size $|\mathcal{V}|$ increases, suggesting that quantum sampling is particularly effective in larger combinatorial spaces where greedy strategies are more likely to settle in suboptimal configurations.

A notable exception is observed for a 96-vertex instance, where QAA outperforms the exact solver within the same time budget, achieving an objective value of 16.50 compared to 16.22 for *GLPK*. This outcome is attributable to the time-limited execution of the classical solver, which fails to reach optimality within the allotted budget, and highlights the potential of quantum methods to deliver competitive solutions even under non-ideal embedding conditions and at scales where classical solvers begin to struggle.

The bootstrap regions reported in Fig. 3.12 provide further insight into the role of measurement statistics. These are computed by subsampling 100 bitstrings from the original pool of 300 and repeating the evaluation across multiple draws. The resulting performance degradation confirms that the number of measurements is a critical factor: while 300 samples prove sufficient for smaller instances, the widening bootstrap bands for larger graphs indicate that a higher shot count would be necessary to adequately explore the solution space and maintain solution quality at scale.

The impact of classical post-processing was assessed via the normalized Hamming distance between raw and refined bitstrings. On average, the refinement step increases the solution size by approximately 10%, with a standard deviation of 0.05, reflecting the variability in the degree of correction required across instances. In some cases, the raw quantum output already constitutes a valid independent set and no correction is needed; in the most challenging instances, however, up to 26% of vertices require addition. This range underscores the importance of post-processing as an integral component of the quantum pipeline, ensuring that noisy or incomplete quantum outputs are consistently

converted into high-quality feasible solutions.

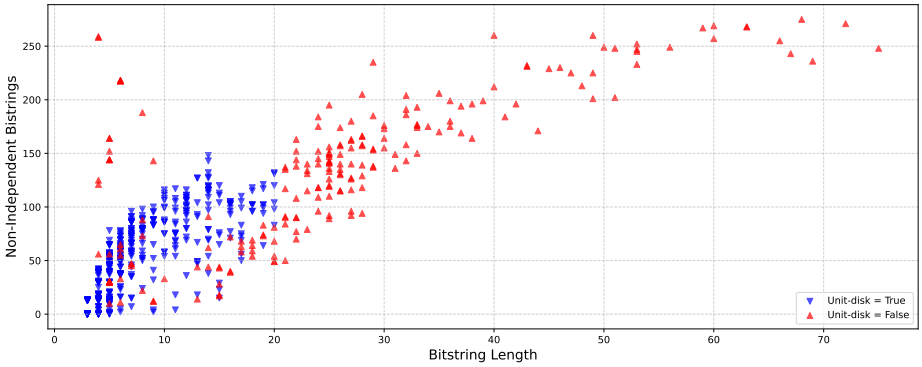
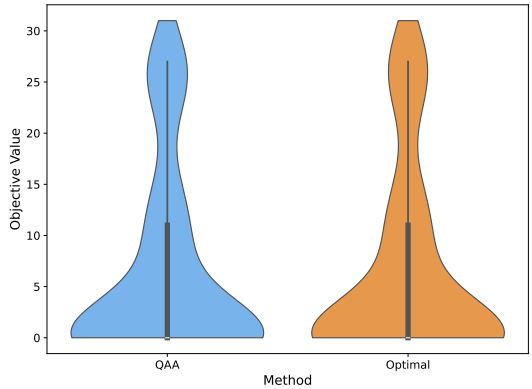
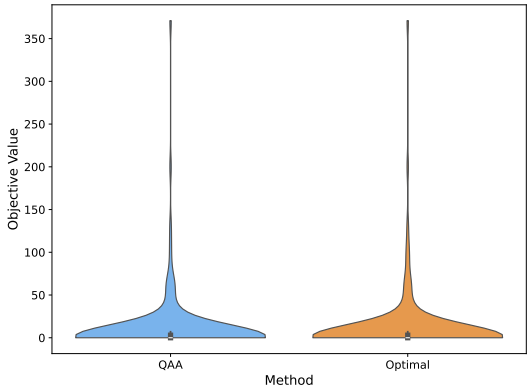


Figure 3.13: Number of non-independent bitstrings as a function of graph size $|\mathcal{V}|$, illustrating the degradation in quantum encoding fidelity for larger and harder-to-embed instances.

A complementary perspective on solution quality is provided by the fraction of *non-independent bitstrings*, i.e., sampled configurations that violate independence constraints and are therefore infeasible. As shown in Fig. 3.13, their occurrence grows with graph size and is strongly influenced by embedding quality. Larger graphs are inherently more difficult to represent as UD instances, introducing structural approximations in the quantum encoding that increase the probability of constraint violations. This trend motivates the development of more accurate embedding strategies for large-scale instances, as discussed in Chapter 2.



(a) Distribution of GSP objective values obtained when supplying satellite sets derived from either QAA or the exact solver to the PSP.



(b) Distribution of SAP objective values for network topologies derived from the full optimization pipeline, comparing quantum and classical SSP solutions.

Figure 3.14: Downstream impact of SSP solution quality on GSP and SAP objective distributions, comparing pipelines seeded with QAA versus exact solver outputs.

Although QAA and the exact solver yield differing SSP solutions in 94 out of 165 instances (51 affecting the objective value and 43 corresponding to alternative optimal configurations of equal cost) their downstream impact on the full optimization pipeline remains limited. This distinction is important: it indicates that a substantial portion of the observed differences do not reflect suboptimality of the quantum method, but rather the existence of multiple equivalent solutions, a common phenomenon in combinatorial problems with symmetric structure.

Figure 3.14 quantifies this downstream robustness by comparing the objective distributions for the GSP and SAP stages across both pipelines. For the GSP, only 9 instances exhibit different objective values, with a Jensen–Shannon divergence of 0.00013, confirming near-identical distributional behavior. In the cases where differences do arise, the quantum-derived solutions tend to produce slightly more balanced gateway assignments, which may be advantageous in practice for load distribution. For the SAP, discrepancies are observed in only 6 instances, though with larger individual deviations: in the worst case, the fully-classical pipeline requires up to 35 additional frequency bands. Despite this, the overall distributions remain highly consistent across both approaches.

Taken together, these results establish that the QAA-based approach provides high-quality SSP solutions while preserving robustness when integrated into the full STIN optimization pipeline. The negligible propagation of SSP differences to downstream stages confirms that the quantum method constitutes an effective and scalable alternative to classical optimization for the graph-based subproblems arising in network design, even in the presence of hardware noise and imperfect embeddings.

3.4 Discussion and Perspectives on Hybrid Quantum-Classical Optimization

The two applications presented in this chapter, vertex coloring and space–terrestrial network optimization, highlight a common methodological pattern: the integration of quantum routines within structured classical optimization frameworks. Despite their different problem formulations, both settings rely on the same underlying quantum computational primitive, namely the solution of MIS/MWIS problems on neutral-atom QPUs. This commonality provides useful insights into when and how hybrid quantum-classical approaches can be effective.

From an algorithmic perspective, a key design principle emerging from this work is that quantum routines are most effective when embedded as *subproblem solvers* within a larger classical structure. In the case of vertex coloring, this is achieved through the QCBP framework, where the quantum routine acts as a pricing oracle within a CG procedure. In the STIN application, the same quantum kernel is used to solve the SSP, which in turn influences downstream optimization problems such as GSP and SAP. In both cases, the quantum component does not replace classical optimization, but rather complements it by efficiently exploring large combinatorial neighborhoods.

From a computational standpoint, the cost of quantum execution is primarily determined by the number of measurements (shots), typically in the range 10^2–10^4, and by the hardware sampling rate. Current devices operate at approximately 1 Hz for the Orion Alpha QPU and up to 5 Hz for Aquila. As a result, total runtime is dominated by the time required to collect sufficient samples. While present-day devices do not yet provide a computational advantage over classical solvers in terms of wall-clock time, especially for small problem instances, this limitation is largely due to hardware constraints. Future generations of neutral-atom devices are expected to reach sampling rates on the order of 100 Hz, which would directly reduce quantum execution times.

From a hardware perspective, also coherence time remains a key limiting factor. Longer coherence would allow for slower, more adiabatic evolutions, potentially improving the quality of the sampled solutions. At the same time, the duration of a quantum evolution and measurement cycle is largely independent of the number of qubits for the problem sizes considered. This contrasts with classical exact methods for combinatorial optimization problems such as VC or MWIS, which exhibit exponential scaling in the worst case. While this does not yet translate into a practical advantage, it suggests a favorable asymptotic behavior as hardware improves.

Another important aspect concerns qubit efficiency and embedding overhead. The approaches developed in this chapter rely on direct graph embeddings that avoid the need for auxiliary variables or large overhead in the number of qubits. This is particularly relevant for neutral-atom architectures, where geometric constraints play a central role. Current neutral-atom platforms, such as Orion Alpha and Aquila, support up to 100 and 256 qubits, respectively, which enables the treatment of moderately sized instances. Looking forward, the scalability of these methods is closely tied to hardware developments.

Overall, the results presented in this chapter should be interpreted as a proof of principle for the effective integration of quantum routines into classical optimization pipelines. Rather than demonstrating immediate computational advantage, the focus has been on identifying algorithmic structures where quantum sampling can provide value, and on designing hybrid methods that are robust to current hardware limitations. Extending these approaches to larger instances, improving embedding strategies, and performing systematic hardware-level benchmarking remain important directions for future work.

Chapter 4

Hybrid Quantum Machine Learning on Neutral Atom Platforms

The previous chapter focused on combinatorial optimization problems and demonstrated how neutral-atom quantum processors can be effectively integrated within classical algorithms through a common computational primitive. The underlying theme was the exploitation of analog quantum dynamics to explore large discrete solution spaces, leveraging the native interaction structure of neutral-atom hardware as a computational resource. Beyond optimization, however, these same physical properties offer interesting opportunities for data-driven tasks, where the objective shifts from identifying a single optimal configuration to extracting informative representations from structured data.

This transition from optimization to learning is not merely a change of application domain, but reflects a broader perspective on the role of quantum hardware in computational pipelines. In both settings, the quantum device is used to perform transformations that are difficult to replicate efficiently on classical hardware, while classical algorithms handle the higher-level inference or search logic. In this chapter, we investigate the use of neutral-atom devices for QML (Quantum Machine Learning), a research direction that has attracted significant attention for its potential to exploit quantum resources to enhance learning tasks [12, 135].

The effectiveness of QML approaches, however, depends critically on the interplay between algorithms and hardware. Gate-based quantum computers have dominated the QML literature, yet neutral-atom platforms offer a qualitatively different paradigm: rather than executing discrete gate sequences, they rely on continuous analog quantum evolution governed by a programmable Hamiltonian [28]. This distinction has important implications for the design of learning algorithms, as the expressivity and inductive bias of the quantum model are directly shaped by the physical dynamics of the device rather than by an explicit circuit architecture.

This feature makes neutral-atom platforms particularly well-suited for hybrid learning strategies, where the quantum system acts as a transformation layer and classical

methods handle training and inference. Within this framework, we explore two complementary approaches. The first relies on quantum kernel methods, where the quantum evolution defines an implicit feature map and similarities between data points are computed via quantum state overlaps [136]. The second is based on quantum reservoir computing, where the quantum system serves as a dynamical reservoir that processes input data through its time evolution, drawing on the rich expressive power of many-body quantum dynamics [137, 138].

Both approaches follow a hybrid paradigm that is well aligned with current NISQ (Noisy Intermediate-Scale Quantum) capabilities. The quantum device is employed to generate expressive feature representations that would be costly to produce classically, while classical algorithms exploit these features for supervised learning tasks. Together, they illustrate how the same neutral-atom hardware used for optimization in the previous chapter can be repurposed as a versatile computational substrate for machine learning, reinforcing the broader thesis that analog quantum platforms offer a unified and practically relevant computational primitive across multiple problem domains.

4.1 Neutral Atom Dynamics for Hybrid Learning

In hybrid quantum machine learning, a central challenge is the construction of feature mappings that transform input data into representations suitable for learning tasks. On neutral-atom platforms, such mappings can be naturally realized through the time evolution of the quantum system.

As discussed in Chapter 1, neutral-atom devices operate by evolving a quantum state under a controllable Hamiltonian, whose parameters—such as the Rabi frequency and the detuning—can be modulated over time. This analog evolution, combined with the interaction mechanisms induced by the Rydberg blockade effect [22], provides a flexible and physically grounded way to encode and process information.

From a machine learning perspective, classical data can be embedded into the quantum system either through the control parameters or through the spatial arrangement of the atoms in the register. The subsequent quantum evolution then acts as a nonlinear transformation, mapping the input into a high-dimensional Hilbert space. Measurements performed at the end of the evolution produce classical features, which can be further processed using standard learning models.

This viewpoint establishes a direct connection between neutral-atom dynamics and two important paradigms in QML. In QRC (Quantum Reservoir Computing), the quantum system is interpreted as a dynamical reservoir that transforms input signals into expressive feature representations. In QEK (Quantum Evolution Kernel) methods, instead, the evolution defines an implicit feature map, and learning is carried out through kernel evaluations [136]. In both cases, the quantum device does not directly perform the learning task, but rather provides a transformation of the data that is potentially difficult to reproduce classically.

The development of QML methods has followed the broader evolution of quantum computing. Early proposals often relied on advanced capabilities such as QRAM or fault-tolerant computation, which remain beyond the reach of current hardware [139]. In the NISQ era, attention has shifted toward algorithms compatible with limited resources, characterized by a reduced number of qubits and short coherence times. This has led to the design of shallow and resource-efficient approaches, many of which are based on variational principles. While several of these methods have been experimentally explored on simplified or synthetic datasets [140, 141], the encoding of realistic data into quantum systems remains a significant challenge. Moreover, the reliance on classical optimization in variational approaches often limits their scalability and makes it difficult to assess potential quantum advantages.

In this context, analog quantum simulators offer an alternative direction. By leveraging continuous-time dynamics instead of parameterized circuits, they provide a natural framework for implementing data-driven quantum evolutions without the overhead of variational optimization. Neutral-atom platforms, in particular, combine programmable interactions with flexible geometries, making them suitable candidates for graph-based learning tasks. Despite these advantages, the exploration of QML techniques on such platforms is still at an early stage, with only a limited number of works addressing their use for learning applications [142, 143].

In the following sections, these properties are exploited to design hybrid learning schemes where quantum evolution acts as a feature generator, while classical models perform the final inference step.

4.1.1 Rydberg Interactions as High-Dimensional Feature Maps

Graph-structured data are widely used in many application domains, including telecommunications [144], biology [145, 146], and social sciences [147]. Their ability to represent interactions between entities makes them particularly expressive, but also computationally demanding. In classical ML (Machine Learning), handling such data often requires complex feature extraction procedures or specialized kernels, whose cost can grow quickly with graph size.

Quantum computing provides an alternative approach by exploiting the exponentially large Hilbert space of quantum systems. This has motivated the development of QML methods tailored to structured data [148, 149, 150]. Among these, quantum kernel methods define implicit feature maps through quantum evolution, allowing similarities between data points to be evaluated via overlaps in the corresponding quantum states. Several realizations have been proposed for gate-based platforms, including quantum convolutional neural networks [151, 152, 153] and kernel-based classifiers [154].

Neutral-atom platforms offer a different perspective, as they rely on continuous-time dynamics rather than discrete gate operations. In this setting, QEKs can be naturally defined by exploiting the physical evolution of the system. The general idea, introduced in [143], is to encode graph-structured inputs into a quantum register and let the

system evolve under a fixed Hamiltonian. The resulting state captures correlations induced by the interactions, effectively mapping the input graph into a high-dimensional feature space. This approach is particularly well suited for graph data, since the interaction pattern of the quantum system can reflect the structure of the input. Extensions of this framework have been proposed to incorporate symmetry properties [155] and to study the geometric aspects of the resulting feature spaces [156, 157]. Experimental validation, however, is still limited, especially on neutral-atom platforms, where hardware constraints and embedding challenges must be taken into account [142].

A key step in this context is the mapping of abstract graph structures onto the physical qubit register. To address this, the embedding method introduced in Chapter 2 is exploited, enabling the construction of atom layouts that reflect the desired connectivity.

In Section 4.2, the QEK framework is employed to define quantum feature maps based on Rydberg dynamics, and to assess their effectiveness on a graph classification task involving protein structures.

4.1.2 Analog Quantum Reservoir Computing on Neutral-Atom Platforms

RC (Reservoir Computing) is a computational framework designed to exploit the dynamic properties of RNNs (Recurrent Neural Networks) for efficient time series processing [158, 159] and pattern recognition [160, 161]. In contrast to conventional RNNs, RC approaches rely on a high-dimensional, nonlinear *fixed* reservoir for the recurrent dynamics, which considerably lowers the computational burden of training. Only the readout layer requires optimization, typically a straightforward linear classifier with a limited number of parameters, rendering RC models compact, computationally efficient, and easy to train. The reservoir operates as a dynamic memory mechanism, projecting input signals into rich internal representations that preserve temporal dependencies and capture intricate patterns. This architecture has demonstrated robustness and adaptability across a wide range of domains, from signal processing and chaotic time series forecasting to neuroscience-inspired computing and hardware implementations in photonic and memristive systems [162].

The quantum counterpart of this framework, QRC (Quantum Reservoir Computing), was first introduced by Fujii and Nakajima as a resource-efficient paradigm for quantum machine learning, leveraging the intrinsic evolution of quantum states as a high-dimensional computational reservoir while keeping training requirements minimal [163]. By replacing the classical reservoir with a quantum system, QRC leverages the exponentially large Hilbert space and the complex many-body dynamics of quantum hardware to generate expressive feature representations that would be costly to produce classically.

Numerous architectures tailored to digital quantum hardware have since emerged. Yasuda et al. introduced a QRC scheme leveraging repeated quantum measurements

on superconducting qubits [164], whereas Kobayashi et al. developed a feedback-based QRC architecture aimed at counteracting the detrimental impact of projective measurements [165]. The potential of analog quantum simulators as reservoirs has received comparatively less attention. Martínez-Peña et al. investigated the influence of dynamical phases, such as thermalization and many-body localization, on the information processing capacity of analog QRC [166], while Bravo et al. realized a quantum analog of recurrent neural networks using Rydberg atom arrays, showcasing the ability to perform cognitive tasks including multitasking, decision-making, and long-term memory retention [167].

Building on this body of work, Kornjača et al. [168] demonstrated the large-scale applicability of QRC on neutral-atom platforms, with implementations scaling up to 108 qubits and benchmarked on several established datasets. Their study introduced three input encoding strategies, i.e. global pulse, local detuning, and position encoding, of which the latter two consistently achieved superior performance and are therefore adopted in the present work. Despite these advances, fundamental questions remain open: it is still unclear which data types and ML tasks are best suited to QRC methods, and real-world applicability has yet to be systematically assessed. The present work aims to contribute to a more principled understanding of these aspects.

In the QRC framework adopted here, input features are encoded into the quantum system by mapping them to the Hamiltonian parameters, which govern the time evolution of the reservoir state, and register design. Output features are then constructed by measuring a set of observable operators at a predetermined sequence of timesteps, capturing the temporal dependence induced by the input. The resulting measurement record constitutes the feature vector that is subsequently fed into a classical classifier for the final prediction.

In Section 4.3, this methodology is applied to a financial use case: credit card default prediction.

4.2 Application I: Protein Graph Classification

This section investigates the use of QML techniques based on neutral-atom platforms for graph classification tasks. The approach relies on QEKs, where the time evolution induced by the neutral-atom Hamiltonian is used to extract features from graph-structured data. These features are employed to compute a kernel, following the formulation in [143], which is then used within a classical SVM (Support Vector Machine) for classification. The performance of this hybrid quantum-classical pipeline is compared with that of a standard classical graph kernel.

As a concrete application, the classification of protein structures is considered using the PROTEINS dataset [86, 85]. Proteins are complex biological macromolecules composed of sequences of amino acids that fold into three-dimensional structures. These structures largely determine the function of the protein, for instance whether it acts as

an enzyme catalyzing biochemical reactions or as a structural or regulatory component. While amino acid sequences provide relevant information, it is often the spatial arrangement and interaction patterns between residues that are most indicative of function. In this setting, proteins can be naturally represented as graphs, where nodes correspond to amino acids and edges encode spatial proximity or interaction relationships. This representation captures both local interactions and global structural organization. The classification task consists in distinguishing between two classes: *enzymes* and *non-enzymes*. This is a challenging problem, as proteins from different classes may share similar sequences while differing in subtle structural features. Capturing such differences requires feature representations capable of modeling complex dependencies between amino acids.

The use of QEKs is motivated by the possibility of encoding these structural properties into the dynamics of a quantum system. By mapping each graph to a configuration of atoms and allowing the system to evolve under Rydberg interactions, one obtains feature representations that incorporate both connectivity and geometry. Similarity between proteins is then evaluated through kernel computations.

Experiments are conducted using the Aquila neutral-atom simulator developed by QuEra Computing [29], which supports up to 256 qubits and is accessible through Amazon Web Services Braket. Due to the computational cost associated with simulating quantum dynamics, part of the analysis relies on classical emulation on reduced instances, which also allows the study of pulse design and its impact on feature quality.

4.2.1 Protein Graph Representation and Dataset Preparation

Proteins are macromolecules composed of chains of amino acids arranged in three-dimensional space, making them naturally amenable to graph-based representations. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed by associating each amino acid with a node $v \in \mathcal{V}$, and introducing an edge between two nodes if the corresponding amino acids lie within 6 Å of each other. This proximity-based connectivity captures the spatial interactions that are crucial for determining the structural and functional properties of the protein.

The PROTEINS dataset [169] is one of the most widely used benchmarks for graph-based machine learning, and is adopted here as the primary evaluation benchmark. It consists of 1113 graphs, each associated with a binary label indicating whether the protein is an *enzyme* or a *non-enzyme*, a distinction of particular relevance in biology, since enzymes catalyze biochemical reactions through structural patterns that are directly reflected in their graph topology. The dataset provides a standardized structure in which graphs, nodes, edges, and labels are stored across multiple files, addressing the lack of a homogeneous representation format that commonly arises with graph data. The full dataset comprises 663 enzyme and 450 non-enzyme instances.

Table 4.1 reports global statistics of the dataset. The large variability in graph size and density, with node counts ranging from 4 to 620 and edge counts from 5 to 1049,

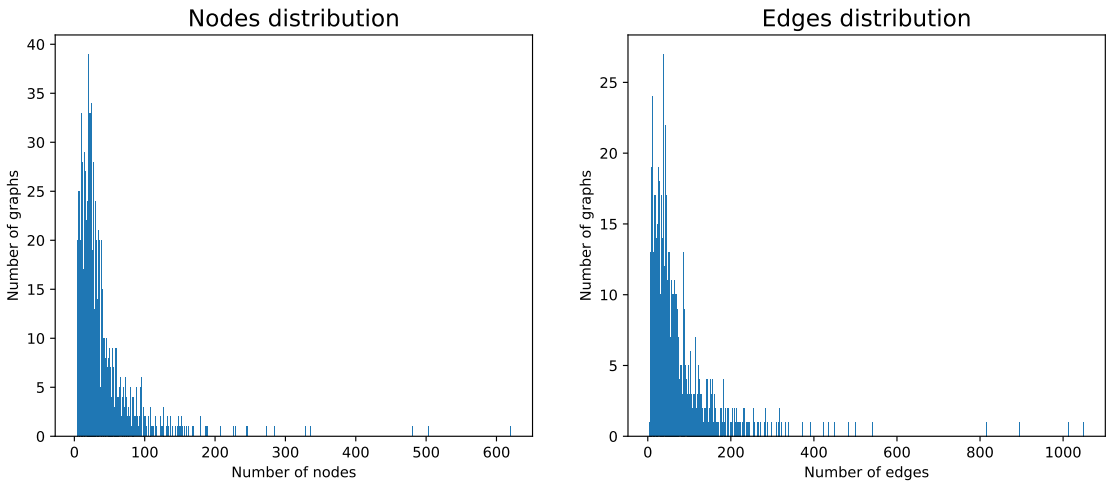


Figure 4.1: Distribution of graph sizes in the PROTEINS dataset, in terms of node count (left) and edge count (right).

poses additional challenges for both classical and quantum learning methods, and motivates the need for careful preprocessing before quantum execution.

Statistic	Value
Min. number of nodes	4
Avg. number of nodes	39.5
Max. number of nodes	620
Min. number of edges	5
Avg. number of edges	72.81
Max. number of edges	1049

Table 4.1: Graph size statistics of the PROTEINS dataset, highlighting the high variability in instance scale.

Two hardware-driven constraints further restrict the subset of usable instances. First, the Aquila platform supports a maximum of 256 qubits, imposing a direct upper bound on the number of nodes that can be embedded. As shown in Fig. 4.1, the majority of instances contain fewer than 100 nodes, indicating that a substantial portion of the dataset is compatible with the hardware capacity. Second, the computational cost of classical emulation of quantum evolution grows exponentially with qubit count, rendering simulation of large instances impractical.

To address these constraints, two reduced subsets are considered. PROTEINS256 retains only graphs with at most 256 nodes, matching the qubit capacity of Aquila, and contains 276 instances. PROTEINS12 further restricts the graph size to at most 12 nodes, a threshold chosen to enable efficient classical emulation of the quantum

dynamics, and contains 143 instances. Finally, both subsets are filtered to include only graphs that admit a unit-disk representation [46], which is a necessary condition for exact embedding into the neutral-atom register.

Preprocessing and Quantum-Compatible Embedding

The computation of the feature representation for each graph relies on sampling from a quantum state evolving under a time-dependent Hamiltonian. In particular, the evolution alternates between a mixing Hamiltonian $\mathcal{H}_\theta$ and an Ising Hamiltonian $\mathcal{H}_\mathcal{G}$, which explicitly depends on the graph topology.

In order to faithfully implement this evolution on neutral-atom hardware, the interaction structure of the device must directly reflect the adjacency relations of the input graph. This is achieved by mapping the graph to a UD (unit-disk) representation, so that the physical interactions induced by the Rydberg blockade reproduce the desired Ising Hamiltonian.

The first preprocessing step therefore consists in embedding each graph into the neutral-atom register, by finding a corresponding UD representation. The embedding procedure follows the approach introduced in Chapter 2, and is here adapted to satisfy the geometric constraints of the target hardware.

When targeting the Aquila platform, the embedding must comply with the following constraints:

- *Minimum vertical row spacing*: $D_{\text{row}} \geq 4\,\mu\text{m}$;
- *Minimum inter-atom distance*: $D_{\min} \geq 4\,\mu\text{m}$;
- *Register dimensions*: atom positions $\vec{p}_i$ must lie within a rectangular region of size $76\,\mu\text{m} \times 75\,\mu\text{m}$.

These constraints ensure that the resulting atomic configuration is physically realizable on the device while preserving, as much as possible, the adjacency structure of the original graph.

The final datasets are obtained by first filtering graphs based on their size, and then applying the DEN-based embedding procedure. Graphs for which no feasible embedding is found are discarded. The resulting datasets are summarized in Table 4.2. The resulting datasets exhibit a certain degree of class imbalance. This effect is particularly pronounced in PROTEINS12, where class 1 represents only 23% of the data. Although less severe, a similar imbalance is also present in PROTEINS256 (approximately 39%).

An example of the embedding process is shown in Figure 4.2, where an initial graph is transformed into a feasible UD representation compatible with the Aquila register. The discrete row structure imposed by the hardware is clearly visible.

It is important to note that the DEN (Distance Encoder Network) model does not guarantee convergence, as it relies on a neural approximation. Moreover, not all graphs admit a valid UD representation, especially under the geometric constraints imposed by

	Number of graphs	#Class 1	#Class 2
PROTEINS12	143	33	110
PROTEINS256	276	108	168

Table 4.2: Composition of the PROTEINS12 and PROTEINS256 subsets after unit-disk filtering, reporting the number of instances per class: enzyme (Class 1) vs. non-enzyme (Class 2).

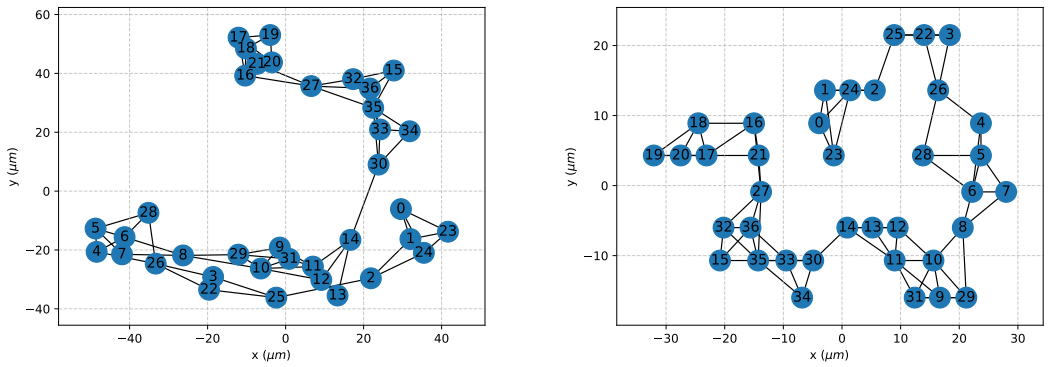


Figure 4.2: Original non-UD graph (left) and its UD embedding (right) produced by the adapted DEN model, with the row-discretized atom layout imposed by the Aquila architecture.

a two-dimensional register. As a result, part of the dataset is discarded during preprocessing. Nevertheless, the remaining instances are sufficient to perform a meaningful evaluation of the QEK framework.

4.2.2 Quantum Kernel Learning on Neutral-Atom Hardware

This section presents the implementation of a QEK on a neutral-atom platform, and its application to graph classification using a SVM. The approach relies on the idea that the time evolution of a quantum system can be used to construct expressive feature maps for structured data.

In this framework, each input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is encoded into the dynamics of a quantum system. The evolution is governed by a Hamiltonian, i.e., a Hermitian operator $\mathcal{H}$ acting on the Hilbert space of the system, which represents its total energy and determines its time evolution through the Schrödinger equation. By designing $\mathcal{H}$ to depend on the graph structure, the resulting quantum state encodes information about connectivity and topology.

More formally, the time evolution generated by the Hamiltonian $\mathcal{H}(t)$ is described by the unitary operator

$$U(t) = \mathcal{T} \exp\left(-\frac{i}{\hbar}\int_0^t \mathcal{H}(\tau)\, d\tau\right),$$

where $\mathcal{T}$ denotes time ordering. On neutral-atom hardware, this evolution is implemented through sequences of piecewise-constant or piecewise-linear control pulses.

To encode the graph structure, two Hamiltonian contributions are introduced:

$$\mathcal{H}_{\mathcal{G}} = \sum_{(i,j)\in\mathcal{E}} C_6 \frac{\hat{n}_i \hat{n}_j}{\|\vec{p}_i - \vec{p}_j\|^6}, \tag{4.1}$$

$$\mathcal{H}_{\theta} = \frac{\theta\hbar}{2} \sum_{i\in\mathcal{V}} (|g_i\rangle\langle r_i| + |r_i\rangle\langle g_i|) - \delta\hbar \sum_{i\in\mathcal{V}} \hat{n}_i, \tag{4.2}$$

where $\mathcal{H}_{\mathcal{G}}$ encodes pairwise interactions reflecting the edge structure of the graph via the Rydberg potential, θ plays the role of a time-dependent Rabi frequency controlling the strength of the mixing term, and δ is the global detuning, both consistent with the native laser control fields of neutral-atom hardware.

The full evolution is constructed as a sequence of alternating applications of these Hamiltonians. For a three-layer configuration, the evolution is parameterized by

$$\Lambda = \{\theta_0, t_0, \theta_1, t_1, \theta_2\},$$

where each t_k denotes the duration of a free evolution and each θ_k controls the strength of the mixing term. This layered structure defines a parametric family of quantum evolutions, as illustrated in Fig. 4.3.

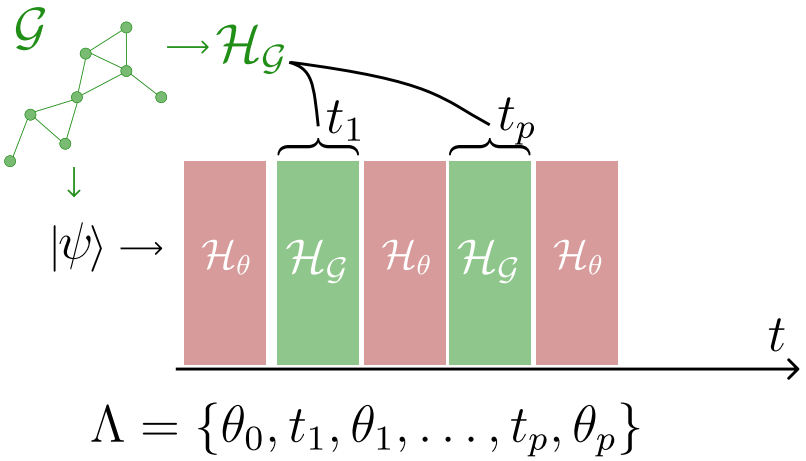


Figure 4.3: Schematic of the layered time evolution, depicting the successive application of each Hamiltonian and the associated parameters Λ.

At the end of the evolution, information is extracted through measurement. In quantum mechanics, a measurement is associated with an observable $\hat{\mathcal{O}}$, i.e., a Hermitian operator whose eigenvalues correspond to possible outcomes and whose eigenvectors define the measurement basis. Measuring the final state $|\psi_f\rangle$ with respect to $\hat{\mathcal{O}}$

yields outcomes $\{o_i\}$ with probabilities

$$p_i = |\langle o_i \mid \psi_f \rangle|^2.$$

On neutral-atom platforms, the natural observable is the local occupation operator $\hat{n}_i = |r_i\rangle\langle r_i|$, which indicates whether atom i is in the excited (Rydberg) state. Measuring all atoms in this basis produces bitstrings that can be directly interpreted as subsets of vertices of the graph, establishing a direct link between quantum measurements and graph configurations.

For a fixed choice of Λ, the overall hybrid quantum-classical workflow proceeds as follows:

1. **Quantum evolution:** for each graph $\mathcal{G}_i$, the corresponding atomic register is initialized and evolved according to the Hamiltonian defined by Λ. Repeated measurements produce a set of samples $M_{\mathcal{G}_i}$.

2. **Classical post-processing:** measurement outcomes are used to construct a probability distribution $\mathcal{P}_{\mathcal{G}_i}$, which captures structural information about the graph through the quantum dynamics.

3. **Kernel-based learning:** the distributions $\mathcal{P}_{\mathcal{G}_i}$ are used to compute similarities between graphs, forming a kernel matrix that is then used to train a SVM. Hyperparameters are selected through cross-validation.

More formally, the measurement process defines a probability distribution

$$\mathcal{P}_{\mathcal{G}}(\Lambda) = (p_1, \dots, p_K),$$

which is then used to compare graphs through the Jensen–Shannon divergence:

$$JS(\mathcal{P}, \mathcal{P}') = H\left(\frac{\mathcal{P} + \mathcal{P}'}{2}\right) - \frac{H(\mathcal{P}) + H(\mathcal{P}')}{2},$$

where the Shannon entropy is defined as

$$H(\mathcal{P}) = -\sum_k p_k \log p_k.$$

The resulting quantum kernel between two graphs $\mathcal{G}$ and $\mathcal{G}'$ is given by

$$\mathcal{K}_\mu(\mathcal{G}, \mathcal{G}') = \exp(-\mu\, JS(\mathcal{P}, \mathcal{P}'))\,, \tag{4.3}$$

where μ is a tunable hyperparameter controlling the sensitivity of the similarity measure.

The parameters Λ play a central role in determining the quality of the feature map induced by the quantum evolution. These parameters are optimized using a Bayesian optimization procedure on a subset of the dataset that can be efficiently simulated classically. Once identified, the same parameters are used to evaluate larger instances directly on Aquila, enabling the study of quantum feature representations on problem sizes that are not accessible through classical emulation.

4.2.3 Results on Quantum Kernel Learning

This section presents the experimental evaluation of the proposed quantum kernel approach. The objective is to assess both the expressivity of the quantum evolution as a feature map and its effectiveness in a downstream graph classification task.

The analysis is carried out in two stages. First, classical emulation is used to study the behavior of the kernel in a controlled setting, enabling parameter optimization and detailed inspection of the induced similarity measures. Then, the optimized configuration is executed on the Aquila neutral-atom platform to evaluate the performance under realistic hardware conditions. Together, these results provide insight into the capabilities and limitations of quantum evolution kernels for learning on graph-structured data.

Classical Emulation

The quantum evolution underlying the QEK is first studied through classical emulation, in order to validate the approach and optimize the pulse parameters before execution on quantum hardware. To this end, simulations are performed using *Bloqade* [170], a framework written in *Julia* [171] developed by QuEra for modeling and interacting with neutral-atom systems. The framework allows one to emulate quantum dynamics, define observables, and generate hardware-compatible pulse descriptions for execution on Aquila.

Despite its computational cost, this emulation step is necessary for two main reasons. First, it provides a controlled environment to assess the validity of the method. Second, it enables the optimization of the pulse parameters Λ, which would otherwise require a prohibitive number of quantum evaluations. In particular, each iteration of the Bayesian optimization procedure requires a total of $N_{\text{shots}} = n_{\text{graphs}} \times n_{\text{shots}} = 143 \cdot 10^3$ measurements, which represents a significant computational burden for current neutral-atom hardware and motivates the use of classical emulation for the optimization phase.

The overall hybrid quantum-classical workflow is summarized in Fig. 4.4. At each iteration, the Bayesian optimization procedure receives a new evaluation $(\Lambda, f(\Lambda))$, where $f(\Lambda)$ denotes the classification score, and updates the posterior distribution accordingly. The acquisition function then proposes the next parameter configuration to evaluate. This optimization is performed on the PROTEINS12 subset, where the reduced graph size renders classical emulation of the quantum evolution tractable.

Energy-based Probability Distributions. The core of the QEK construction is the comparison between probability distributions associated with different graphs. Since Aquila allows measurements only in the computational basis, the relevant observable (energy) is computed through classical post-processing of measurement outcomes.

Given a graph $\mathcal{G}$ and a set of measurement outcomes

$$M_{\mathcal{G}} = \{m_1, \dots, m_k\}, \quad m_i = (n_1, \dots, n_{|\mathcal{V}|}),$$

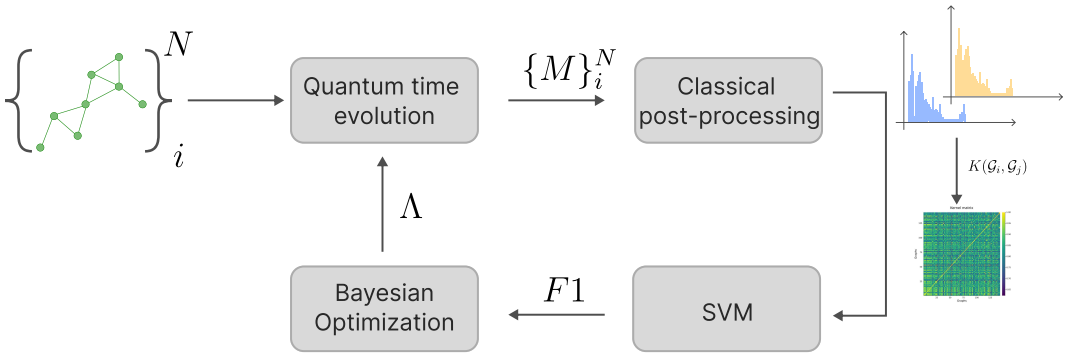


Figure 4.4: Hybrid quantum-classical workflow for Bayesian optimization of the kernel parameters Λ, combining classical emulation with a Gaussian process surrogate model.

where

$$n_i = \begin{cases} 1 & \text{if atom } i \text{ is measured in } |g\rangle\,, \\ 0 & \text{if atom } i \text{ is measured in } |r\rangle\,, \end{cases}$$

each measurement corresponds to a bitstring encoding a configuration of the system. Due to imperfect register initialization, the effective number of valid samples satisfies $k \leq n_{\text{shots}}$.

To each bitstring m_i, an energy value is associated:

$$e_i = \sum_{i<j} \frac{C_6}{|\overrightarrow{p_i} - \overrightarrow{p_j}|^6}\, n_i n_j,$$

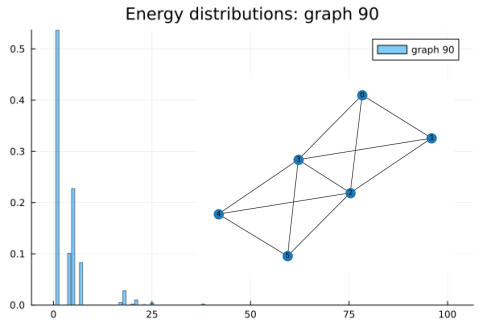
which reflects the interaction energy of the configuration. Repeating this process over all valid measurements yields a set of energies $E_{\mathcal{G}}$ for each graph.

In order to compare different graphs, their energy distributions must share a common support. Let

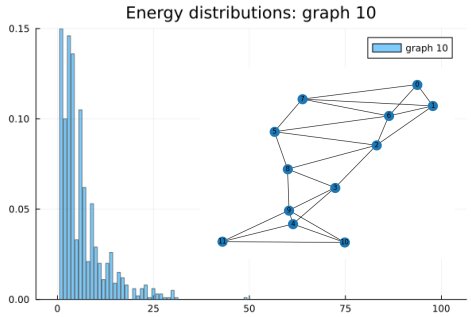
$$e_{\min} = \min_i\{\min E_{\mathcal{G}_i}\}, \qquad e_{\max} = \max_i\{\max E_{\mathcal{G}_i}\},$$

denote the global minimum and maximum energy values across the dataset. The interval $[e_{\min}, e_{\max}]$ is partitioned into 100 bins, and each distribution is normalized to obtain a probability distribution $\mathcal{P}(\mathcal{G})$.

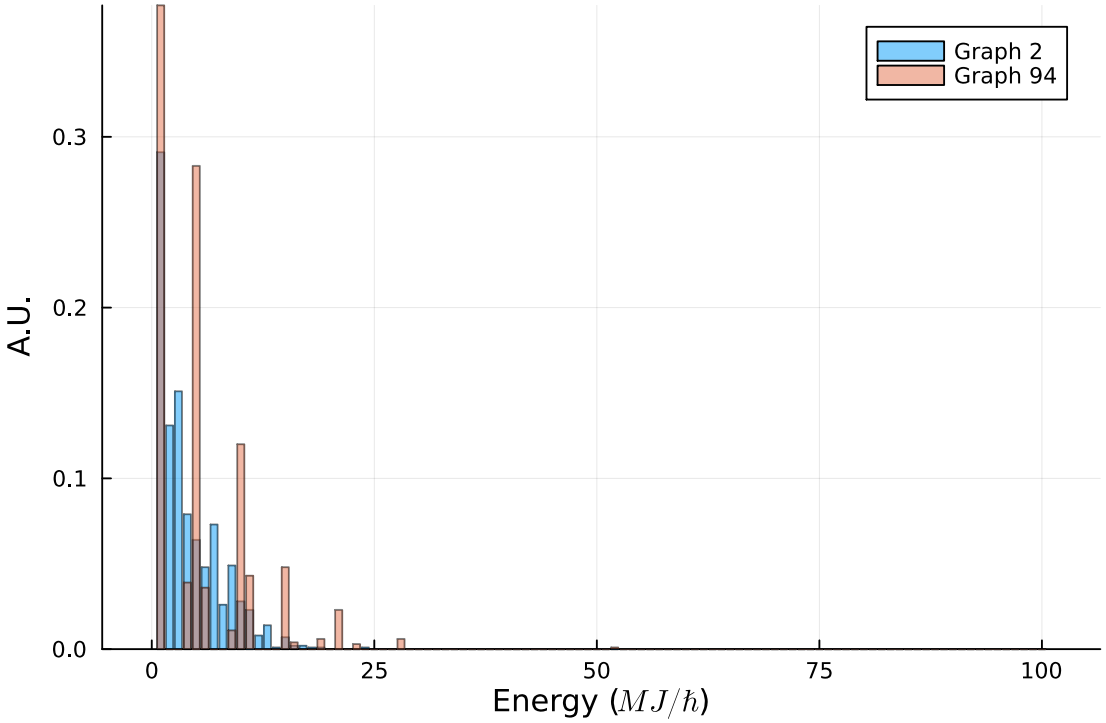
Examples of energy distributions obtained for enzyme and non-enzyme instances are shown in Fig. 4.5, illustrating the class-dependent structure captured by the quantum kernel evolution.



(a) Energy distribution for an enzyme instance.



(b) Energy distribution for a non-enzyme instance.



(c) Overlay of energy distributions for an enzyme and a non-enzyme instance, illustrating class discriminability.

Figure 4.5: Sampled energy distributions produced by the quantum kernel evolution for representative enzyme and non-enzyme graph instances from PROTEINS12.

Kernel Construction and Learning. Once the probability distributions are obtained, the quantum kernel is computed according to Eq. (4.3). Following [143], the hyperparameter is set to $\mu = 1$, yielding kernel values in the interval $[1/2, 1]$.

The kernel matrix is defined as

$$K_{i,j} = \mathcal{K}(\mathcal{G}_i, \mathcal{G}_j),$$

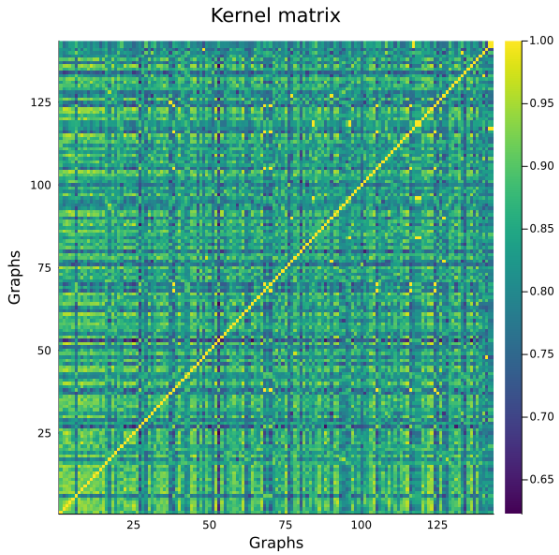


Figure 4.6: Kernel matrix of PROTEINS12 dataset, with $\mu = 1$.

and represents pairwise similarities between graphs. A visualization of the kernel matrix for the PROTEINS12 dataset is shown in Fig. 4.6. A clear structure emerges, with higher similarity values observed among graphs belonging to the same class (in particular enzymes), indicating that the quantum kernel captures relevant structural features.

The kernel matrix is then used to train an SVM. Model selection is performed using a K-fold cross-validation scheme with $K = 10$, combined with a grid search over two hyperparameters: the regularization parameter C, sampled over 100 logarithmically spaced values in $[10^{-4}, 10^{4}]$, and the class weights $w = [1, x]$, with x ranging over 30 values in $[1, 1000]$. The parameter C controls the trade-off between margin maximization and training errors, while the class weights mitigate the effect of class imbalance.

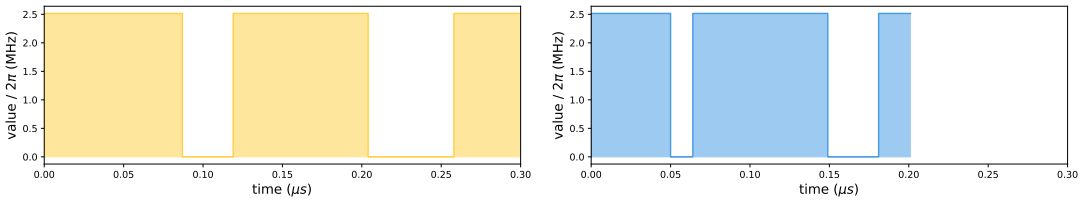
Pulse Optimization. The quantum evolution is defined by a sequence of alternating mixing and free-evolution periods. The parameterization used for optimization is

$$\Lambda_{\text{BO}} = \{\tau_0, t_0, \tau_1, t_1, \tau_2\},$$

where τ_i denote the durations of mixing pulses and t_i the durations of free evolution. The corresponding rotation angles satisfy $\theta_i = \Omega_0 \tau_i$.

The optimization objective is to maximize the classification performance (F1-score), viewed as a black-box function of Λ. Bayesian optimization [172] is used to efficiently explore the parameter space. The optimized waveform is compared to the reference configuration Λ_a from [143] in Fig. 4.7.

The optimized pulse exhibits a shorter total duration (approximately 200 ns compared to 317 ns), which is expected to reduce the impact of noise. Differences in the



(a) Baseline parameter set Λ_a. **(b)** Bayesian-optimized parameter set Λ_{BO}.

Figure 4.7: Control waveforms generated by the baseline and Bayesian-optimized parameter sets, illustrating the effect of optimization on the pulse shape.

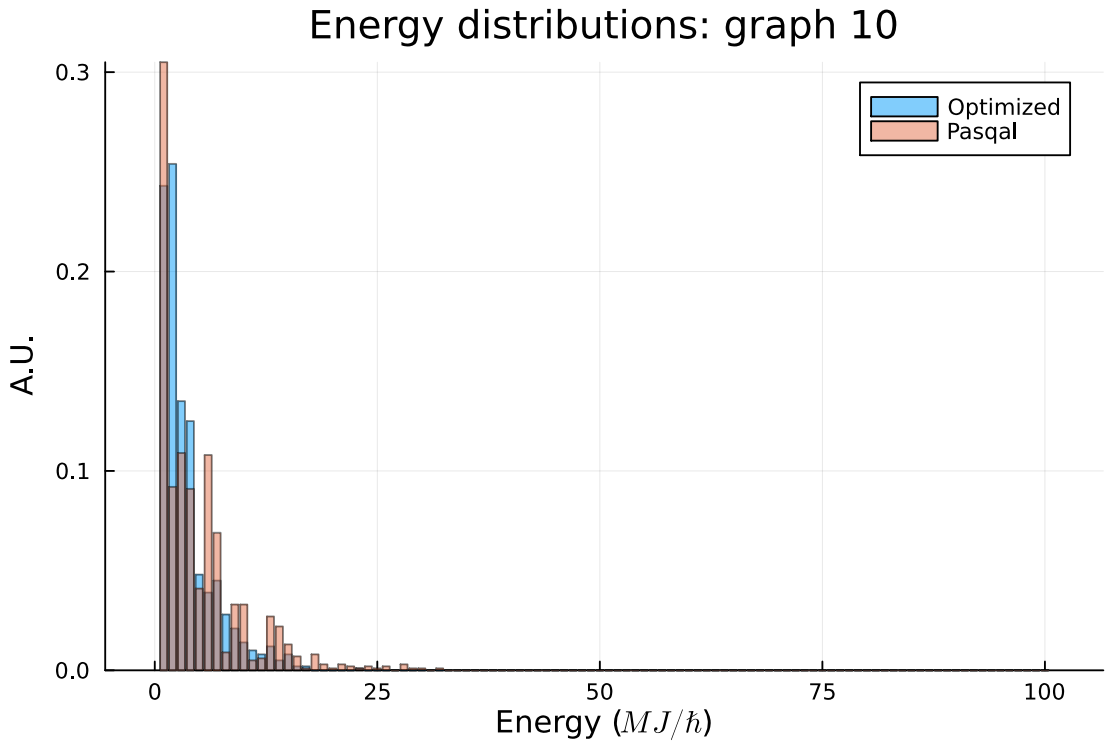


Figure 4.8: Energy distributions induced by the baseline parameter set Λ_a and the Bayesian-optimized set Λ_{BO}, highlighting the shift in the sampled distribution following optimization.

distribution of pulse durations also affect the resulting energy distributions, as illustrated in Fig. 4.8.

Classification Results. The performance of the proposed method is evaluated using standard metrics (F1-score, accuracy, precision, and recall) and compared against both the baseline quantum configuration Λ_a and a classical graph kernel, the SPK (Shortest Path Kernel) [173]. Results are reported in Table 4.3.

Due to the limited size of the PROTEINS12 subset, all methods exhibit relatively modest absolute performance. Nonetheless, the results reveal meaningful trends. The

	F-1 (%)	Accuracy (%)	Precision (%)	Recall (%)
Λ_a	41.9	44.8	29.3	89.1
Λ_{BO}	**46.3**	52.4	37.1	85
Shortest Path kernel	44.1	**61.4**	35	68

Table 4.3: Classification performance on PROTEINS12 for the baseline quantum kernel Λ_a, the Bayesian-optimized kernel Λ_{BO}, and the classical SPK, evaluated across four standard metrics.

Bayesian-optimized kernel Λ_{BO} achieves the highest F1 score of 46.3%, improving by approximately 4.4 percentage points over the baseline Λ_a and marginally outperforming the classical SPK. This improvement reflects the ability of Bayesian optimization to identify pulse parameters that induce more discriminative energy distributions, as illustrated in Fig. 4.8.

The higher raw accuracy of the SPK (61.4%) warrants careful interpretation. The PROTEINS12 subset is class-imbalanced, with non-enzyme instances outnumbering enzymes by a ratio of approximately 3:1 (see Table 4.2). In this setting, a classifier can achieve deceptively high accuracy by systematically favouring the majority class, which is reflected in the lower recall of the SPK (68%) compared to both quantum configurations. The F1-score, which balances precision and recall, is therefore the more informative metric here, and by this measure the optimized quantum kernel demonstrates competitive performance relative to the classical baseline.

Quantum Kernel Performance on Aquila

The proposed methodology is evaluated on the Aquila neutral-atom platform to assess its performance under realistic hardware conditions. Unlike the emulation study, which was confined to the reduced PROTEINS12 dataset, the hardware experiments are extended to the larger PROTEINS256 dataset, enabling a more comprehensive assessment of scalability on realistic graph sizes.

Interaction with the device is handled through the *BloqadeHardware* library, which translates the Hamiltonian description into a hardware-compatible format and verifies compliance with the underlying physical constraints. Each graph in the dataset corresponds to a single quantum evolution task on Aquila, and the total number of measurements is fixed to $N_{\text{graphs}} \times n_{\text{shots}} = 276 \cdot 10^3$ for each parameter configuration Λ_a and Λ_{BO}, consistently with the emulation setting (see Section 4.2.3).

Before analyzing the classification results, it is instructive to examine how the hardware execution compares to classical emulation. Since reconstructing the full quantum state would require an exponential number of measurements, the comparison is performed at the level of energy distributions derived from the sampled bitstrings. Figure 4.9 shows representative distributions for a randomly selected graph instance.

The two distributions share a similar overall structure, indicating that hardware noise does not fundamentally alter the energy landscape for this task. Nonetheless,

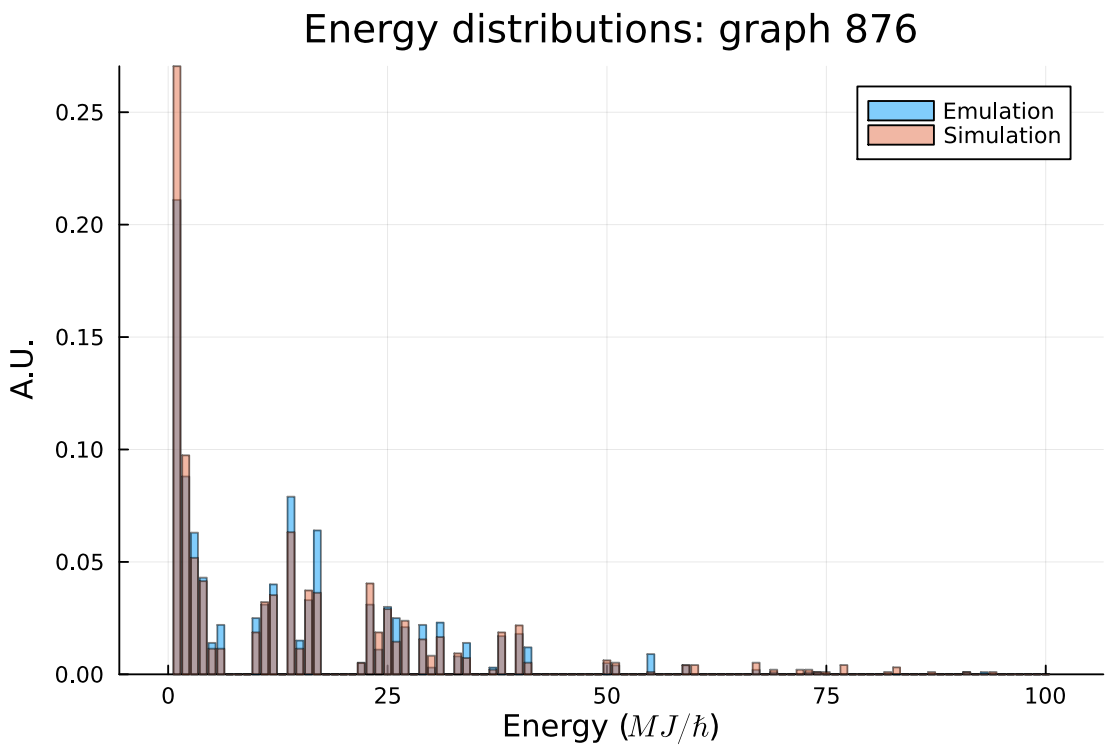


Figure 4.9: Energy distributions for a representative PROTEINS12 instance, contrasting noiseless classical emulation with hardware execution on Aquila under Λ_{BO}.

notable differences are apparent. The hardware execution exhibits a higher probability mass at low-energy configurations, attributable to coupling effects toward the ground state that are absent in noiseless emulation. Conversely, higher-energy configurations appear more frequently on hardware, likely due to detection errors that misclassify ground-state atoms $|g\rangle$ as excited states $|r\rangle$. Despite these discrepancies, the overall distributional similarity suggests that the quantum kernel retains its discriminative structure under realistic noise conditions.

Results on PROTEINS12. The classification pipeline is first applied to PROTEINS12 using the same procedure described in Section 4.2.3, with the parameter optimization step omitted for hardware execution. As discussed previously, the optimized configuration Λ_{BO} reduces the total evolution time by approximately 33% relative to Λ_a, which is expected to mitigate decoherence and noise accumulation. The results are reported in Table 4.4.

The hardware results confirm the effectiveness of the approach under realistic operating conditions. The optimized configuration Λ_{BO} achieves the best overall performance, improving the F1-score by approximately 3 percentage points over its emulated

	F-1 (%)	Accuracy (%)	Precision (%)	Recall (%)
Λ_a (Emulation)	41.9	44.8	29.3	89.1
Λ_{BO} (Emulation)	**46.3**	52.4	37.1	85
Shortest Path kernel	44.1	**61.4**	35	68
Λ_a (Aquila)	48.0	62.2	35.9	80.8
Λ_{BO} (Aquila)	**49.0**	**63.7**	**39.5**	78.0

Table 4.4: Classification performance on PROTEINS12 for emulation and Aquila hardware execution under Λ_a and Λ_{BO}, alongside the classical SPK baseline.

counterpart and outperforming both the classical SPK and the non-optimized configuration. A consistent improvement is observed across both parameter sets when moving from emulation to hardware execution, a counterintuitive finding that suggests certain noise effects may act as a form of implicit regularization, ultimately benefiting generalization in the classification task.

Results on PROTEINS256. The pipeline is then applied to the larger PROTEINS256 dataset, with results summarized in Table 4.5. Performance improves substantially relative to PROTEINS12, primarily due to the larger number of training samples and the more balanced class distribution. In this setting, the quantum kernel achieves results closely matching the classical SPK, with marginal advantages in F1-score, accuracy, and precision.

	F-1 (%)	Accuracy (%)	Precision (%)	Recall (%)
Λ_{BO} (Aquila)	**65.6**	**65.4**	**55.1**	86.6
Shortest Path kernel	65.3	64.9	53.5	**87.1**

Table 4.5: Classification performance on PROTEINS256, comparing the Aquila hardware execution of Λ_{BO} against the classical SPK baseline.

It is worth emphasizing that Λ_{BO} was optimized exclusively on PROTEINS12, yet generalizes well to the significantly larger PROTEINS256 instances. This cross-dataset transferability suggests that the optimized pulse parameters capture structural properties of the quantum evolution that are robust to graph size. Furthermore, the narrowing performance gap between quantum and classical kernels on the larger dataset is consistent with the theoretical expectation that quantum feature maps may confer a relative advantage in low-data regimes [174], where the expressive power of the quantum kernel is less diluted by abundant training signal.

Shot Efficiency Analysis. Finally, the sensitivity of classification performance to the number of measurements is investigated by applying a subsampling procedure to the hardware outcomes, simulating experiments with progressively reduced shot counts. The results are reported in Table 4.6.

Number of shots	F-1 (%)	Accuracy (%)	Precision (%)	Recall (%)
10	59.3	63.1	52.9	71.0
100	65.1	64.4	53.8	87.0
1000	**65.6**	**65.4**	**55.1**	86.6

Table 4.6: Sensitivity of classification performance on PROTEINS256 to the number of measurement shots, evaluated under Λ_{BO} on Aquila.

The results demonstrate a favorable shot-efficiency trade-off. Reducing the shot count from 1000 to 100 incurs only a marginal performance degradation across all metrics, while a more pronounced drop is observed at 10 shots. This indicates that a moderate number of measurements is sufficient to produce reliable energy distribution estimates, and that the approach is amenable to significant reductions in quantum resource usage without substantially compromising classification quality. This finding has practical implications for hardware deployment, where minimizing the number of QPU calls is a primary operational constraint.

4.3 Application II: Credit Risk Prediction

This section presents a second application of QML, based on the QRC paradigm, in a setting that differs significantly from the graph-based problems considered previously. Here, the focus is on tabular data and on a real-world financial task, namely credit risk prediction. The objective is to determine whether a customer will default on their credit card payment in the following month, a problem that is both practically relevant and inherently challenging due to the strong imbalance between default and non-default cases.

The dataset considered in this study is publicly available [175] and has been widely used in the literature [176, 177, 178, 179]. This enables direct comparison with a broad range of classical machine learning approaches, including KNN (K-Nearest Neighbours), logistic regression, discriminant analysis, Naïve Bayes, ensemble methods, and neural networks. In particular, DNNs provide a strong nonlinear baseline for this task.

In this context, the goal is not to replace classical models, but to investigate whether quantum dynamics can provide an alternative feature transformation that enhances classification performance. The approach follows the QRC framework, where a fixed dynamical system processes the input data and learning is performed only at the classical level. This makes the methodology particularly suitable for NISQ devices, as it avoids the need for variational training of quantum parameters.

The implementation relies on the neutral-atom platform Aquila [29], which provides up to 256 qubits arranged in a programmable two-dimensional array. A key feature of this device, already exploited in the optimization application discussed in Chapter 3, is the possibility of applying site-dependent control through local detuning. This

capability enables flexible encoding of classical data into the system dynamics.

The quantum evolution is governed by the same Hamiltonian introduced in Section 3.3.2 (see Eq. (3.44)), allowing a direct reuse of the physical model across different applications. However, in contrast to the optimization setting, where the interaction graph encodes the problem structure, here the connectivity is fixed and the input data are injected through the control parameters.

Two encoding strategies are considered and evaluated independently. In the first case, classical features are mapped to the local detuning $\delta_i(t)$, directly affecting the excitation dynamics of each atom. In the second case, the encoding is performed through small perturbations of the atomic positions $\vec{p}_i$, which modify the interaction terms V_{ij} and thus indirectly influence the system evolution. In both cases, the Hamiltonian becomes input-dependent, and the resulting quantum dynamics implement a nonlinear transformation of the data.

The proposed approach is therefore based on a hybrid quantum-classical pipeline, where the quantum system acts as a reservoir that produces feature representations, and a classical model performs the final classification. The impact of hardware noise is explicitly evaluated by comparing results obtained on Aquila with those obtained through noiseless classical emulation. This allows assessing the robustness of the approach under realistic NISQ conditions.

Overall, this application highlights how the same neutral-atom platform can be used beyond combinatorial optimization, supporting learning tasks where the role of the quantum system is to provide expressive, data-driven feature maps rather than directly encode a problem structure.

4.3.1 Quantum Reservoir for Credit Card Default Classification

This section describes the hybrid quantum-classical pipeline used to address the credit risk prediction task. The methodology follows the QRC paradigm and is structured so as to isolate the contribution of the quantum reservoir with respect to classical baselines.

At a high level, the pipeline consists of three main stages:

- **Preprocessing**: the raw dataset is cleaned and transformed using standard techniques, producing a set of normalized and informative features suitable for downstream processing;
- **Reservoir computing layer**: the preprocessed features are mapped into a higher-dimensional representation. This step is performed either by the quantum system or by a classical analogue, enabling a direct comparison between the two approaches;
- **Classification**: simple classifiers are trained on the transformed features to perform the binary prediction task.

A schematic overview of the pipeline is shown in Fig. 4.10. The design of each stage aims at ensuring a fair comparison between quantum and classical approaches, while keeping the overall workflow consistent with the principles of QRC, where the reservoir is fixed and only the classical readout is trained.

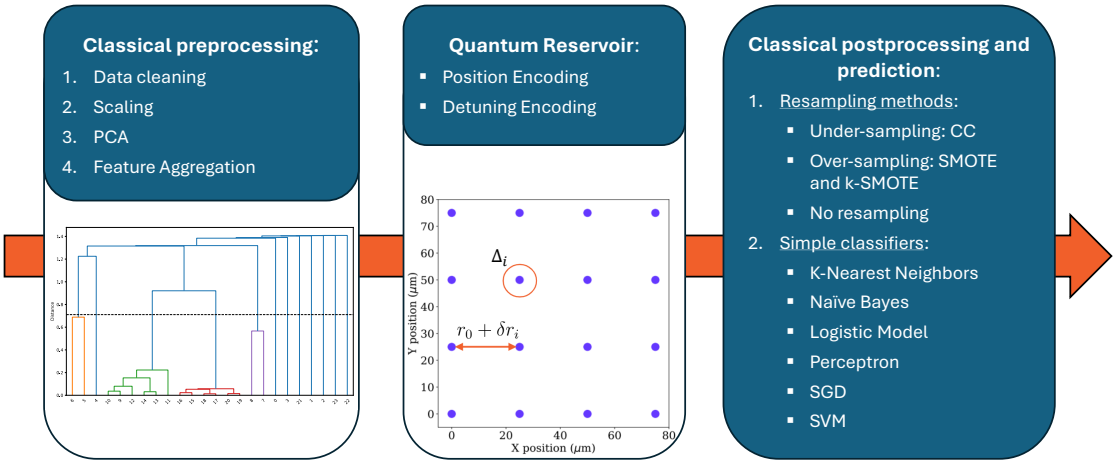


Figure 4.10: Hybrid quantum-classical pipeline for credit risk prediction.

To better assess the contribution of the quantum layer, two additional classical baselines are considered:

- **CRC (Classical Reservoir Computing)**: the quantum reservoir is replaced by a classical dynamical system, while keeping the rest of the pipeline unchanged;
- **FC (Fully Classical)**: the classifiers are applied directly to the preprocessed features, without any reservoir transformation.

These baselines allow disentangling the effect of the reservoir transformation from that of the classifier itself. In addition, a DNN (Deep Neural Network) model is included as a nonlinear benchmark, providing a reference for comparison with more expressive classical architectures.

Benchmark Dataset

Credit card default prediction is a well-documented problem in the machine learning literature, with several public benchmarks available for evaluating classification methods. In this work, we adopt the dataset introduced in [175], which contains records of credit card clients in Taiwan over the period from April to September 2005.

The dataset comprises 30,000 samples, each described by 23 features and a binary label indicating whether the client defaulted on their payment in the following month. The features span four categories:

- **Credit data:** the amount of credit extended (in NT dollars);
- **Demographic factors:** sex, education level, marital status, and age;
- **Repayment history:** six monthly repayment status values (*PAY_1–PAY_6*), where -1 denotes timely payment and values 1–9 indicate delays of one to nine months;
- **Financial history:** six monthly bill statement amounts (*BILL_AMT1–BILL_AMT6*) and six previous payment amounts (*PAY_AMT1–PAY_AMT6*).

Upon inspection, we identified inconsistencies in several features. The *MARRIAGE* feature, documented with values 1 (married), 2 (single), and 3 (other), contained 54 samples carrying an undocumented value of 0. Similarly, *EDUCATION*, documented with values 1–4, included 345 samples with undocumented values (0, 5, or 6). Given their small proportion relative to the full dataset, these samples were removed. The repayment history features (*PAY_1–PAY_6*) also contained values outside the documented range; as these affected over 10,000 samples, removal was impractical, and those values were retained. After cleaning, the dataset reduces to 29,601 samples, hereafter referred to as CARDS_30000.

Due to the limited availability of quantum resources, the full dataset could not be executed on Aquila. Two nested subsets were therefore constructed for experiments on quantum hardware: CARDS_2500 (2,571 samples) and CARDS_1000 (1,000 samples), both stratified by class label to preserve class distribution. The three datasets satisfy the inclusion relation:

$$\text{CARDS_1000} \subseteq \text{CARDS_2500} \subseteq \text{CARDS_30000}.$$

This experimental design also allows us to assess QRC performance under limited data availability, a setting in which reservoir computing is expected to have an advantage, as it typically requires fewer training samples than conventional approaches [180].

4.3.2 Preprocessing

The preprocessing pipeline is designed to provide the reservoir with the most informative and compact feature representation possible. It consists of four sequential stages: feature scaling, dimensionality reduction via PCA (Principal Component Analysis), feature aggregation via clustering, and class balancing via resampling.

Feature Scaling

The dataset contains two nominal categorical variables, *SEX* and *MARRIAGE*, which lack a natural ordering. These are converted into boolean features using one-hot encoding, expanding the total feature count from 23 to 24. The *EDUCATION* feature, being ordinal, is retained as a numerical variable to preserve its inherent ordering and avoid an unnecessary increase in dimensionality. All numerical features are then scaled to the range $[-1, +1]$ using an absolute value scaler, as required by the subsequent PCA step. The implementation relies on Python's `scikit-learn` library [181].

Dimensionality Reduction via PCA

Principal Component Analysis [182] is applied to the scaled features to remove redundancy and identify the directions of maximum variance in the data. Components are retained iteratively until each additional component contributes less than 0.001 to the total explained variance; this criterion yields 11 principal components, collectively capturing approximately 99% of the variance, as illustrated in Fig. 4.11.

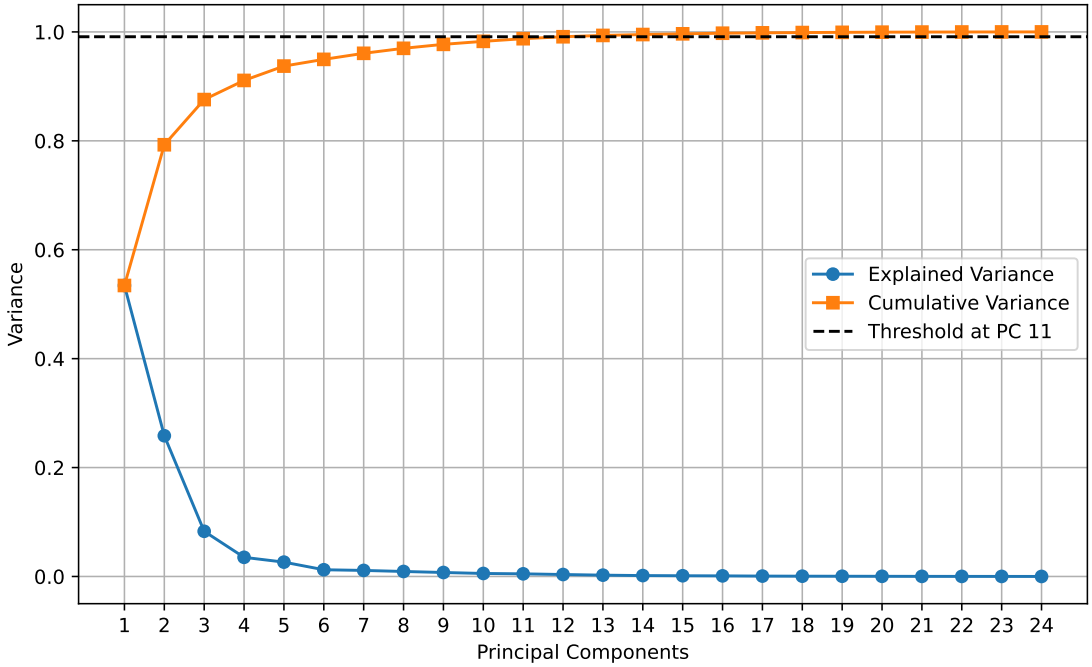


Figure 4.11: PCA scree plot for CARDS_30000, reporting the cumulative variance retained as a function of the number of retained components.

Feature Aggregation via Clustering

Rather than directly using the principal components as input features, we instead leverage the PCA scores to identify groups of correlated original features, which are then aggregated. Specifically, agglomerative (bottom-up hierarchical) clustering is applied in the 11-dimensional principal component space, using a distance threshold set to half the maximum pairwise distance among features. This process groups the 24 features into 12 clusters, each capturing a coherent pattern of correlated information; the resulting clusters are reported in Table 4.7, and visualised in the correlation circle of Fig. 4.12, where features belonging to the same cluster share the same colour. Within each cluster, the scaled feature values are averaged to produce a single representative feature, yielding a final input dimensionality of 12.

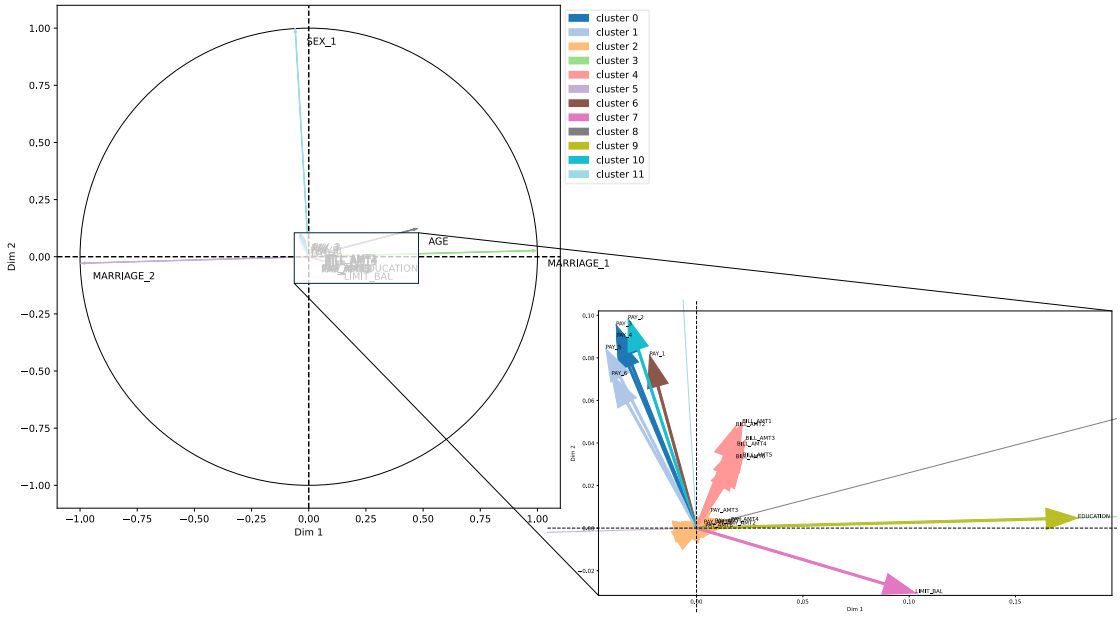


Figure 4.12: Correlation circle of feature loadings on the first two principal components for CARDS_30000, with colors denoting cluster membership.

This two-stage reduction — first via PCA, then via clustering — serves a dual purpose. It eliminates non-informative features and consolidates redundant ones, while also enabling more efficient sample encoding on the QPU (Quantum Processing Unit): with fewer input features, multiple replications of the same sample can be accommodated within the QPU register, reducing the quantum resources required. The subsequent reservoir computing step then expands the representation back into a higher-dimensional space, with the explicit goal of capturing non-linear feature interactions that lie beyond the reach of the linear methods applied here.

Table 4.7: Feature groupings found by agglomerative clustering on CARDS_30000.

Cluster	Features
Cluster 0	*PAY_3*, *PAY_4*
Cluster 1	*PAY_5*, *PAY_6*
Cluster 2	*PAY_AMT1*, ..., *PAY_AMT6*
Cluster 3	*MARRIAGE_1*
Cluster 4	*BILL_AMT1*, ..., *BILL_AMT6*
Cluster 5	*MARRIAGE_2*
Cluster 6	*PAY_1*
Cluster 7	*LIMIT_BAL*
Cluster 8	*AGE*
Cluster 9	*EDUCATION*
Cluster 10	*PAY_2*
Cluster 11	*SEX_1*

4.3.3 Reservoir Encoding and Feature Extraction

After preprocessing, each data sample is represented by a feature vector of reduced dimensionality. The purpose of the reservoir computing layer is to transform these features into a higher-dimensional representation, capturing nonlinear correlations that are not accessible through classical linear transformations.

Within the QRC paradigm, the quantum system is not trained. Instead, it acts as a dynamical system that processes the input through its intrinsic evolution, while learning is delegated to a classical model. In this context, the key design choice is how to encode classical features into the quantum system.

Quantum Reservoir Computing

A variety of QRC approaches have been proposed in recent years. In terms of scale, Kornjača et al. [168] demonstrated execution on neutral-atom platforms in the hundreds-qubit regime, suggesting potential advantages in scalability over classical ML methods. Motivated by these results, a similar approach is adopted in this classification task.

To execute a QRC routine, classical features obtained after preprocessing must be encoded into the quantum system. For the considered platform, Aquila [29], two different encoding strategies are explored independently.

In both cases, atoms are arranged along a one-dimensional array, such that interactions are primarily limited to nearest neighbors. Using the Hamiltonian defined in Eq. (3.44), the two encoding strategies are defined as follows:

- *Position encoding*: this approach encodes features into the inter-atomic distances. For a given sample, the distance between the i^{th} and $(i+1)^{th}$ atoms is defined

as

$$r_{i,i+1} = r_0 + \delta r_i = r_0(1 + \lambda f^i), \tag{4.4}$$

where f^i is the i^{th} feature, λ is a scaling factor, and r_0 is the reference distance. This encoding modifies the interaction term V_{ij} through the dependence on $|\vec{p}_i - \vec{p}_j|$. In this case, a uniform global detuning is applied, i.e.,

$$\delta_i(t) = \delta(t), \quad \forall i. \tag{4.5}$$

Encoding an n-dimensional sample requires $n + 1$ atoms.

- *Detuning encoding*: in this strategy, features are directly mapped onto the local detuning amplitudes:

$$\delta_i(t) = \delta(t) + \delta_l f^i, \tag{4.6}$$

where δ_l is a scaling coefficient. In this case, the inter-atomic distances remain fixed,

$$r_{i,i+1} = r_0, \quad \forall i, \tag{4.7}$$

and the number of atoms is equal to the number of features.

Once the input is encoded, the system is initialized in the ground state $|\psi_{t=0}\rangle = |0 \dots 0\rangle$ and evolves according to Eq. (3.44). During the evolution, expectation values of local and pairwise observables are computed at discrete timesteps:

$$\langle Z_i \rangle = \langle \psi | Z_i | \psi \rangle, \quad \langle Z_i Z_j \rangle = \langle \psi | Z_i Z_j | \psi \rangle. \tag{4.8}$$

These quantities define the output of the reservoir for a given sample.

For $n = 12$ input features, the detuning encoding produces

$$n + \binom{n}{2} = 12 + 66 = 78$$

features per timestep. Considering 5 timesteps, the total number of output features is

$$78 \times 5 = 390.$$

In the case of position encoding, one additional atom is required, yielding

$$(n+1) + \binom{n+1}{2} = 13 + 78 = 91$$

features per timestep, for a total of

$$91 \times 5 = 455$$

features.

The resulting high-dimensional representations are then used as input to the classification stage.

To define the quantum evolution, the control parameters $\Omega(t)$ and $\delta_i(t)$ must be specified. A piecewise pulse profile is used for both quantities, consisting of a rapid ramp-up and ramp-down of $0.05\,\mu s$ to a maximum amplitude, respectively Ω_{max} for the rabi pulse and δ_{global} for $\delta(t)$, as shown in Fig. 4.13. To remain within the coherence time of the device (approximately $4\,\mu s$), the evolution is discretized into 5 timesteps of $0.5\,\mu s$ each.

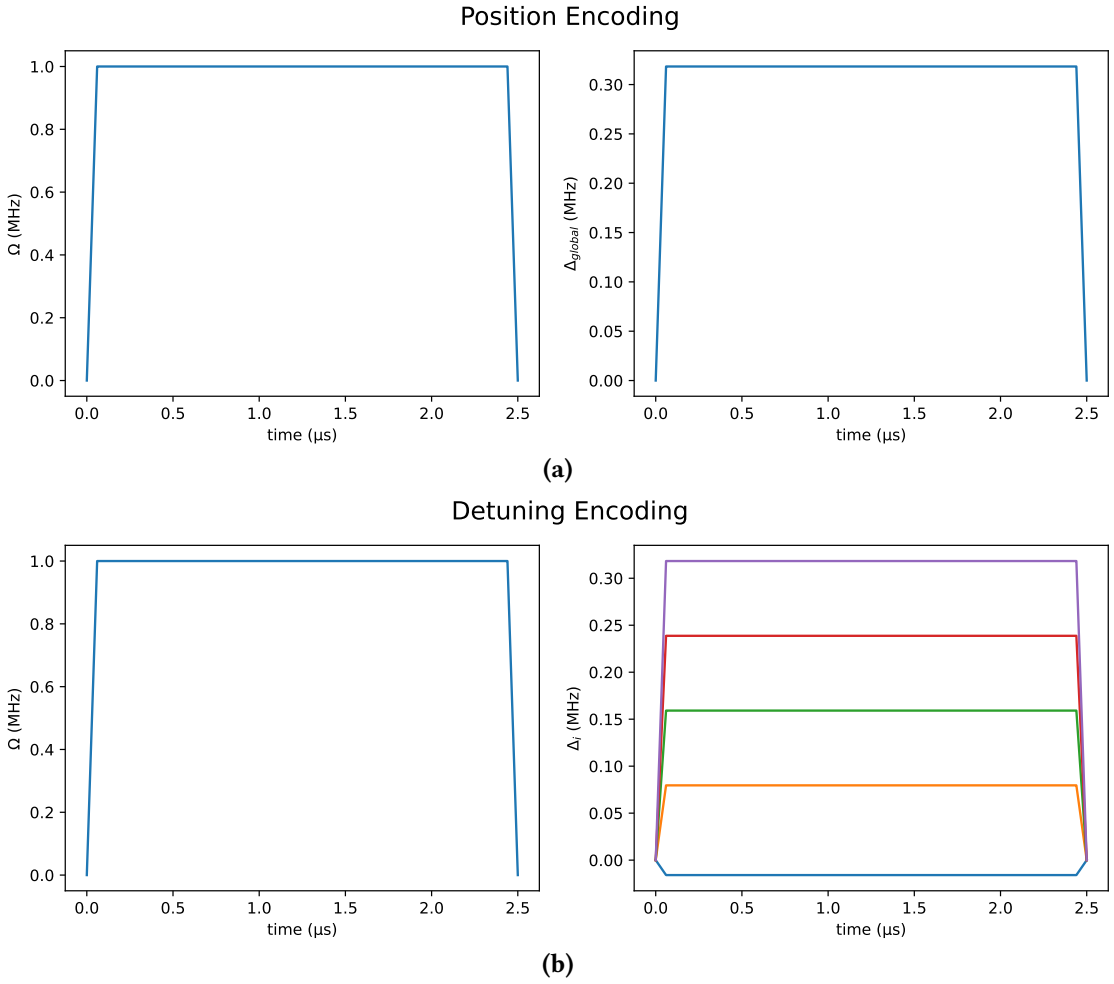


Figure 4.13: Illustrative pulse sequences for position encoding (a) and local detuning encoding (b), with the latter displayed on a reduced qubit register for visual clarity.

Classical Reservoir Computing

To assess the contribution of the quantum reservoir, a classical counterpart is also considered. The CRC model is obtained by mapping each qubit to a classical spin vector $\hat{\mathcal{S}}$, following [168].

Starting from the quantum Hamiltonian in Eq. (3.44), the corresponding classical dynamics are described by:

$$\frac{\partial \hat{\mathcal{H}}}{\partial \hat{\mathcal{S}}_i} = \frac{\Omega(t)}{2}\hat{x} + \left[-\frac{\delta_i(t)}{2} + \sum_{i \neq j} \frac{V_{ij}}{4}\left(1 + \hat{\mathcal{S}}_j^{(z)}\right)\right]\hat{z}. \tag{4.9}$$

The evolution of the N classical spins is computed numerically by solving the corresponding system of differential equations. Using the same parameters as in the quantum case, the projections along the z-axis and their pairwise correlations are extracted at each timestep.

The role of the classical reservoir within the pipeline is identical to the quantum one: preprocessed features are mapped into a higher-dimensional representation, which is then used as input for the classification stage.

4.3.4 Classification

The final stage of the pipeline focuses on the binary classification task. At this point, each sample has been transformed by the reservoir computing layer into a high-dimensional feature representation, capturing nonlinear relationships among the original variables. These features constitute the input to the classification models.

A key challenge in this setting is the strong class imbalance of the dataset. To address this issue, a resampling step is introduced prior to training. Importantly, resampling is applied to the feature representations obtained after the reservoir transformation, ensuring that both quantum and classical pipelines are treated consistently.

Class Balancing via Resampling

An inherent difficulty in many machine learning tasks is class imbalance, which tends to skew model predictions toward the majority class. The CARDS_30000 dataset exhibits a significant imbalance, with only 6,605 defaulters (class 1) out of 29,601 samples.

To allow models to learn effectively from both classes, resampling techniques are applied exclusively to the training set, leaving the validation and test sets unchanged. The same approach is adopted for the smaller subsets CARDS_2500 and CARDS_1000. The resulting class distributions are reported in Tables 4.8, 4.9, and 4.10. All resampling methods are implemented using the `imblearn` library [183].

Both oversampling and undersampling strategies are considered.

Table 4.8: Training class distribution by resampling strategy for CARDS_30000.

Resampling Method	Training Set	Class 1	Class 0
None	20720	4623	16097
SMOTE	32194	16097	16097
K-SMOTE	32198	16101	16097
Cluster Centroids	9246	4623	4623

Table 4.9: Training class distribution by resampling strategy for CARDS_2500.

Resampling Method	Training Set Size	Class 1	Class 0
None	1799	402	1397
SMOTE	2794	1397	1397
K-SMOTE	2797	1400	1397
Cluster Centroids	804	402	402

Table 4.10: Training class distribution by resampling strategy for CARDS_1000.

Resampling Method	Training Set Size	Class 1	Class 0
None	700	156	544
SMOTE	1088	544	544
K-SMOTE	1091	547	544
Cluster Centroids	312	156	156

Oversampling Techniques. Oversampling artificially increases the representation of the minority class by generating synthetic samples. We evaluate two variants of the SMOTE (Synthetic Minority Over-sampling Technique) [184].

SMOTE generates synthetic samples by interpolating between a minority instance x and one of its k-nearest neighbors x_k:

$$x_{\text{new}} = x + \lambda(x_k - x), \quad \lambda \sim \mathcal{U}(0,1),$$

with $k = 5$ in our setting.

While effective, SMOTE may over-represent dense regions of the minority class and generate noisy samples near class boundaries. To mitigate these issues, we also consider K-SMOTE (K-Means SMOTE), which first clusters minority samples using K-Means (with $K = 2$) and then applies SMOTE within each cluster. This preserves the structure of the minority class and reduces the risk of generating unrealistic samples.

Undersampling Techniques. Undersampling reduces the size of the majority class to balance the dataset. We employ the CC (Cluster Centroids) method [185], which applies K-Means clustering to the majority class and replaces groups of samples with their centroids. The number of clusters is chosen to match the size of the minority class, resulting in a balanced dataset. This approach retains only the most representative majority samples while preserving the overall feature space structure.

Classification Models

The features generated by the reservoir computing layer are used as input to a set of classical classifiers. In line with the QRC paradigm, preference is given to models that are simple to train, so that performance differences can be primarily attributed to the feature transformation rather than to the classifier complexity.

All models are implemented using the `scikit-learn` library [181]. Hyperparameters are selected via grid search using the F1-score on the validation set, which is more suitable than accuracy for imbalanced datasets. Where applicable, class weights are used to compensate for class imbalance. The explored configurations are summarized in Table 4.11.

- **Support Vector Machine (SVM).** SVM is a linear classifier that separates the two classes using a hyperplane maximizing the margin between them. When perfect separation is not possible, a soft-margin formulation is adopted, controlled by the regularization parameter C. Larger values of C penalize misclassification more strongly. Class weights are included to account for imbalance.

- **Naïve Bayes (NB).** The Gaussian NB (Naïve Bayes) classifier models each feature as normally distributed, assuming conditional independence among features. No hyperparameter tuning is required, and training and validation sets are merged to improve estimation.

- **Logistic Model (LM).** The logistic regression model is a parametric, discriminative binary classifier that models the probability of the positive class through the logistic (sigmoid) function:

$$P(y = 1 \mid \mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}, \tag{4.10}$$

  where $\mathbf{x}$ is the input feature vector, $\mathbf{w}$ the weight vector, and b the bias term. The probability of the negative class is given by $P(y = 0 \mid \mathbf{x}) = 1 - P(y = 1 \mid \mathbf{x})$. The model parameters are estimated by minimizing the logistic loss function. To improve generalization and avoid overfitting, an ℓ_2 regularization term is introduced, leading to the following objective:

$$\min_{\mathbf{w}} \sum_i \log\left(1 + e^{-y_i(\mathbf{w}^T \mathbf{x}_i + b)}\right) + \frac{1}{2C}\|\mathbf{w}\|_2^2. \tag{4.11}$$

  The hyperparameter $C > 0$ controls the trade-off between data fitting and regularization. In particular, smaller values of C enforce stronger regularization, leading to smoother decision boundaries; larger values of C reduce the regularization effect, allowing the model to better fit the training data.

- **Perceptron (P).** The Perceptron is a linear classification algorithm based on an online learning paradigm. It aims to find a separating hyperplane between the two classes by iteratively updating its parameters when misclassifications occur. The decision function is defined as:

$$y = \text{sign}(\mathbf{w}^T\mathbf{x} + b). \tag{4.12}$$

  During training, the model processes one sample at a time. If a sample is misclassified, the weights are updated according to:

$$\mathbf{w} \leftarrow \mathbf{w} + \eta\, y\, \mathbf{x}, \tag{4.13}$$

  where η is the learning rate, which controls the magnitude of the update. In this work, the Perceptron is implemented using a stochastic gradient descent scheme without regularization, and trained for a fixed number of epochs.

- **Stochastic Gradient Descent (SGD).** SGD (Stochastic Gradient Descent) is used to train a linear classifier with Hinge loss:

$$L(\mathbf{w}) = \sum_i \max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)) + \lambda\|\mathbf{w}\|_2^2. \tag{4.14}$$

  Compared to the Perceptron, SGD updates weights at every iteration and includes regularization.

  K-Nearest Neighbors (KNN). KNN assigns a class based on the majority label among the k nearest samples in feature space.

Table 4.11: Hyperparameter configurations

Model	Hyperparameter	Values Tested	Class Weights
KNN	k	$\{2, 3, \ldots, 15\}$	No
SGD	Epochs	$\{10, 20, \ldots, 100\}$	Yes
Naïve Bayes	None	–	No
Logistic Model	C	$\{10^{-3}, \ldots, 10^{2}\}$	Yes
Perceptron	η	$\{10^{-1}, 10^{-2}, 10^{-3}\}$	Yes
SVM	C	$\{10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$	Yes

Deep Neural Network Baseline. In addition to these models, a deep learning baseline is considered to evaluate the performance of the proposed pipeline against a fully classical nonlinear approach. The DNN model is applied directly to the preprocessed features, without any reservoir transformation. This allows assessing whether the

quantum or classical reservoir provides an advantage over standard deep learning approaches.

The architecture consists of six fully connected layers with ReLU activations and dropout regularization, with hidden sizes $(64, 128, 256, 128, 64)$. The model is implemented using `PyTorch` [186].

Training is performed using cross-entropy loss and the Adam optimizer, with early stopping (patience of 30 epochs). Hyperparameters are selected via grid search over:

- learning rate: $\{0.1, 0.01, 0.001\}$,
- batch size: $\{64, 128, 256\}$.

The best-performing configuration on the validation set is then evaluated on the test set.

4.3.5 Performance Evaluation and Hardware Validation

The experimental evaluation of the proposed hybrid QRC pipeline is conducted in two stages. First, extensive classical emulation is performed to assess the methodology in a noiseless setting, enabling parameter tuning and benchmarking on large datasets. Then, the approach is executed on the Aquila QPU to evaluate its performance under realistic hardware conditions.

Classical Emulation Results

The first stage consists of a large-scale emulation campaign using the `CARDS_30000` dataset. This phase allows evaluating both classification performance and computational cost in a controlled environment, as well as defining suitable parameters for the encoding strategies.

The noiseless emulation of the quantum dynamics is performed using the *Julia* implementation of *Bloqade* [187]. Given that both encoding strategies are anticipated to yield comparable performance under ideal conditions [168], experiments on the full dataset are carried out using detuning encoding only. Position encoding is additionally evaluated on the smaller `CARDS_1000` dataset, which is also used for hardware experiments.

The physical parameters are fixed to $r_0 = 10\,\mu m$ and $\Omega_{\text{max}} = 2\pi\,\text{rad}/\mu s$, following [168]. Additional tuning is performed for each encoding scheme. For **position encoding**, the optimal configuration is obtained for $\lambda = 1.0$ and $\delta_{\text{global}} = 2.0\,\text{rad}/\mu s$, providing a good trade-off between performance and compatibility with the Aquila QPU. For **detuning encoding**, varying $\delta_{\text{global}} \in [0, 9]\,\text{rad}/\mu s$ has limited impact on classification performance, and $\delta_{\text{global}} = \pi\,\text{rad}/\mu s$ is selected.

Shot-based Estimation. The QRC features $\langle\psi|Z_i|\psi\rangle$ and $\langle\psi|Z_iZ_j|\psi\rangle$ are computed both using statevector simulations and shot-based measurements. The latter is used to estimate the number of measurements required to accurately reconstruct the feature space under realistic conditions.

Fig. 4.14 shows the F1-score as a function of the number of measurements. The results indicate that performance reaches a plateau after a few hundred shots, suggesting that accurate feature estimation can be achieved without excessive sampling. Based on this analysis, the number of shots per timestep and per data point is fixed to 240, in order to balance accuracy and QPU resource utilization.

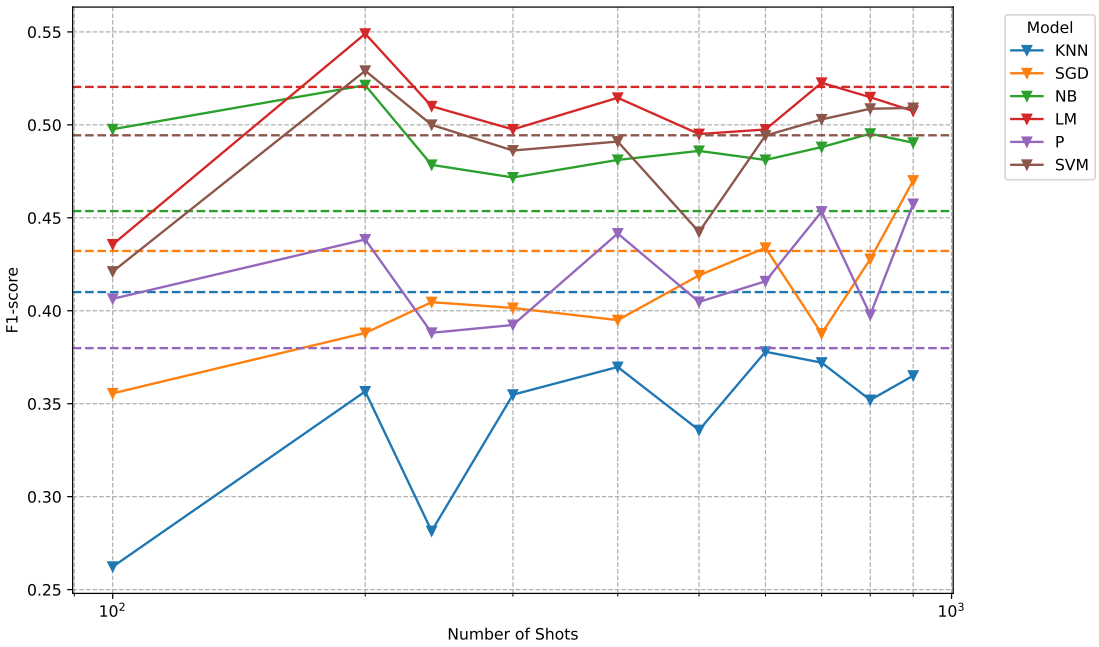


Figure 4.14: F1 score vs. number of Hamiltonian simulations, with dotted lines indicating noiseless statevector emulation results as a reference baseline.

Classification Performance and Training Cost. The classification results obtained on the CARDS_30000 dataset are reported in Fig. 4.15. The F1-score achieved by the emulated QRC models is comparable to that of the DNN, particularly in the statevector (sv) setting, indicating that the reservoir dynamics are able to extract informative nonlinear features from the input data.

In addition to classification performance, it is important to evaluate the computational cost of the proposed approach. Table 4.12 reports the training times of the considered classifiers when applied to the QRC feature space, compared with the DNN baseline trained on the original features. With the exception of SVM, all reservoir-based classifiers exhibit significantly lower training times than the DNN, often by one or two

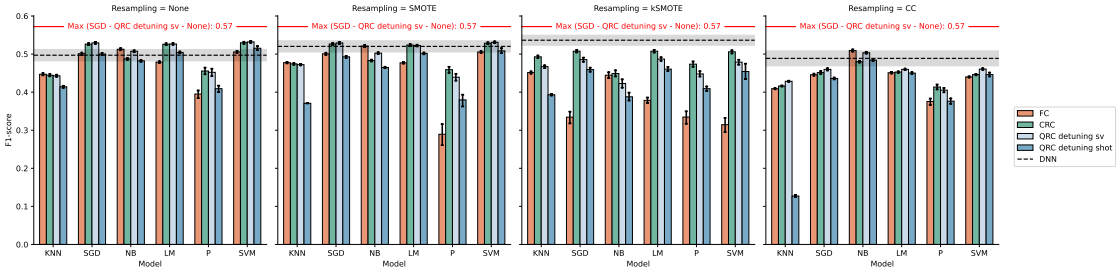


Figure 4.15: Classification benchmark on CARDS_30000, comparing emulated QRC (detuning encoding), CRC, and FC against the DNN baseline. Shaded area denotes the DNN standard deviation; the red line marks the peak performance across all classifiers.

orders of magnitude. This reflects one of the key advantages of QRC, where the computational burden is shifted from training to feature generation. In contrast, the SVM shows prohibitive training times due to its sensitivity to the high dimensionality of the reservoir feature space, even when using a linear kernel.

Table 4.12: Training time comparison (in seconds) across the evaluated classifiers on CARDS_30000 under various resampling strategies. The DNN is trained directly on the original 12 features, whereas the remaining classifiers operate on the feature set produced by the QRC (detuning encoding). Consistent with expectations, oversampling methods lead to longer training times, while undersampling reduces them.

Classifier	**None**	**SMOTE**	**kSMOTE**	**CC**
KNN	0.12 ± 0.01	0.23 ± 0.01	0.24 ± 0.11	0.10 ± 0.01
SGD	2.86 ± 0.34	3.43 ± 0.40	2.82 ± 0.31	1.22 ± 0.15
Naïve Bayes	0.11 ± 0.01	0.23 ± 0.01	0.20 ± 0.01	0.10 ± 0.01
Logistic Model	0.14 ± 0.01	0.31 ± 0.02	0.30 ± 0.10	0.08 ± 0.01
Perceptron	0.18 ± 0.02	0.28 ± 0.02	0.27 ± 0.01	0.09 ± 0.01
SVM	1515.93 ± 397.65	4359.86 ± 1076.96	3976.44 ± 1077.47	150.71 ± 83.96
DNN	26.52 ± 0.26	43.32 ± 2.77	42.97 ± 1.77	13.62 ± 0.62

Execution on Aquila QPU and Experimental Setup

After validating the methodology through classical emulation, the full pipeline is executed on the Aquila QPU using the reduced datasets CARDS_2500 and CARDS_1000. The use of smaller datasets is required due to current limitations in quantum resources and execution time. As in the emulation setting, each data sample corresponds to a single Hamiltonian evolution, and the total number of measurements is determined by the number of samples and the number of shots per sample. The objective of this phase is to assess the impact of hardware noise on both the learned feature representations and the resulting classification performance, while preserving a direct comparison with the noiseless baseline.

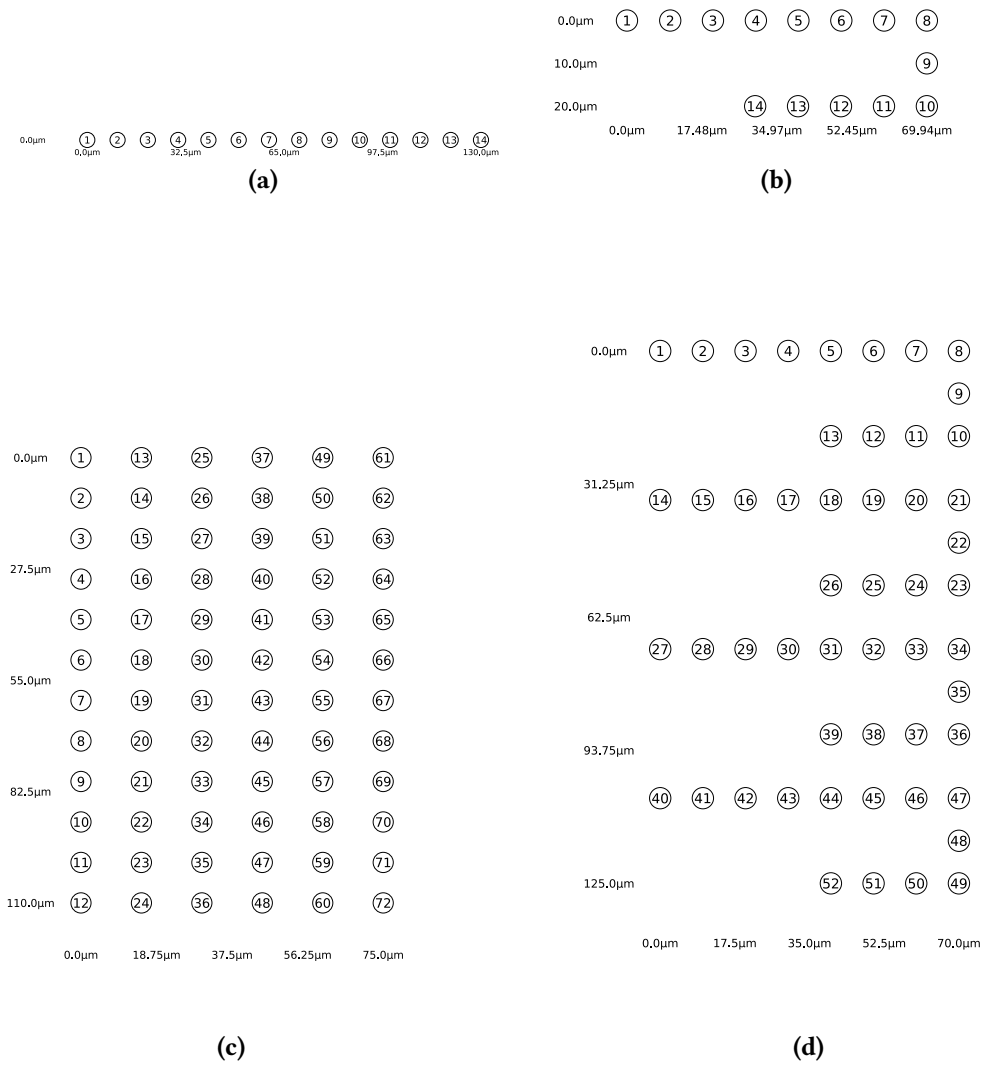


Figure 4.16: Comparison of register encodings: (a) column, (b) hook-shaped, (c) replicated column, (d) replicated hooks.

Use of Replicas for Sampling Efficiency. Experiments are conducted on the Aquila neutral-atom platform [29], which provides a two-dimensional register of size $75\ \mu m \times 128\ \mu m$. In contrast to the one-dimensional arrays used in emulation, the qubit layout must now satisfy hardware constraints, including a minimum inter-atomic spacing of $4\ \mu m$. To embed the logical atom array into the physical register, two strategies are considered: a column-wise layout when the array fits within the register height, and a serpentine (hook-shaped) embedding otherwise. In addition, multiple replicas of the same array can be placed within the register to improve the effective sampling rate. Examples of these configurations are reported in Fig. 4.16.

In the considered setup, both encoding strategies fit within the register height, allowing the use of a column-wise embedding with $N_{\text{repl}} = 6$ replicas (Fig. 4.16c). Replicas are separated by 15 μm, exceeding both the Rydberg interaction radius and the reference distance r_0, in order to suppress cross-replica interactions.

Noise Impact and Shot Optimization. A further distinction with respect to the emulation setting concerns the measurement process. Since non-destructive measurements are not available on the physical platform, the reservoir evolution must be restarted from the initial state $|\psi_{t=0}\rangle$ at each timestep. This results in progressively longer pulse sequences for later timesteps, which increases exposure to decoherence effects. To quantify the impact of quantum noise, we analyze the statistical correlation between QRC features obtained from Aquila and those obtained via shot-based emulation. This analysis is performed on a subset of 10 randomly selected samples from the CARDS_2500 dataset using detuning encoding. The results, shown in Fig. 4.17, indicate that correlation decreases for later timesteps, reflecting the accumulation of noise and decoherence. Moreover, increasing the number of shots beyond approximately 200 does not significantly improve the correlation. Based on this analysis, and considering $N_{\text{repl}} = 6$, we select a total number of measurements $n_{\text{opt}} = 240$, corresponding to 40 shots per timestep and per sample.

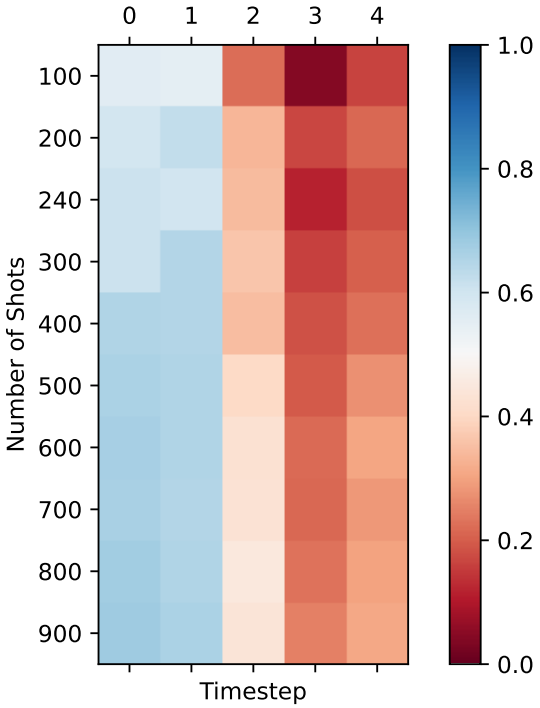


Figure 4.17: Pearson correlation between Aquila QRC outputs and shot-based emulation as a function of shot count and timestep.

Results on CARDS_1000**.** Both encoding strategies are first evaluated on the smallest CARDS_1000 dataset, which represents the most challenging setting due to its limited training set size. The classification results, reported in Fig. 4.18, reveal the expected performance degradation attributable to hardware noise: emulated results consistently outperform their QPU counterparts across all classifiers and resampling strategies.

Comparing the two encoding strategies, detuning encoding achieves equal or superior performance relative to position encoding in the majority of configurations. This discrepancy can be attributed to the differing sensitivity of the two encodings to hardware imperfections: position encoding is inherently more susceptible to shot-to-shot fluctuations in atom placement, which directly perturb the interaction graph underlying the reservoir dynamics. Local detuning amplitudes, by contrast, are controlled more stably by the laser fields, making detuning encoding more resilient to the positional noise characteristic of neutral-atom devices. This observation motivates the exclusive use of detuning encoding in the subsequent experiments on the larger dataset.

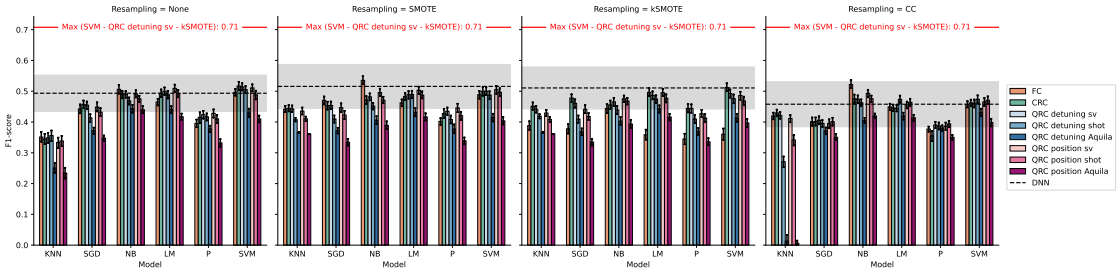


Figure 4.18: Classification benchmark on CARDS_1000, contrasting Aquila hardware and emulated QRC under detuning and position encoding, CRC, and FC against the DNN baseline.

Results on CARDS_2500**.** Building on the observations from the CARDS_1000 experiments, subsequent evaluations are conducted on the larger CARDS_2500 dataset using detuning encoding exclusively. The resulting QRC feature representations are compared against both statevector and shot-based emulation in Fig. 4.19. While the overall structure of the feature space is largely preserved across execution modes, deviations attributable to hardware noise are clearly visible, manifesting as a broadening and displacement of feature clusters relative to the noiseless reference.

A more granular perspective is provided in Fig. 4.20, where the QRC features for a single representative sample are tracked across timesteps. The influence of decoherence becomes increasingly apparent from intermediate timesteps onward, manifesting as a progressive divergence from the noiseless trajectory. This temporal degradation is consistent with the expected accumulation of coherence errors over the duration of the quantum evolution.

Despite the noise-induced feature degradation, the classification results reported in

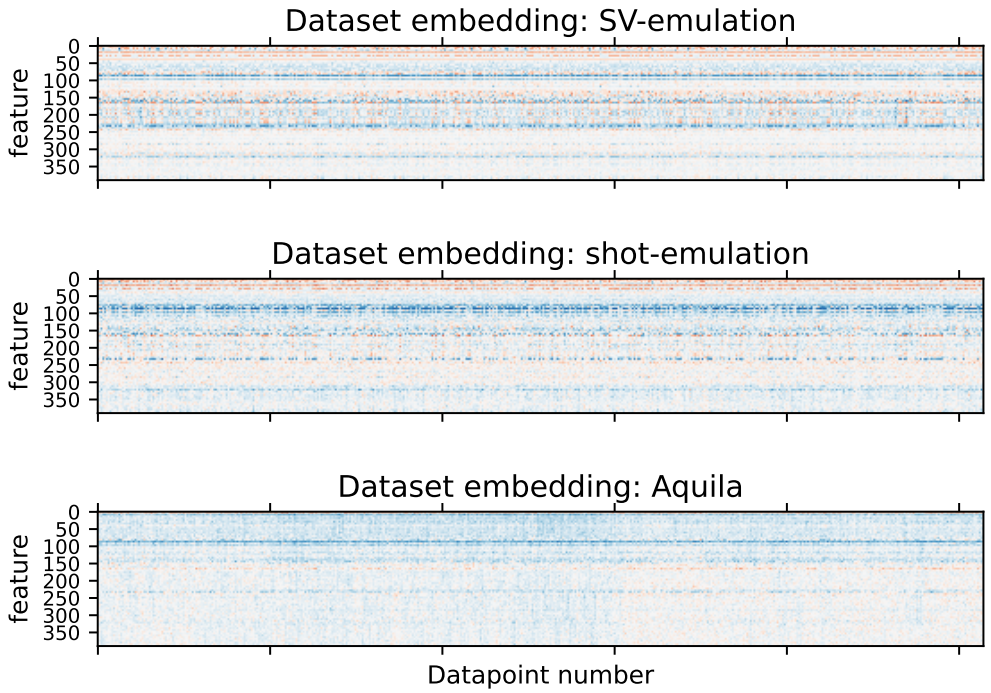


Figure 4.19: QRC feature space on CARDS_2500 under detuning encoding, contrasting Aquila hardware outputs with statevector (SV) and shot-based emulation.

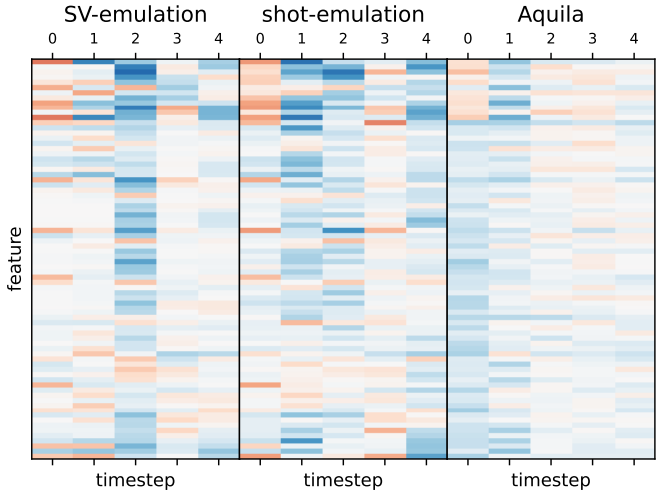


Figure 4.20: Timestep-resolved QRC features for a representative CARDS_2500 sample, illustrating the progressive divergence between Aquila hardware and noiseless emulation due to decoherence.

Fig. 4.21 demonstrate that hardware performance remains competitive across all metrics. Notably, the gap between QPU and emulation results is narrower on CARDS_2500

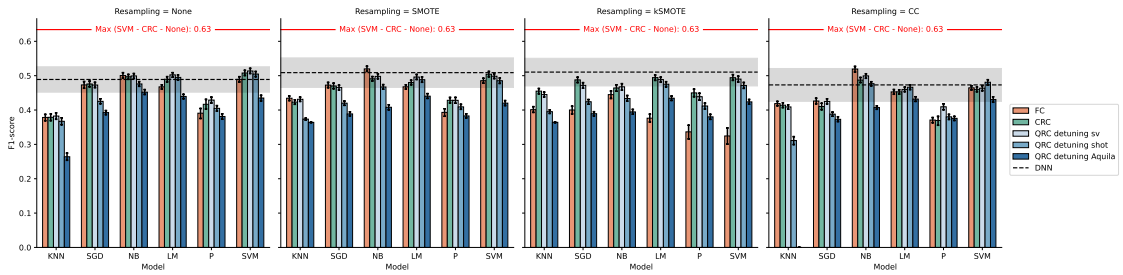


Figure 4.21: Classification benchmark on CARDS_2500, contrasting Aquila hardware and emulated QRC (detuning encoding), CRC, and FC against the DNN baseline.

than on CARDS_1000, suggesting that the increased number of training samples enables the classical classifiers to learn more robust decision boundaries that partially compensate for the variability introduced by quantum noise. This trend is consistent with the general expectation that noise robustness improves with dataset size, as the classifier has more signal from which to disentangle hardware-induced perturbations.

Taken together, these results confirm that the QRC framework can be effectively deployed on current neutral-atom hardware for real-world classification tasks. While hardware noise inevitably degrades performance relative to ideal emulation, the hybrid quantum-classical architecture demonstrates meaningful resilience: classical post-processing absorbs a significant portion of the noise impact, and performance degradation diminishes as the training set grows. These findings underscore the practical viability of NISQ-compatible hybrid architectures and motivate further investigation into noise mitigation strategies tailored to analog quantum reservoirs.

4.4 Discussion and Perspectives on Hybrid Quantum Machine Learning

The results presented in this chapter provide a comprehensive assessment of hybrid quantum machine learning approaches on neutral-atom platforms, spanning two distinct application domains: graph classification via Quantum Evolution Kernels and tabular data classification via Quantum Reservoir Computing. Despite the differences in data representation and learning paradigms, both studies rely on the same underlying analog quantum dynamics, highlighting the versatility of neutral-atom systems for quantum-enhanced learning.

A first key observation concerns the role of noise and its impact on model performance. While classical emulation provides a controlled, noiseless baseline, the behavior observed on the QPU is not uniformly degraded across the two applications. In the case of the QEK, execution on Aquila achieves performance comparable to, and in some cases slightly exceeding, the emulated results, with an F1-score of 65.6% on

the PROTEINS256 dataset, comparable to the classical Shortest Path Kernel. This suggests that noise does not necessarily hinder the kernel-based approach and may even introduce beneficial stochastic effects. In contrast, for the QRC application, hardware noise leads to a more noticeable degradation in performance with respect to emulation. Nevertheless, in both settings the relative trends across models and configurations are preserved, indicating that the underlying quantum feature representations remain informative even under realistic NISQ conditions. Overall, these results highlight the robustness of hybrid approaches, where learning is delegated to classical models that can partially compensate for imperfections in the quantum evolution.

A second important aspect concerns the encoding of classical information into the quantum system. In the graph-based setting, the encoding is structural, as the problem instance directly defines the interaction graph of the Hamiltonian. In contrast, the QRC application relies on explicit encoding strategies, such as local detuning modulation and position-based perturbations. The comparison between these approaches reveals that encoding plays a crucial role in determining both stability and performance: while position encoding offers a more direct mapping to physical interactions, it is also more sensitive to hardware imperfections, whereas detuning-based encoding provides a more controlled and robust alternative. This highlights a general design principle for QML on analog devices: encoding strategies must balance expressivity with hardware compatibility.

From a scalability perspective, both applications demonstrate that relevant quantum-enhanced learning tasks can already be executed on current neutral-atom devices. The ability to manipulate hundreds of qubits, combined with flexible interaction patterns, enables the exploration of problem instances beyond the reach of exact classical simulation. In the QEK case, this allows the processing of graph datasets with non-trivial connectivity, while in the QRC framework, the use of replicas enables the efficient extraction of high-dimensional feature spaces. Nevertheless, practical limitations remain, including limited coherence times, restricted measurement capabilities, and the time cost associated with large numbers of shots. These constraints necessitate careful design choices, such as limiting circuit duration, optimizing the number of measurements, and exploiting parallelism through register replication.

Another relevant outcome concerns the role of hybridization in mitigating current hardware limitations. In both applications, the quantum device is not trained directly; instead, it acts as a feature generator, while the learning task is carried out by classical components of the pipeline. This separation proves advantageous, as it allows classical methods to compensate for noise and extract meaningful patterns from imperfect quantum data. In the QRC application, for instance, increasing the size of the training dataset improves the robustness of the learned models. The classical resampling techniques steps operate on the quantum-generated features and help mitigate the impact of noise introduced during quantum evolution. These observations reinforce the suitability of hybrid quantum-classical paradigms in the NISQ era, where carefully designed classical pipelines play a crucial role in unlocking the practical utility of quantum-generated

representations.

Overall, the results presented in this chapter demonstrate that neutral-atom quantum simulators already provide a viable platform for exploring practical QML workflows. While no clear quantum advantage is observed at this stage, both the QEK and QRC approaches achieve performance comparable to classical baselines, while offering distinct modeling capabilities and alternative representations of the data. As hardware continues to improve in terms of coherence, control precision, and system size, these methods are expected to benefit from enhanced stability and expressivity.

Chapter 5

Conclusion

This thesis has investigated how quantum computing can be brought closer to practical and industrial applications in the NISQ (Noisy Intermediate-Scale Quantum) era, with a particular focus on neutral-atom quantum processors operating in the analog regime. Rather than targeting theoretical quantum advantage under idealized assumptions, the work has addressed the full pipeline required to translate real-world problems into executable quantum workflows, from hardware-aware embedding, through hybrid combinatorial optimization, to quantum-enhanced machine learning. By integrating these three research directions around shared computational primitives, the thesis provides a coherent and practically grounded framework for leveraging current quantum hardware despite its limitations.

5.1 Summary of Contributions

The first contribution of this thesis concerns the development of *embedding methodologies* for neutral-atom quantum processors. Embedding is a critical and often underestimated step in quantum computing: it defines how a problem is translated into the physical degrees of freedom of the hardware, and its quality directly conditions the performance of any downstream algorithm. In the neutral-atom setting, this task is particularly challenging due to geometric constraints imposed by the Rydberg blockade mechanism, device-specific register geometries, and distance-dependent interaction patterns. The proposed DEN (Distance Encoder Network) framework addresses these challenges through a neural optimization approach that learns feasible atom configurations for a wide range of graph instances. Adaptations to the Orion Alpha and Aquila QPUs (Quantum Processing Units) — including discrete lattice projection and row-spacing constraints — demonstrate that hardware adaptation must be treated as an integral component of the embedding procedure rather than a final-stage correction. The results establish that the DEN achieves significantly higher feasibility rates than classical exact solvers, at a fraction of the computational cost.

Building on this foundation, the thesis has developed hybrid quantum-classical approaches to combinatorial optimization. Three progressively more sophisticated methods have been presented for the vertex coloring problem, which is directly connected to industrially relevant applications such as PCI (Principal Cell Identifier) assignment in cellular networks. The *Greedy-it-MIS* baseline establishes the core connection between independent sets and valid colorings; the BBQ-mIS algorithm introduces a BB (Branch-and-Bound) framework that exploits the distribution of quantum measurements to explore multiple candidate color classes; and finally, the QCBP (Quantum-Classical Branch-and-Price) framework integrates quantum-assisted column generation within a full BP (Branch-and-Price) scheme, achieving near-optimal solutions on 97.9% of benchmark instances while substantially reducing QPU usage compared to BBQ-mIS. A second optimization application demonstrates that the same quantum kernel, based on QAA (Quantum Adiabatic Algorithm) to solve MWIS (Maximum-Weighted Independent Set) problems, can be reused within a different pipeline, addressing satellite-terrestrial network design problems within the QRISTIN project through MWIS-alike quantum routines on the Aquila QPU. Together, these contributions validate the hybrid decomposition paradigm as a practical and scalable strategy for combinatorial optimization on NISQ devices.

The third contribution concerns *quantum machine learning* on neutral-atom platforms. Two complementary case studies are presented, both of which leverage analog quantum dynamics as a feature generator within a hybrid pipeline. In the first, QEKs (Quantum Evolution Kernels) are used to classify protein graphs from the PROTEINS dataset: pulse parameters are optimized via Bayesian optimization on classical emulation, and the resulting configurations are transferred to the Aquila QPU, where an F1-score of 65.6% is achieved on the PROTEINS256 dataset, matching the classical Shortest Path Kernel. In the second, a QRC (Quantum Reservoir Computing) framework is applied to credit card default prediction using tabular data, demonstrating that detuning-encoded quantum reservoirs can produce feature representations competitive with a DNN (Deep Neural Network) baseline, with significantly shorter training times.

5.2 Embedding as a Key Part of Quantum Pipelines

A recurring theme across all three research directions is that embedding is not a preliminary technical step, but a first-class component of the quantum computational pipeline. In classical computing, data representation is largely decoupled from hardware; in quantum computing, and particularly on analog neutral-atom devices, the two are inseparable. The spatial arrangement of atoms defines the effective Hamiltonian, which in turn shapes the solution landscape explored by the quantum evolution.

This tight coupling between geometry and computation has several practical implications. First, the quality of the embedding conditions the quality of any downstream result: poor atom placement leads to missing or spurious interactions that corrupt the

intended problem structure. Second, not all problems admit a valid embedding under realistic hardware constraints, making feasibility itself a non-trivial design objective. Third, the same problem can often be reformulated in multiple ways, as illustrated by the shift from the QUBO (Quadratic Unconstrained Binary Optimization) formulation of vertex coloring to MWIS-based decomposition, with significant implications for embedding complexity and hardware compatibility.

More broadly, the results of this thesis support the view that embedding and algorithm design should be developed jointly. The choice of encoding influences the structure of the solution space, the behavior of the quantum evolution, and the effectiveness of classical post-processing. In the QEK application, for instance, the adjacency structure of the protein graph directly shapes the Ising Hamiltonian and the discriminative capacity of the resulting kernel. In the QRC setting, the choice between position and detuning encoding determines both expressivity and robustness to noise. These observations motivate a design principle that runs throughout the thesis: hardware-aware algorithm design is not a constraint to work around, but a source of structure that can be actively exploited.

5.3 Quantum-Classical Hybridization as a Paradigm

A second cross-cutting insight is that purely quantum approaches are insufficient in the current NISQ regime, and that the most effective strategies are those that carefully distribute computation between quantum and classical components. In the optimization setting, quantum routines act as pricing oracles or sampling engines within classical decomposition frameworks, contributing candidate solutions that accelerate search without bearing the full burden of the computation. In the machine learning setting, the quantum device generates feature representations through its intrinsic dynamics, while learning is entirely delegated to classical models.

This separation of roles proves advantageous for several reasons. Classical algorithms are robust to noise and can compensate for imperfections in the quantum output; they can handle global consistency constraints that are difficult to encode physically; and they provide convergence guarantees that the quantum component alone cannot offer. At the same time, the quantum component contributes properties that are difficult to replicate classically: the ability to explore large combinatorial neighborhoods through coherent sampling, the generation of high-dimensional feature spaces through many-body dynamics, and the potential for favorable asymptotic scaling as hardware improves.

The QCBP framework provides a clear illustration of this balance. By replacing QAOA (Quantum Approximate Optimization Algorithm) with QAA and reusing the independent sets generated during column generation for the BB branching phase, the algorithm achieves a threefold benefit: it reduces QPU calls, improves lower bounds

through the relaxed RMP (Reduced Master Problem), and maintains near-optimal solution quality. The result is a framework that extracts more computational value per quantum evaluation than its predecessors, by ensuring that each QPU call contributes simultaneously to multiple components of the optimization.

5.4 Limitations and Open Challenges

Despite the promising results, several limitations must be acknowledged. Hardware noise and limited coherence times remain the primary constraints on achievable performance. While hybrid approaches partially mitigate these effects, as demonstrated by the robustness of the QCBP framework to non-unit-disk embeddings and by the resilience of QRC classifiers to decoherence, noise still imposes a ceiling on the quality and reliability of quantum-generated outputs, particularly for longer evolutions or larger systems.

Embedding remains a significant bottleneck. The DEN model achieves high feasibility rates but cannot guarantee convergence, and its success rate decreases as graph size and structural complexity increase. For instances with hundreds of vertices approaching the device limit, feasibility rates drop substantially, constraining the range of problems that can be addressed without approximation. A deeper characterization of which graph classes admit feasible embeddings under realistic hardware constraints remains an important open problem. More broadly, embedding, together with the pulse design stage, in analog quantum computing plays the role that compilation and transpilation play in gate-based architectures: it is the layer that translates an abstract problem description into a physically executable configuration. Unlike gate-based systems, however, analog platforms do not benefit from a unified and automated compilation stack. Each new device, with its own register geometry, interaction radius, and control constraints, currently requires problem-specific adaptation, placing a significant burden on the algorithm designer and limiting the portability of quantum solutions across hardware generations and providers. Developing systematic and hardware-agnostic embedding and compilation tools for analog quantum architectures therefore represents a key open challenge, and one whose resolution would substantially lower the barrier to the broader adoption of quantum computing.

Scalability is further limited by the cost of quantum execution. At current sampling rates of approximately $1–5\,\mathrm{Hz}$ on available neutral-atom devices, the total execution time for a single optimization run or machine learning pipeline can amount to several hours. While the present work is primarily a proof of principle rather than a demonstration of computational advantage, these resource costs must be addressed before hybrid quantum methods can be practically deployed at scale.

Finally, assessing quantum utility remains a difficult and multi-dimensional question. The results presented in this thesis show competitive performance with classical methods, but not systematic superiority. Establishing conditions under which

quantum-enhanced methods provide a reliable practical advantage will require more comprehensive benchmarking studies, larger hardware experiments, and a clearer theoretical understanding of the regimes in which quantum feature maps or quantum samplers outperform their classical counterparts.

5.5 Future Directions

The results of this thesis suggest several concrete directions for future research.

On the *embedding* side, a systematic investigation of which graph classes admit feasible constrained unit-disk embeddings would clarify the scope of applicability of analog quantum methods. From an algorithmic perspective, improving the adjacency-gap maximization within the ELF (Embedding Loss Function) minimization, introducing convergence guarantees, and exploring active learning strategies that select embedding parameters based on downstream task performance are all promising directions. Device-aware co-design, in which embedding strategies are jointly optimized with the target QPU architecture rather than adapted post hoc, is expected to become increasingly important as hardware diversity grows.

A closely related priority concerns the development of systematic *compilation and mapping tools* for analog quantum architectures. In the gate-based quantum computing ecosystem, significant progress has been made in automated transpilation, circuit optimization, and hardware-aware compilation, enabling researchers to target different devices without redesigning their algorithms from scratch. Analog platforms currently lack an equivalent abstraction layer. Developing compiler toolchains that automate the translation from high-level problem descriptions to device-compatible pulse sequences and atom configurations, while respecting hardware constraints and optimizing for noise robustness, would substantially lower the barrier to entry for new users and accelerate the deployment of quantum algorithms across heterogeneous hardware. The embedding methodology developed in this thesis constitutes a first step in this direction, but a complete compilation stack would additionally need to handle pulse optimization, error mitigation, and cross-device portability in an integrated and automated manner.

On the *optimization* side, the QCBP framework provides a sound foundation for extension. Incorporating more expressive branching strategies, adaptive column generation, stopping criteria, and noise-robust quantum subroutines could further improve performance on non-unit-disk instances. Scaling the approach to larger graphs using multi-level decomposition or problem-specific warm-starting heuristics would expand its applicability to industrial-scale instances. The transfer of the MWIS quantum kernel to other graph-structured optimization problems beyond vertex coloring, such as scheduling, facility location, or network flow, is a natural next step supported by the generality of the QRISTIN experiments.

On the *machine learning* side, the most immediate priority is a systematic investigation of noise mitigation strategies tailored to analog quantum reservoirs. Error mitigation techniques such as zero-noise extrapolation, probabilistic error cancellation, and readout correction could substantially improve the fidelity of quantum-generated features. Beyond noise, developing principled methods for pulse parameter optimization, moving beyond Bayesian optimization on small subsets to more scalable and theoretically grounded approaches, would strengthen the generalization of optimized configurations across graph sizes and data distributions. The broader question of which problem structures and data types are best suited to QEK and QRC frameworks remains largely open, and systematic empirical studies on diverse benchmark tasks would significantly advance the understanding of their capabilities and limitations.

More broadly, the continued improvement of neutral-atom hardware (in terms of qubit counts, coherence times, control fidelity, and sampling rates) is expected to directly expand the range of feasible applications. The emergence of mid-circuit measurement capabilities, reconfigurable atom arrays, and optical tweezer systems with three-dimensional control will open qualitatively new possibilities for both optimization and learning tasks. Tracking these developments and adapting the methodological framework accordingly will be an ongoing research program.

5.6 Closing Remarks

This thesis demonstrates that meaningful progress toward practical quantum computing can be achieved by grounding algorithm design in the physical reality of available hardware. The combination of hardware-aware embedding, principled hybrid decomposition, and quantum-assisted feature generation provides a coherent and implementable framework for addressing real-world computational challenges with NISQ devices.

The neutral-atom platform, with its programmable geometric interactions and analog computational model, emerges from this work as a versatile substrate that supports a unified computational primitive across combinatorial optimization, graph classification, and tabular machine learning. This unifying thread is not coincidental: it reflects the fundamental alignment between the physical interactions of Rydberg atoms and the structural properties of a broad class of hard combinatorial and learning problems.

While a definitive demonstration of quantum advantage remains a goal for future work, the results presented here confirm that quantum-classical hybrid architectures, carefully designed around the constraints and capabilities of available hardware, are already capable of producing practically relevant computational outcomes. This positions neutral-atom quantum computing as a credible and increasingly capable component of future hybrid computational infrastructures.

Bibliography

[1] Iulia Buluta and Franco Nori. "Quantum simulators." In: *Science* 326.5949 (2009), pp. 108–111.

[2] Abhinav Kandala et al. "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets." In: *Nature* 549.7671 (2017), pp. 242–246.

[3] Sam McArdle et al. "Quantum computational chemistry." In: *Reviews of Modern Physics* 92.1 (2020), p. 015003.

[4] Gary Kochenberger et al. "The unconstrained binary quadratic programming problem: a survey." In: *Journal of combinatorial optimization* 28.1 (2014), pp. 58–81.

[5] Tadashi Kadowaki and Hidetoshi Nishimori. "Quantum Annealing in the Transverse Ising Model." In: *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics* 58.5 (Apr. 1998), pp. 5355–5363. DOI: 10.1103/PhysRevE.58.5355. arXiv: 9804280 [cond-mat]. URL: http://arxiv.org/abs/cond-mat/9804280%20http://dx.doi.org/10.1103/PhysRevE.58.5355.

[6] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. "A quantum approximate optimization algorithm." In: *arXiv preprint arXiv:1411.4028* (2014).

[7] Haimeng Zhao et al. "Exponential quantum advantage in processing massive classical data." In: *arXiv preprint arXiv:2604.07639* (2026).

[8] Hsin-Yuan Huang, Richard Kueng, and John Preskill. "Information-theoretic bounds on quantum advantage in machine learning." In: *Physical Review Letters* 126.19 (2021), p. 190505.

[9] John Preskill. "Quantum computing in the NISQ era and beyond." In: *Quantum* 2 (2018), p. 79.

[10] Kishor Bharti et al. "Noisy intermediate-scale quantum algorithms." In: *Reviews of Modern Physics* 94.1 (2022), p. 015004.

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

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

[13] Jarrod R McClean et al. "The theory of variational hybrid quantum-classical algorithms." In: *New Journal of Physics* 18.2 (2016), p. 023023.

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

[15] Samuel Mugel et al. "Hybrid quantum investment optimization with minimal holding period." In: *Scientific Reports* 11.1 (2021), p. 19587.

[16] Ken M Nakanishi, Keisuke Fujii, and Synge Todo. "Sequential minimal optimization for quantum-classical hybrid algorithms." In: *Physical Review Research* 2.4 (2020), p. 043158.

[17] TM Graham et al. "Multi-qubit entanglement and algorithms on a neutral-atom quantum computer." In: *Nature* 604.7906 (2022), pp. 457–462.

[18] Ming-Wei Li et al. "A ship motion forecasting approach based on empirical mode decomposition method hybrid deep learning network and quantum butterfly optimization algorithm." In: *Nonlinear dynamics* 107.3 (2022), pp. 2447–2467.

[19] Robert de Keijzer, Oliver Tse, and Servaas Kokkelmans. "Pulse based Variational Quantum Optimal Control for hybrid quantum computing." In: *Quantum* 7 (Jan. 2023), p. 908. ISSN: 2521-327X. DOI: 10.22331/q-2023-01-26-908. URL: https://doi.org/10.22331/q-2023-01-26-908.

[20] Morten Kjaergaard et al. "Superconducting qubits: Current state of play." In: *Annual Review of Condensed Matter Physics* 11.1 (2020), pp. 369–395.

[21] Colin D Bruzewicz et al. "Trapped-ion quantum computing: Progress and challenges." In: *Applied physics reviews* 6.2 (2019).

[22] Mark Saffman, Thad G Walker, and Klaus Mølmer. "Quantum information with Rydberg atoms." In: *Reviews of modern physics* 82.3 (2010), pp. 2313–2363.

[23] Antoine Browaeys and Thierry Lahaye. "Many-body physics with individually controlled Rydberg atoms." In: *Nature Physics* 16.2 (2020), pp. 132–142.

[24] Sergei Slussarenko and Geoff J Pryde. "Photonic quantum information processing: A concise review." In: *Applied physics reviews* 6.4 (2019).

[25] Floris A Zwanenburg et al. "Silicon quantum electronics." In: *Reviews of modern physics* 85.3 (2013), pp. 961–1019.

[26] Mark W Johnson et al. "Quantum annealing with manufactured spins." In: *Nature* 473.7346 (2011), pp. 194–198.

[27] Thomas F Gallagher. "Rydberg atoms." In: *Reports on Progress in Physics* 51.2 (1988), p. 143.

[28] Loïc Henriet et al. "Quantum Computing with Neutral Atoms." In: *Quantum* 4 (Sept. 2020), p. 327. ISSN: 2521-327X. DOI: 10.22331/q-2020-09-21-327.

[29] Jonathan Wurtz et al. "Aquila: QuEra's 256-qubit neutral-atom quantum computer." In: *arXiv preprint arXiv:2306.11727* (2023).

[30] D. Jaksch et al. "Fast Quantum Gates for Neutral Atoms." In: *Physical Review Letters* 85.10 (Sept. 2000), pp. 2208–2211. ISSN: 1079-7114. DOI: 10.1103/physrevlett.85.2208. URL: http://dx.doi.org/10.1103/PhysRevLett.85.2208.

[31] Alpha Gaëtan et al. "Observation of collective excitation of two individual atoms in the Rydberg blockade regime." In: *Nature Physics* 5.2 (Jan. 2009), pp. 115–118. ISSN: 1745-2481. DOI: 10.1038/nphys1183. URL: http://dx.doi.org/10.1038/nphys1183.

[32] Andrew Lucas. "Ising Formulations of Many NP Problems." In: *Frontiers in Physics* 2 (2014). ISSN: 2296-424X. DOI: 10.3389/fphy.2014.00005.

[33] Fred Glover, Gary Kochenberger, and Yu Du. "A tutorial on formulating and using QUBO models." In: *arXiv preprint arXiv:1811.11538* (2018).

[34] M. Morgado and S. Whitlock. "Quantum simulation and computing with Rydberg-interacting qubits." In: *AVS Quantum Science* 3.2 (May 2021). ISSN: 2639-0213. DOI: 10.1116/5.0036562. URL: http://dx.doi.org/10.1116/5.0036562.

[35] Chiara Vercellino et al. "Neural-powered unit disk graph embedding: qubits connectivity for some qubo problems." In: *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE. 2022, pp. 186–196.

[36] Chiara Vercellino et al. "Neural optimization for quantum architectures: graph embedding problems with Distance Encoder Networks." In: *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE. 2023, pp. 380–389.

[37] Chiara Vercellino et al. "Harnessing DEN models for quantum computing tasks on neutral atom QPUs." In: *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 2. IEEE. 2024, pp. 217–222.

[38] Giacomo Vitali et al. *Towards Optimal Graph Coloring Using Rydberg Atoms*. 2021.

[39] Chiara Vercellino et al. "BBQ-mIS: a parallel quantum algorithm for graph coloring problems." In: *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 2. IEEE. 2023, pp. 141–147.

[40] Chiara Vercellino et al. "Hybrid Quantum-Classical Branch-and-Price Method for the Vertex Coloring Problem." In: *arXiv preprint arXiv:2508.18887* (2025).

[41] Chiara Vercellino et al. "Quantum-Assisted Design of Space-Terrestrial Integrated Networks." In: *arXiv preprint arXiv:2602.04350* (2026).

[42] Edoardo Giusto et al. "Harnessing a 256-qubit Neutral Atom Simulator for Graph Classification." In: *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 1. IEEE. 2024, pp. 296–305.

[43] Giacomo Vitali et al. "Quantum Reservoir Computing for Credit Card Default Prediction on a Neutral Atom Platform." In: *arXiv preprint arXiv:2510.04747* (2025).

[44] Hannes Pichler et al. "Quantum Optimization for Maximum Independent Set Using Rydberg Atom Arrays." In: *Bulletin of the American Physical Society* 64 (2019).

[45] T. Wilk et al. "Entanglement of Two Individual Neutral Atoms Using Rydberg Blockade." In: *Phys. Rev. Lett.* 104 (1 Jan. 2010), p. 010502. DOI: 10.1103/PhysRevLett.104.010502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.104.010502.

[46] Brent N. Clark, Charles J. Colbourn, and David S. Johnson. "Unit disk graphs." In: *Discrete Mathematics* 86.1 (1990), pp. 165–177. ISSN: 0012-365X. DOI: https://doi.org/10.1016/0012-365X(90)90358-O. URL: https://www.sciencedirect.com/science/article/pii/0012365X90903580.

[47] Martin Charles Golumbic. *Algorithmic graph theory and perfect graphs*. Elsevier, 2004.

[48] Pasqal. *Send a job to Fresnel*. https://docs.pasqal.com/cloud/fresnel-job/. Pasqal Documentation (Cloud documentation), Last updated: June 30, 2025. Accessed: August 14, 2025. 2025.

[49] Wolfgang Lechner, Philipp Hauke, and Peter Zoller. "A quantum annealing architecture with all-to-all connectivity from local interactions." In: *Science advances* 1.9 (2015), e1500838.

[50] Xingze Qiu, Peter Zoller, and Xiaopeng Li. "Programmable quantum annealing architectures with Ising quantum wires." In: *PRX Quantum* 1.2 (2020), p. 020311.

[51] Minhyuk Kim et al. "Rydberg Quantum Wires for Maximum Independent Set Problems with Nonplanar and High-Degree Graphs." In: *arXiv:2109.03517 [physics, physics:quant-ph]* (Sept. 2021). arXiv: 2109.03517 [physics, physics:quant-ph].

[52] Neil Robertson and Paul D Seymour. "Graph minors. XIII. The disjoint paths problem." In: *Journal of combinatorial theory, Series B* 63.1 (1995), pp. 65–110.

[53] Jun Cai, William G. Macready, and Aidan Roy. "A Practical Heuristic for Finding Graph Minors." In: *arXiv:1406.2741 [quant-ph]* (June 2014). arXiv: 1406.2741 [quant-ph].

[54] Vicky Choi. "Minor-embedding in adiabatic quantum computation: I. The parameter setting problem." In: *Quantum Information Processing* 7.5 (2008), pp. 193–209.

[55] Henrique Silvério et al. "Pulser: An Open-Source Package for the Design of Pulse Sequences in Programmable Neutral-Atom Arrays." In: *arXiv:2104.15044 [quant-ph]* (Jan. 2022). arXiv: 2104.15044 [quant-ph].

[56] Donatella Ciampini, Oliver Morsch, and Ennio Arimondo. "Ultracold Rubidium Atoms Excited to Rydberg Levels." In: *Journal of Atomic, Molecular, Condensed Matter and Nano Physics* 2.3 (2015), pp. 161–167.

[57] CJ Picken et al. "Entanglement of neutral-atom qubits with long ground-Rydberg coherence times." In: *Quantum Science and Technology* 4.1 (2018), p. 015011.

[58] Hai-Chau Chang and Lih-Chung Wang. "A simple proof of thue's theorem on circle packing." In: *arXiv preprint arXiv:1009.4322* (2010).

[59] Heinz Breu and David G. Kirkpatrick. "Unit Disk Graph Recognition Is NP-hard." In: *Computational Geometry* 9.1-2 (Jan. 1998), pp. 3–24. ISSN: 09257721. DOI: 10.1016/S0925-7721(97)00014-X.

[60] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. "Unit Disk Graph Approximation." In: *Proceedings of the 2004 Joint Workshop on Foundations of Mobile Computing - DIALM-POMC '04*. Philadelphia, PA, USA: ACM Press, 2004, p. 17. ISBN: 978-1-58113-921-1. DOI: 10.1145/1022630.1022634.

[61] Fabian Kuhn, Rogert Wattenhofer, and Aaron Zollinger. "Ad-hoc networks beyond unit disk graphs." In: *Proceedings of the 2003 joint workshop on Foundations of mobile computing*. 2003, pp. 69–78.

[62] Sujoy Bhore et al. "Unit Disk Representations of Embedded Trees, Outerplanar and Multi-Legged Graphs." In: *arXiv:2103.08416 [cs]* (Aug. 2021). arXiv: 2103.08416 [cs].

[63] Maciej M Sysło. "Characterizations of outerplanar graphs." In: *Discrete Mathematics* 26.1 (1979), pp. 47–53.

[64] Douglas Brent West et al. *Introduction to graph theory*. Vol. 2. Prentice hall Upper Saddle River, 2001.

[65] Thomas Moscibroda et al. "Virtual coordinates for ad hoc and sensor networks." In: *Proceedings of the 2004 joint workshop on Foundations of mobile computing*. 2004, pp. 8–16.

[66] Sriram Pemmaraju and Imran Pirwani. "Good quality virtual realization of unit disk graphs." In: *Journal of Computational Geometry (Old Web Site)* 2.1 (2011), pp. 69–91.

[67] T Rusterholz. *On the Approximation on Unit Disk Graph Coordinates*. Tech. rep. Tech. rep, 2003.

[68] Jiangwei Zhou et al. "Practical virtual coordinates for large wireless sensor networks." In: *The 18th IEEE International Conference on Network Protocols*. IEEE. 2010, pp. 41–51.

[69] Constantin Dalyac et al. "Qualifying quantum approaches for hard industrial optimization problems. A case study in the field of smart-charging of electric vehicles." In: *EPJ Quantum Technology* 8.1 (May 2021). ISSN: 2196-0763. DOI: 10.1140/epjqt/s40507-021-00100-3. URL: http://dx.doi.org/10.1140/epjqt/s40507-021-00100-3.

[70] Bob Bixby. "The gurobi optimizer." In: *Transp. Re-search Part B* 41.2 (2007), pp. 159–178.

[71] Andreas Wächter and Lorenz T Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming." In: *Mathematical programming* 106.1 (2006), pp. 25–57.

[72] Matteo Fischetti and Jason Jo. "Deep neural networks and mixed integer linear optimization." In: *Constraints* 23.3 (2018), pp. 296–309.

[73] Brandon Amos and J Zico Kolter. "Optnet: Differentiable optimization as a layer in neural networks." In: *International Conference on Machine Learning*. PMLR. 2017, pp. 136–145.

[74] Aaditya Chandrasekhar and Krishnan Suresh. "TOuNN: topology optimization using neural networks." In: *Structural and Multidisciplinary Optimization* 63.3 (2021), pp. 1135–1149.

[75] Aaditya Chandrasekhar and Krishnan Suresh. "Multi-material topology optimization using neural networks." In: *Computer-Aided Design* 136 (2021), p. 103017.

[76] Leo Ardon. "Reinforcement Learning to Solve NP-hard Problems: an Application to the CVRP." In: *arXiv preprint arXiv:2201.05393* (2022).

[77] Kenshin Abe et al. "Solving np-hard problems on graphs with extended alphago zero." In: *arXiv preprint arXiv:1905.11623* (2019).

[78] Evan McCarty et al. "NN-Baker: A Neural-network Infused Algorithmic Framework for Optimization Problems on Geometric Intersection Graphs." In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 23023–23035.

[79] Thomas MJ Fruchterman and Edward M Reingold. "Graph drawing by force-directed placement." In: *Software: Practice and experience* 21.11 (1991), pp. 1129–1164.

[80] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX." In: *Proceedings of the 7th Python in Science Conference*. Ed. by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, 2008, pp. 11–15.

[81] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization." In: *International Conference on Learning Representations*.

[82] GNU Project. *GNU Linear Programming Kit, Version 5.0*. Accessed: 2025-05-02. Free Software Foundation. 2020. URL: https://www.gnu.org/software/glpk/.

[83] COIN-OR. *Ipopt: Interior Point Optimizer*. https://github.com/coin-or/Ipopt. Accessed: 2025. 2025.

[84] Ravi Boppana and Magnús M Halldórsson. "Approximating maximum independent sets by excluding subgraphs." In: *BIT Numerical Mathematics* 32.2 (1992), pp. 180–196.

[85] Karsten M Borgwardt et al. "Protein function prediction via graph kernels." In: *Bioinformatics* 21.suppl_1 (2005), pp. i47–i56.

[86] Paul D Dobson and Andrew J Doig. "Distinguishing enzyme structures from non-enzymes without alignments." In: *Journal of molecular biology* 330.4 (2003), pp. 771–783.

[87] E Urban et al. "Observation of Rydberg blockade between two atoms." In: *Nature Physics* 5.2 (2009), pp. 110–114.

[88] H Paul Williams. *Model building in mathematical programming*. John Wiley & Sons, 2013.

[89] Fred Glover, Gary Kochenberger, and Yu Du. "Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models." In: *4OR* 17.4 (2019), pp. 335–371.

[90] Constantin Dalyac et al. "Graph algorithms with neutral atom quantum processors." In: *The European Physical Journal A* 60.9 (2024), pp. 1–15.

[91] M Yassine Naghmouchi and Wesley da Silva Coelho. "Mixed-integer linear programming solver using Benders decomposition assisted by a neutral-atom quantum processor." In: *Physical Review A* 110.1 (2024), p. 012434.

[92] Wesley da Silva Coelho, Loic Henriet, and Louis-Paul Henry. "Quantum pricing-based column-generation framework for hard combinatorial problems." In: *Physical Review A* 107.3 (2023), p. 032426.

[93] Claudio Sanavio, Edoardo Tignone, and Elisa Ercolessi. "Hybrid classical–quantum branch-and-bound algorithm for solving integer linear problems." In: *Entropy* 26.4 (2024), p. 345.

[94] J BnnoBRs. "Partitioning procedures for solving mixed-variables programming problems." In: *Numer. Math* 4.1 (1962), pp. 238–252.

[95] Zhongqi Zhao, Lei Fan, and Zhu Han. "Hybrid quantum benders' decomposition for mixed-integer linear programming." In: *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2022, pp. 2536–2540.

[96] Fang Gao et al. "Hybrid Quantum-Classical General Benders Decomposition Algorithm for Unit Commitment with Multiple Networked Microgrids." In: *arXiv preprint arXiv:2210.06678* (2022).

[97] Chin-Yao Chang et al. In: *arXiv preprint arXiv:2010.07852* (2020).

[98] Anuj Mehrotra and Michael A Trick. "A column generation approach for graph coloring." In: *informs Journal on Computing* 8.4 (1996), pp. 344–354.

[99] Barış Yıldız, Okan Arslan, and Oya Ekin Karaşan. "A branch and price approach for routing and refueling station location model." In: *European Journal of Operational Research* 248.3 (2016), pp. 815–826.

[100] Mauro Dell'Amico, Giovanni Righini, and Matteo Salani. "A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection." In: *Transportation science* 40.2 (2006), pp. 235–247.

[101] Jonathan F Bard and Narameth Nananukul. "A branch-and-price algorithm for an integrated production and inventory routing problem." In: *Computers & Operations Research* 37.12 (2010), pp. 2202–2217.

[102] Miguel F Anjos, Antonio J Conejo, et al. "Unit commitment in electric energy systems." In: *Foundations and Trends® in Electric Energy Systems* 1.4 (2017), pp. 220–310.

[103] Jaeho Choi and Joongheon Kim. "A tutorial on quantum approximate optimization algorithm (QAOA): Fundamentals and applications." In: *2019 international conference on information and communication technology convergence (ICTC)*. IEEE. 2019, pp. 138–142.

[104] Marika Svensson et al. "Hybrid Quantum-Classical Heuristic to Solve Large-Scale Integer Linear Programs." In: *Physical Review Applied* 20.3 (2023), p. 034062.

[105] Tony Tran et al. "A hybrid quantum-classical approach to solving scheduling problems." In: *Proceedings of the International Symposium on Combinatorial Search*. Vol. 7. 1. 2016, pp. 98–106.

[106] *Using QAOA and QAA to solve a UD-MIS problem*. https://pulser.readthedocs.io/en/latest/tutorials/qaoa_mis.html. Accessed: 2022-11-05.

[107] Shihab Jimaa et al. "LTE-A an Overview and Future Research Areas." In: *Proceedings of the 2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications*. WIMOB '11. USA: IEEE Computer Society, 2011, pp. 395–399. ISBN: 9781457720130. DOI: 10.1109/WiMOB.2011.6085370. URL: https://doi.org/10.1109/WiMOB.2011.6085370.

[108] H. Yang et al. "Interference Self-Coordination: A Proposal to Enhance Reliability of System-Level Information in OFDM-Based Mobile Networks via PCI Planning." In: *IEEE Transactions on Wireless Communications* 13.4 (2014), pp. 1874–1887.

[109] Rocío Acedo-Hernández et al. "A PCI planning algorithm for jointly reducing reference signal collisions in LTE uplink and downlink." In: *Computer Networks* 119 (2017), pp. 112–123.

[110] Hakan Kavlak and Hakki Ilk. "PCI Planning Strategies for Long Term Evolution Networks." In: *NETWORKING 2012 Workshops*. Ed. by Zdenek Becvar, Robert Bestak, and Lukas Kencl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 151–156. ISBN: 978-3-642-30039-4.

[111] Jihong Gui, Zhipeng Jiang, and Suixiang Gao. "PCI Planning Based on Binary Quadratic Programming in LTE/LTE-A Networks." In: *IEEE Access* 7 (2019), pp. 203–214. ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2885313.

[112] Fatos Xhafa and Ajith Abraham. "Computational models and heuristic methods for Grid scheduling problems." In: *Future generation computer systems* 26.4 (2010), pp. 608–621.

[113] Safa Khalouli, Fatima Ghedjati, and Abdelaziz Hamzaoui. "A meta-heuristic approach to solve a JIT scheduling problem in hybrid flow shop." In: *Engineering Applications of Artificial Intelligence* 23.5 (2010), pp. 765–771.

[114] Fermín Alfredo Tang Montané and Roberto Diéguez Galvao. "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service." In: *Computers & Operations Research* 33.3 (2006), pp. 595–619.

[115] Tameem Albash and Daniel A Lidar. "Adiabatic quantum computation." In: *Reviews of Modern Physics* 90.1 (2018), p. 015002.

[116] Alane Marie de Lima and Renato Carmo. "Exact Algorithms for the Graph Coloring Problem." In: *Revista de Informática Teórica e Aplicada* 25.4 (Nov. 2018), pp. 57–73. DOI: 10.22456/2175-2745.80721. URL: https://seer.ufrgs.br/index.php/rita/article/view/RITA_Vol25_Nr4_57.

[117] A. Hoffman and Leonard Howes. "On eigenvalues and colorings of graphs. II." In: *Annals of the New York Academy of Sciences* 175 (Mar. 2007), pp. 238–242. DOI: 10.1111/j.1749-6632.1970.tb56474.x.

[118] Clive Elphick and Pawel Wocjan. "An inertial lower bound for the chromatic number of a graph." In: *arXiv preprint arXiv:1605.01978* (2016).

[119] CS Edwards and CH Elphick. "Lower bounds for the clique and the chromatic numbers of a graph." In: *Discrete Applied Mathematics* 5.1 (1983), pp. 51–64.

[120] Jesper Makholm Byskov. "Enumerating maximal independent sets with applications to graph colouring." In: *Operations Research Letters* 32.6 (2004), pp. 547–556.

[121] Michel Fabrice Serret, Bertrand Marchand, and Thomas Ayral. "Solving Optimization Problems with Rydberg Analog Quantum Computers: Realistic Requirements for Quantum Advantage Using Noisy Simulation and Classical Benchmarks." In: *Physical Review A* 102.5 (Nov. 2020), p. 052617. ISSN: 2469-9926, 2469-9934. DOI: 10.1103/PhysRevA.102.052617.

[122] Pierre Hansen, Martine Labbé, and David Schindl. "Set covering and packing formulations of graph coloring: Algorithms and first polyhedral results." In: *Discrete Optimization* 6.2 (2009), pp. 135–147.

[123] Dominique Feillet. "A tutorial on column generation and branch-and-price for vehicle routing problems." In: *4or* 8.4 (2010), pp. 407–424.

[124] Martin Savelsbergh. "A branch-and-price algorithm for the generalized assignment problem." In: *Operations research* 45.6 (1997), pp. 831–841.

[125] François Vanderbeck. "Branching in branch-and-price: a generic scheme." In: *Mathematical Programming* 130 (2011), pp. 249–294.

[126] Ruslan Sadykov and François Vanderbeck. "Bin packing with conflicts: a generic branch-and-price algorithm." In: *INFORMS Journal on Computing* 25.2 (2013), pp. 244–255.

[127] Linling Kuang et al. "Radio resource management in future terrestrial-satellite communication networks." In: *IEEE Wireless Communications* 24.5 (2017), pp. 81–87.

[128] Elias Yaacoub and Mohamed-Slim Alouini. "A Key 6G Challenge and Opportunity-Connecting the Remaining 4 Billions: A Survey on Rural Connectivity." In: (2019).

[129] Yongtao Su et al. "Broadband LEO satellite communications: Architectures and key technologies." In: *IEEE Wireless Communications* 26.2 (2019), pp. 55–61.

[130] Xiangming Zhu and Chunxiao Jiang. "Integrated satellite-terrestrial networks toward 6G: Architectures, applications, and challenges." In: *IEEE Internet of Things Journal* 9.1 (2021), pp. 437–461.

[131] Sami Ma et al. "LEO Satellite Network Access in the Wild: Potentials, Experiences, and Challenges." In: *IEEE Network* (2024).

[132] Edric Matwiejew et al. "Continuous-time quantum walk-based ansätze on neutral atom hardware." In: *arXiv preprint arXiv:2509.00386* (2025).

[133] William E Hart et al. *Pyomo—Optimization Modeling in Python*. Tech. rep. Springer, 2017.

[134] Jonathan Wurtz, Stefan H Sack, and Sheng-Tao Wang. "Solving non-native combinatorial optimization problems using hybrid quantum-classical algorithms." In: *IEEE Transactions on Quantum Engineering* (2024).

[135] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. "An introduction to quantum machine learning." In: *Contemporary Physics* 56.2 (2015), pp. 172–185.

[136] Maria Schuld and Nathan Killoran. "Quantum machine learning in feature Hilbert spaces." In: *Physical review letters* 122.4 (2019), p. 040504.

[137] Keisuke Fujii and Kohei Nakajima. "Harnessing disordered-ensemble quantum dynamics for machine learning." In: *Physical Review Applied* 8.2 (2017), p. 024030.

[138] Keisuke Fujii and Kohei Nakajima. "Quantum reservoir computing: a reservoir approach toward quantum machine learning on near-term quantum devices." In: *Reservoir Computing: Theory, Physical Implementations, and Applications*. Springer, 2021, pp. 423–450.

[139] David Peral-García, Juan Cruz-Benito, and Francisco José García-Peñalvo. "Systematic literature review: Quantum machine learning and its applications." In: *Computer Science Review* 51 (2024), p. 100619. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2024.100619. URL: https://www.sciencedirect.com/science/article/pii/S1574013724000030.

[140] Ricardo Daniel Monteiro Simões et al. "Experimental evaluation of quantum machine learning algorithms." In: *IEEE access* 11 (2023), pp. 6197–6208.

[141] Kushal Batra et al. "Quantum machine learning algorithms for drug discovery applications." In: *Journal of chemical information and modeling* 61.6 (2021), pp. 2641–2647.

[142] Boris Albrecht et al. "Quantum feature maps for graph machine learning on a neutral atom quantum processor." In: *Physical Review A* 107.4 (Apr. 2023). ISSN: 2469-9934. DOI: 10.1103/physreva.107.042615. URL: http://dx.doi.org/10.1103/PhysRevA.107.042615.

[143] Louis-Paul Henry et al. "Quantum evolution kernel: Machine learning on graphs with programmable arrays of qubits." In: *Physical Review A* 104.3 (2021), p. 032416.

[144] Tahereh Pourhabibi et al. "Fraud detection: A systematic literature review of graph-based anomaly detection approaches." In: *Decision Support Systems* 133 (2020), p. 113303. ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2020.113303. URL: https://www.sciencedirect.com/science/article/pii/S0167923620300580.

[145] Athanasios Theocharidis et al. "Network Visualisation and Analysis of Gene Expression Data using BioLayout." In: *Nature protocols* 4 (Oct. 2009), pp. 1535–50. DOI: 10.1038/nprot.2009.177.

[146] Giulia Muzio, Leslie O'Bray, and Karsten Borgwardt. "Biological network analysis with deep learning." In: *Briefings in Bioinformatics* 22.2 (Nov. 2020), pp. 1515–1530. ISSN: 1477-4054. DOI: 10.1093/bib/bbaa257. eprint: https://academic.oup.com/bib/article-pdf/22/2/1515/36655309/bbaa257.pdf. URL: https://doi.org/10.1093/bib/bbaa257.

[147] Carlos D. Correa and Kwan-Liu Ma. "Visualizing Social Networks." In: *Social Network Data Analytics*. Ed. by Charu C. Aggarwal. Boston, MA: Springer US, 2011, pp. 307–326. ISBN: 978-1-4419-8462-3. DOI: 10.1007/978-1-4419-8462-3_11. URL: https://doi.org/10.1007/978-1-4419-8462-3_11.

[148] Maria Schuld and Nathan Killoran. "Quantum Machine Learning in Feature Hilbert Spaces." In: *Physical Review Letters* 122.4 (Feb. 2019). ISSN: 1079-7114. DOI: 10.1103/physrevlett.122.040504. URL: http://dx.doi.org/10.1103/PhysRevLett.122.040504.

[149] Maria Schuld et al. "Measuring the similarity of graphs with a Gaussian boson sampler." In: *Phys. Rev. A* 101 (3 Mar. 2020), p. 032314. DOI: 10.1103/PhysRevA.101.032314. URL: https://link.aps.org/doi/10.1103/PhysRevA.101.032314.

[150] Kaito Kishi et al. "Graph Kernels Encoding Features of All Subgraphs by Quantum Superposition." In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 12.3 (Sept. 2022), pp. 602–613. ISSN: 2156-3365. DOI: 10.1109/jetcas.2022.3200837. URL: http://dx.doi.org/10.1109/JETCAS.2022.3200837.

[151] Iris Cong, Soonwon Choi, and Mikhail D. Lukin. "Quantum convolutional neural networks." In: *Nature Physics* 15.12 (Aug. 2019), pp. 1273–1278. ISSN: 1745-2481. DOI: 10.1038/s41567-019-0648-8. URL: http://dx.doi.org/10.1038/s41567-019-0648-8.

[152] Jin Zheng, Qing Gao, and Yanxuan Lv. *Quantum Graph Convolutional Neural Networks*. 2021. arXiv: 2107.03257 [eess.SP].

[153] Guillaume Verdon et al. *Quantum Graph Neural Networks*. 2019. arXiv: 1909.12264 [quant-ph].

[154] Vojtěch Havlíček et al. "Supervised learning with quantum-enhanced feature spaces." In: *Nature* 567.7747 (Mar. 2019), pp. 209–212. ISSN: 1476-4687. DOI: 10.1038/s41586-019-0980-2. URL: http://dx.doi.org/10.1038/s41586-019-0980-2.

[155] Péter Mernyei, Konstantinos Meichanetzidis, and İsmail İlkan Ceylan. *Equivariant Quantum Graph Circuits*. 2022. arXiv: 2112.05261 [cs.LG].

[156] Andrea Skolik et al. *Equivariant quantum circuits for learning on weighted graphs*. 2023. arXiv: 2205.06109 [quant-ph].

[157] Martín Larocca et al. "Group-Invariant Quantum Machine Learning." In: *PRX Quantum* 3.3 (Sept. 2022). ISSN: 2691-3399. DOI: 10.1103/prxquantum.3.030341. URL: http://dx.doi.org/10.1103/PRXQuantum.3.030341.

[158] Herbert Jaeger. "Adaptive nonlinear system identification with Echo state networks." In: *Proceedings of the 15th International Conference on Neural Information Processing Systems*. NIPS'02. Cambridge, MA, USA: MIT Press, 2002, pp. 609–616.

[159] Troy Arcomano et al. "A Hybrid Atmospheric Model Incorporating Machine Learning Can Capture Dynamical Processes Not Captured by Its Physics-Based Component." In: *grl* 50.8, e2022GL102649 (Apr. 2023), e2022GL102649. DOI: 10.1029/2022GL102649.

[160] Azarakhsh Jalalvand, Glenn Van Wallendael, and Rik Van De Walle. "Real-Time Reservoir Computing Network-Based Systems for Detection Tasks on Visual Contents." In: *2015 7th International Conference on Computational Intelligence, Communication Systems and Networks*. 2015, pp. 146–151. DOI: 10.1109/CICSyN.2015.35.

[161] Nils Schaetti, Michel Salomon, and Raphaël Couturier. "Echo State Networks-Based Reservoir Computing for MNIST Handwritten Digits Recognition." In: *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*. 2016, pp. 484–491. DOI: 10.1109/CSE-EUC-DCABES.2016.229.

[162] Gouhei Tanaka et al. "Recent advances in physical reservoir computing: A review." In: *Neural Networks* 115 (2019), pp. 100–123. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2019.03.005. URL: https://www.sciencedirect.com/science/article/pii/S0893608019300784.

[163] Keisuke Fujii and Kohei Nakajima. "Harnessing Disordered-Ensemble Quantum Dynamics for Machine Learning." In: *Phys. Rev. Appl.* 8 (2 Aug. 2017), p. 024030. DOI: 10.1103/PhysRevApplied.8.024030. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.8.024030.

[164] Toshiki Yasuda et al. "Quantum reservoir computing with repeated measurements on superconducting devices." In: (Oct. 2023). arXiv: 2310.06706 [quant-ph].

[165] Kaito Kobayashi, Keisuke Fujii, and Naoki Yamamoto. "Feedback-Driven Quantum Reservoir Computing for Time-Series Analysis." In: *PRX Quantum* 5 (4 Nov. 2024), p. 040325. DOI: 10.1103/PRXQuantum.5.040325. URL: https://link.aps.org/doi/10.1103/PRXQuantum.5.040325.

[166] Rodrigo Martínez-Peña et al. "Dynamical Phase Transitions in Quantum Reservoir Computing." In: *Physical Review Letters* 127.10 (Aug. 2021). ISSN: 1079-7114. DOI: 10.1103/physrevlett.127.100502. URL: http://dx.doi.org/10.1103/PhysRevLett.127.100502.

[167] Rodrigo Araiza Bravo et al. "Quantum Reservoir Computing Using Arrays of Rydberg Atoms." In: *PRX Quantum* 3 (3 Aug. 2022), p. 030325. DOI: 10.1103/PRXQuantum.3.030325. URL: https://link.aps.org/doi/10.1103/PRXQuantum.3.030325.

[168] Milan Kornjača et al. *Large-scale quantum reservoir learning with an analog quantum computer*. 2024. arXiv: 2407.02553 [quant-ph]. URL: https://arxiv.org/abs/2407.02553.

[169] Ryan A. Rossi and Nesreen K. Ahmed. "The Network Data Repository with Interactive Graph Analytics and Visualization." In: *AAAI*. 2015. URL: https://networkrepository.com.

[170] *Bloqade.jl*. URL: https://queracomputing.github.io/Bloqade.jl/stable/ (visited on 02/13/2024).

[171] *The Julia Programming Language*. URL: https://julialang.org/ (visited on 02/13/2024).

[172] Peter I Frazier. "A tutorial on Bayesian optimization." In: *arXiv preprint arXiv:1807.02811* (2018).

[173] Karsten M Borgwardt and Hans-Peter Kriegel. "Shortest-path kernels on graphs." In: *Fifth IEEE international conference on data mining (ICDM'05)*. IEEE. 2005, 8–pp.

[174] Matthias C. Caro et al. "Generalization in quantum machine learning from few training data." In: *Nature Communications* 13.1 (Aug. 2022). ISSN: 2041-1723. DOI: 10.1038/s41467-022-32550-3. URL: http://dx.doi.org/10.1038/s41467-022-32550-3.

[175] I-Cheng Yeh. *Default of Credit Card Clients*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C55S3H. 2009.

[176] I-Cheng Yeh and Che-hui Lien. "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients." In: *Expert Systems with Applications* 36.2, Part 1 (2009), pp. 2473–2480. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2007.12.020. URL: https://www.sciencedirect.com/science/article/pii/S0957417407006719.

[177] Sheikh Rabiul Islam, William Eberle, and Sheikh Khaled Ghafoor. *Credit Default Mining Using Combined Machine Learning and Heuristic Approach*. 2018. arXiv: 1807.01176 [cs.LG]. URL: https://arxiv.org/abs/1807.01176.

[178] Tengke Xiong et al. "Personal bankruptcy prediction by mining credit card data." In: *Expert Systems with Applications* 40.2 (2013), pp. 665–676. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.07.072. URL: https://www.sciencedirect.com/science/article/pii/S0957417412009281.

[179] Hongya Lu, Haifeng Wang, and Sang Won Yoon. "Real Time Credit Card Default Classification Using Adaptive Boosting-Based Online Learning Algorithm." In: *Proceedings of the 2017 Industrial and Systems Engineering Research Conference*. May 2017.

[180] Daniel J Gauthier et al. "Next generation reservoir computing." In: *Nature communications* 12.1 (2021), pp. 1–8.

[181] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[182] Gareth James et al. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. URL: https://faculty.marshall.usc.edu/gareth-james/ISL/.

[183] Guillaume Lemaître, Fernando Nogueira, and Christos K Aridas. "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning." In: *Journal of machine learning research* 18.17 (2017), pp. 1–5.

[184] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique." In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[185] Wei-Chao Lin et al. "Clustering-based undersampling in class-imbalanced data." In: *Information Sciences* 409 (2017), pp. 17–26.

[186] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. "PyTorch." In: *Programming with TensorFlow: solution for edge computing applications* (2021), pp. 87–104.

[187] Yicheng Luo and collaborators. *Bloqade.jl: A Julia package for simulating programmable quantum systems*. https://queracomputing.github.io/Bloqade.jl/dev/Bloqade.jl. Accessed: 2025-05-13. 2023.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was LuaLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi =scudo`. This class is available in every up-to-date and complete TEX-system installation.