

Article

Dissipative Realization of a Quantum Distance-Based Classifier Using Open Quantum Walks

Pedro Linck Maciel ^{1,*} , Graeme Pleasance ² , Francesco Petruccione ^{2,3}  and Nadja K. Bernardes ¹ 

¹ Departamento de Física, Centro de Ciências Exatas e da Natureza, Universidade Federal de Pernambuco, Recife 50670-901, Brazil; nadja.bernardes@ufpe.br

² National Institute for Theoretical and Computational Sciences (NITheCS), Stellenbosch 7600, South Africa; gpleasance1@gmail.com (G.P.)

³ School for Data Science and Computational Thinking, Stellenbosch University, Stellenbosch 7600, South Africa

* Correspondence: pedro.linck@ufpe.br

Abstract

Open quantum walks (OQWs) constitute a class of quantum walks whose dynamics are entirely driven by interactions with the environment. It is well known that OQWs provide a general framework for implementing dissipative quantum computation. In this work, we demonstrate the feasibility of running the previously proposed quantum distance-based classifier within the open quantum walk computation model, and we show that its expected runtime remains finite even in the slower regime.

Keywords: open quantum systems; quantum machine learning; dissipative quantum computation; open quantum walks; distance-based classifier; quantum distance-based classifier; Hadamard classifier

1. Introduction

Quantum computation, with its ability to explore the principles of quantum mechanics, has emerged as a promising paradigm for solving complex computational problems exponentially faster than classical computers. However, the development of practical and scalable quantum computers faces numerous challenges, including the detrimental effects of noise and decoherence. Quantum systems are inherently susceptible to environmental interactions, which can lead to loss of quantum information and degradation in computational performance.

To overcome these obstacles, researchers have been exploring a field known as dissipative quantum computation [1–3]. Dissipative quantum computation aims to explore the unavoidable interactions between a quantum system and its surrounding environment, turning them from a detrimental effect into a resource for performing computational tasks. Instead of isolating the quantum system from its environment, as is commonly done in traditional quantum computing approaches, dissipative computation harnesses the dissipative dynamics to manipulate and process information.

An efficient implementation of dissipative quantum computing algorithms is the open quantum walk (OQW) [2,4–6]. Linear OQWs have been used for the design of algorithms [5], and its dynamics and computational limits have been established [7]. OQWs offer a noise-resilient framework for dissipative quantum computation in which, at least in principle, one can engineer a reservoir so that the computation is driven entirely by the environment's dissipative dynamics. Moreover, for several OQW schemes there exists a useful class of convergence theorems [8,9]. They also provide a natural way to incorporate a Markovian



Academic Editors: Paolo Bordone and Gioacchino Massimo Palma

Received: 29 December 2025

Revised: 26 January 2026

Accepted: 15 February 2026

Published: 19 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

structure that is not readily available in standard coined quantum walks. In particular, linear open quantum walks have well-established convergence properties and a simple steady state [7]. More broadly, because open system evolution extends beyond unitary dynamics, the OQW formalism allows one to explore general Kraus maps, opening the door to richer and more complex dynamical behavior.

Random walks, a fundamental concept in mathematics and physics, describe the stochastic motion of a walker in discrete steps, jumping from one node to another. Open quantum walks offer a novel framework for exploring the interplay between quantum dynamics and environmental noise, shedding light on the behavior of quantum systems under realistic and imperfect conditions. In this model, the transition between the nodes is driven purely by the dissipative interaction with an environment.

In the context of computation, a key area where quantum technology may prove to be beneficial is the field of machine learning [10–12]. Quantum machine learning using open quantum systems has also demonstrated advantages [3,13–16]. Among the most basic machine learning tasks is classification, typically implemented via classifier algorithms [17]. In this work, we focus on a linear classifier that has been simulated on a quantum computer by Schuld et al. [18,19]. Recently, there have also been proposals exploring dissipative quantum systems to construct quantum classifiers [3,20,21]. The idea is to encode the initialization information in the preparation of the physical system and the weight in the environmental factors (interaction and environmental states).

This work presents an implementation of a quantum distance-based classifier in an OQW model as a proof of principle. Section 2 summarizes the OQW results needed for computation in this framework. Section 3 reviews the classifier and establishes, with proofs, both correctness and the expected post-selection success probability, extending earlier empirical claims. Section 4 provides the OQW realization and validates the algorithm. Finally, a conclusion and future perspectives are presented in Section 5.

2. Open Quantum Walks

In this section, we develop the basic tools of open quantum walks needed to understand the results of this paper.

2.1. Basic Properties of Open Quantum Walks

A quantum walk [22,23] is a form of quantum evolution that occurs on a graph. The vertices of the graph form the basis of the graph space where the quantum walker is situated and can jump between nodes depending on both the graph structure and the internal walker state [24]. The dynamics, or time evolution, of a discrete quantum walk refers to the repeated application of the single-step evolution rule. Several quantum walks have been shown to model quantum computation [2,25,26]. Here, we focus on OQWs, a class of quantum walks whose dynamics is entirely induced by the environment.

A general quantum evolution (closed or open) is described as a map $\Lambda: \mathcal{B}(\mathcal{V}) \rightarrow \mathcal{B}(\mathcal{V})$, where $\mathcal{B}(\mathcal{V})$ denotes the algebra of bounded operators in the Hilbert space $\mathcal{V}$. The map Λ must be linear, completely positive and trace-preserving [27]. It is known [27] that if $\dim(\mathcal{V}) = d$, then we can write

$$\Lambda(\rho) = \sum_{i=0}^{k-1} K_i \rho K_i^\dagger, \quad (1)$$

where $k \leq d^2$ and

$$\sum_i K_i^\dagger K_i = \mathbb{1}. \quad (2)$$

This set of operators $\{K_i\}_{i=0}^{k-1}$ is called a set of Kraus operators for the map Λ .

To formally define an open quantum walk, we need to define $\mathcal{H}$, the (internal) Hilbert space of the walker, and the graph Hilbert space $\mathcal{G}$ where the walk is made. We mark a basis $\{|i\rangle\}_{i \in G}$ for the graph space, where G is the set of vertices of the underlying graph. For each $i \in G$ we also have a set $\{B_i^j\}_{j \in G}$ of linear operators $B_i^j: \mathcal{H} \rightarrow \mathcal{H}$ that satisfy

$$\sum_j B_i^{j\dagger} B_i^j = \mathbb{1}, \quad (3)$$

which will act on the internal state when a jump is made from node i to node j . In order to capture the jump together with this change in internal state, we define operators

$$M_i^j = B_i^j \otimes |j\rangle\langle i| \in \mathcal{B}(\mathcal{H} \otimes \mathcal{G}). \quad (4)$$

One can verify that Equation (3) implies

$$\sum_{i,j} M_i^{j\dagger} M_i^j = \mathbb{1}, \quad (5)$$

and therefore this set of operators defines a quantum channel. This set of Kraus operators defines an open quantum walk. An open quantum walk can be seen pictorially as a graph with labeled edges, where each edge from i to j is labeled B_i^j . Therefore, all information about the open quantum walk evolution map can be extracted from a given graphic representation. The recursive relations for an OQW with initial state $\rho_0 \in \mathcal{B}(\mathcal{H} \otimes \mathcal{G})$ are

$$\begin{cases} \rho^{[0]} = \rho_0, \\ \rho^{[n]} = \sum_{i,j} M_i^j \rho^{[n-1]} M_i^{j\dagger}, \quad n \geq 1 \end{cases} \quad (6)$$

where $\rho^{[n]}$ is the state of the open quantum walk after n steps. It is known [4] that if $\rho_0 = \sum_{i,j} \rho_{ij} \otimes |j\rangle\langle i|$ for some trace-class matrices ρ_{ij} satisfying $\sum_i \text{Tr}(\rho_{ii}) = 1$, then after any $n \geq 1$ steps, $\rho^{[n]}$ has the form

$$\rho^{[n]} = \sum_i \rho_{ii}^{[n]} \otimes |i\rangle\langle i|, \quad (7)$$

where $\rho_{ii}^{[n]}$ are some trace-class matrices determined by the evolution defined by Equation (6). Note that if even a single step is made, the walker decouples from the graph space. A graphical representation of a general OQW is shown in Figure 1. The edges from vertex $|i\rangle$ to vertex $|j\rangle$ are labeled B_i^j , and any omitted edges are taken to correspond to the zero operator.

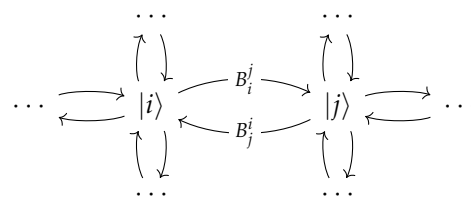


Figure 1. An arbitrary open quantum walk can be represented by this visual diagram. If there is an omitted edge in a particular diagram, this means that the corresponding operator B_i^j is zero.

For example, for the two-node graph in Figure 2, we have operators $B_0^0, B_0^1, B_1^0, B_1^1$ satisfying

$$B_0^{0\dagger} B_0^0 + B_0^{1\dagger} B_0^1 = I, \quad (8)$$

$$B_1^{1\dagger} B_1^1 + B_1^{0\dagger} B_1^0 = I. \quad (9)$$

The state of the walker after n steps will be

$$\rho^{[n]} = \rho_0^{[n]} \otimes |0\rangle\langle 0| + \rho_1^{[n]} \otimes |1\rangle\langle 1|, \quad (10)$$

where

$$\rho_0^{[n]} = B_0^0 \rho_0^{[n-1]} B_0^{0\dagger} + B_1^0 \rho_1^{[n-1]} B_1^{0\dagger}, \quad (11)$$

$$\rho_1^{[n]} = B_0^1 \rho_0^{[n-1]} B_0^{1\dagger} + B_1^1 \rho_1^{[n-1]} B_1^{1\dagger}. \quad (12)$$

If we have, for instance, $B_0^0 = \sqrt{\lambda}I$, $B_0^1 = \sqrt{\omega}U$, $B_1^0 = \sqrt{\lambda}U^\dagger$, $B_1^1 = \sqrt{\omega}I$, where U is any unitary operator, $\omega, \lambda \geq 0$ are any non-negative real numbers satisfying $\omega + \lambda = 1$ and the initial state of the QW is $\rho_0 = |\psi\rangle\langle\psi| \otimes |0\rangle\langle 0|$, then we have [7]

$$\rho^{[n]} = \lambda |\psi\rangle\langle\psi| \otimes |0\rangle\langle 0| + \omega U |\psi\rangle\langle\psi| U^\dagger \otimes |1\rangle\langle 1|. \quad (13)$$

In this way, if we want to perform quantum computation and apply a gate U to the state $|\psi\rangle$, we have to post-select the result in node $|1\rangle$, which has a success probability of ω .

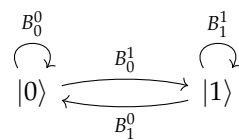


Figure 2. An arbitrary two-node open quantum walk.

2.2. Linear Open Quantum Walks for Quantum Computation

In this work, we use a linear topology of an open quantum walk [7] to perform quantum computation (i.e., the preparation of quantum states), where all Kraus operators of the evolution are proportional to unitaries. The graphic representation for this QW model is shown in Figure 3. Note that if $\rho_0 = |\psi\rangle\langle\psi| \otimes |0\rangle\langle 0|$ we have (using the convention $U_{-1} = \mathbb{1}$)

$$\rho^{[n]} = \sum_i p_i^{(n)} U_{i-1} \cdots U_0 |\psi\rangle\langle\psi| U_0^\dagger \cdots U_{i-1}^\dagger \otimes |i\rangle\langle i|, \quad (14)$$

where $p_i^{(n)}$ is the probability of finding the walker in node i after n steps. This can be seen as a classical Markov chain [7] with a transition matrix [28–30]

$$T = \begin{bmatrix} \lambda & \lambda & 0 & & \\ \omega & 0 & \lambda & & \\ 0 & \omega & 0 & & \\ \vdots & 0 & \omega & & \\ \vdots & & 0 & \ddots & \\ & & & \ddots & \ddots \end{bmatrix}, \quad (15)$$

and the steady-state π can be calculated:

$$\pi_m = \frac{a^m(a-1)}{a^N-1}, \quad (16)$$

where $a = \frac{\omega}{\lambda} = \frac{\omega}{1-\omega}$.

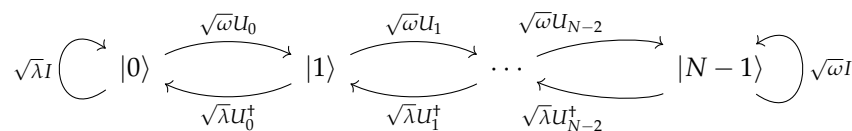


Figure 3. The diagram corresponding to the linear OQW model. Each U_i is a unitary operator, and $\omega, \lambda \geq 0$ are such that $\omega + \lambda = 1$.

3. Distance-Based Classifier Algorithms

In this section, we develop the basic tools for classical and quantum distance-based classifiers needed to understand the results of this paper, and we provide rigorous proofs of several claims about the quantum distance-based classifier that were previously supported empirically.

3.1. Classical Distance-Based Classifier

We begin by highlighting the basic principles of a classical distance-based classifier [17]. In a general classification task, one considers a vector space of parameters χ together with a finite set of possible classes $C = 0, 1, \dots, c-1$, where each vector of parameters is uniquely associated with a class. As an illustrative example, the parameter vector may represent measurable attributes such as the height and weight of individuals, while the classes correspond to distinct categories, for instance, children and adults. The goal of the classification task is to infer the class associated with a given parameter vector. When a labeled dataset is available (meaning that the class corresponding to each parameter vector is known) the classification procedure is referred to as supervised learning.

Here, we focus on distance-based classifiers. This class of algorithms requires a kernel function $K: \chi \times \chi \rightarrow \mathbb{R}_{\geq 0}$, which is typically defined as an inner product $K(\vec{x}, \vec{x}') = \langle \varphi(\vec{x}), \varphi(\vec{x}') \rangle$, where $\varphi: \chi \rightarrow \chi$ is a feature-mapping function and $\langle \cdot, \cdot \rangle$ can be any inner product, and in this work we use the usual inner product in $\mathbb{R}^N$: $\langle (x_1, \dots, x_N), (x'_1, \dots, x'_N) \rangle = x_1 x'_1 + \dots + x_N x'_N$. We assume $C = \{-1, 1\}$ and that we receive a dataset $D = \{(\vec{x}_m, y_m)\}_{m=0}^{M-1}$, where each $\vec{x}_i$ is a parameter vector and $y_i \in C$ denotes its corresponding class. The inference rule is then defined as

$$y = \text{sgn} \left(\sum_m y_m K(\vec{x}_m, \vec{x}) \right). \quad (17)$$

This equation admits a natural probabilistic interpretation. Interpreting $K(\vec{x}_m, \vec{x})$ as an unnormalized probability weight, the inference rule assigns the class of the input parameter $\vec{x}$ according to the sign of its weighted mean value: a positive mean corresponds to $y = 1$, while a negative mean corresponds to $y = -1$. This interpretation is derived from the fact that when $\vec{x}$ is close to $\vec{x}_m$, their inner product approaches its maximum value. This will correspond, in the probabilistic picture, to a likelihood close to one. In this sense, two adjacent parameter vectors are highly likely to belong to the same class and K can thus be viewed as a likelihood function that quantifies their proximity.

In this work, we adopt the following kernel function:

$$K(\vec{x}, \vec{x}') = 1 - \frac{1}{4M} |\vec{x} - \vec{x}'|^2. \quad (18)$$

3.2. Quantum Distance-Based Quantum Classifier

The main idea of the distance-based quantum classifier is to encode the data points in the amplitudes of a quantum state and, by exploring interference effects, to be able to predict the class of this point [18]. In the original work [18], the algorithm is built such that

it gives us the same prediction as the classical distance-based algorithm with the kernel function given by Equation (18).

The canonical way to embed a normalized parameter vector $\vec{x} = (x_1, \dots, x_N)$ in a quantum state is to use an N -dimensional Hilbert space and encode as follows:

$$|\psi_{\vec{x}}\rangle = \sum_{i \geq 1} x_i |i\rangle. \quad (19)$$

To perform the quantum implementation of the distance-based classifier outlined in Section 3.1, we first start with a state

$$|\mathcal{D}\rangle = \frac{1}{\sqrt{2M}} \sum_{m=0}^{M-1} |m\rangle (|0\rangle |\psi_{\vec{x}}\rangle + |1\rangle |\psi_{\vec{x}_m}\rangle) |y_m\rangle, \quad (20)$$

where $|y_m\rangle = |0\rangle$ if $y_m = -1$ and $|y_m\rangle = |1\rangle$ if $y_m = 1$. The first system works as a flag, the second one is an ancillary qubit, and the third one encodes the parameter vectors. We perform a Hadamard gate in the ancilla qubit, obtaining

$$\frac{1}{2\sqrt{M}} \sum_{m=0}^{M-1} |m\rangle (|0\rangle |\psi_{\vec{x}+\vec{x}_m}\rangle + |1\rangle |\psi_{\vec{x}-\vec{x}_m}\rangle) |y_m\rangle, \quad (21)$$

where $|\psi_{\vec{x} \pm \vec{x}_m}\rangle = |\psi_{\vec{x}}\rangle \pm |\psi_{\vec{x}_m}\rangle$. Now we perform a post-selection on the state of the ancilla, selecting the state $|0\rangle$:

$$\frac{1}{2\sqrt{p_{\text{acc}}M}} \sum_{m=0}^{M-1} \sum_{i=1}^N |m\rangle \langle x^i + x_m^i | i \rangle |y_m\rangle, \quad (22)$$

where

$$p_{\text{acc}} = \frac{1}{4M} \sum_m \langle \psi_{\vec{x}+\vec{x}_m} | \psi_{\vec{x}+\vec{x}_m} \rangle = \frac{1}{4M} \sum_m ||\vec{x} + \vec{x}_m||^2 \quad (23)$$

is the success probability for the post-selection. It is also experimentally known [18] that if the data is standardized (i.e., if the data is distributed with mean value equal to zero and standard deviation equal to one), then $p_{\text{acc}} \approx 1/2$. We prove this fact analytically, considering the data as a standardized random variable:

$$\begin{aligned} \mathbb{E}(p_{\text{acc}}) &= \frac{1}{4M} \sum_m \mathbb{E}[||\vec{x} + \vec{x}_m||^2] \\ &= \frac{1}{4M} \sum_m \mathbb{E}[2 + 2\langle \vec{x}, \vec{x}_m \rangle] \\ &= \frac{1}{4M} \sum_m (2 + 2\langle \vec{x}, \mathbb{E}[\vec{x}_m] \rangle) \\ &= \frac{1}{4M} \sum_m 2 \\ &= \frac{1}{2}, \end{aligned} \quad (24)$$

where $\mathbb{E}(\vec{x}_m) = 0$ since the dataset is standardized. We define

$$p(y = -1) = \frac{1}{4Mp_{\text{acc}}} \sum_{m: |y_m\rangle=|0\rangle} ||\vec{x} + \vec{x}_m||^2, \quad (25)$$

which corresponds to the sum of the amplitudes of each state with $|y_m\rangle = |0\rangle$. We also note that since the parameters are normalized, we have the following.

$$\begin{aligned} \frac{1}{4Mp_{\text{acc}}} \sum_m ||\vec{x} + \vec{x}_m||^2 + \frac{1}{4Mp_{\text{acc}}} \sum_m ||\vec{x} - \vec{x}_m||^2 \\ = \frac{1}{4Mp_{\text{acc}}} \sum_m (2||\vec{x}||^2 + 2||\vec{x}_m||^2) = \frac{1}{p_{\text{acc}}}. \end{aligned} \quad (26)$$

We can therefore write

$$p(y = -1) = \frac{1}{p_{\text{acc}}} \left(1 - \frac{1}{4M} \sum_{m: |y_m\rangle=|0\rangle} ||\vec{x} - \vec{x}_m||^2 \right), \quad (27)$$

and then we have

$$p(y = 1) = \frac{1}{p_{\text{acc}}} \left(1 - \frac{1}{4M} \sum_{m: |y_m\rangle=|1\rangle} ||\vec{x} - \vec{x}_m||^2 \right). \quad (28)$$

Since $p(y = 1) = 1 - p(y = -1)$, we predict $y = +1$ if $p(y = 1) > p(y = -1)$ and $y = -1$ otherwise. It is shown [18] that this algorithm produces the same output as the classical distance-based classifier outlined in Section 3.1. Here, we give a formal proof of the correctness of this algorithm. First of all, we have the following.

$$\mathbb{E}[y] = -1 \cdot p(y = -1) + 1 \cdot p(y = 1) = p(y = 1) - p(y = -1), \quad (29)$$

and also have

$$\begin{aligned} \sum_m y_m \left(1 - \frac{1}{4N} ||\vec{x}_m - \vec{x}||^2 \right) &= \\ \sum_{m: y_m=-1} - \left(1 - \frac{1}{4N} ||\vec{x}_m - \vec{x}||^2 \right) &+ \\ \sum_{m: y_m=1} \left(1 - \frac{1}{4N} ||\vec{x}_m - \vec{x}||^2 \right) &= \\ p_{\text{acc}}(p(y = 1) - p(y = -1)) &= p_{\text{acc}} \cdot \mathbb{E}[y]. \end{aligned} \quad (30)$$

Note that the sign of $\mathbb{E}[y]$ determines the outcome of the quantum distance-based classifier, while the sign on the left-hand side of Equation (30) corresponds to the result of the classical distance-based classifier. From this relation, if $p(y = -1) > p(y = 1)$, both the classical and quantum classifiers predict $y = -1$. The same reasoning applies to the case $y = 1$, demonstrating that the classical and quantum classifiers implement the same classification rule.

3.3. Quantum Circuit for Distance-Based Classifier

For a dataset containing two training examples with opposite labels, $\mathcal{D}' = \{(\vec{x}_0, -1), (\vec{x}_1, 1)\}$, drawn from an arbitrarily large underlying dataset, and a test point $\vec{x}$ specified by two features, it was shown in Ref. [19] that the corresponding classifier can be implemented using a simple two-qubit quantum circuit. We now follow the construction of this circuit. The feature vectors are defined as

$$|x_0\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle - \sin\left(\frac{\theta}{2}\right)|1\rangle, \quad (31)$$

$$|x_1\rangle = \cos\left(\frac{\phi}{2}\right)|0\rangle - \sin\left(\frac{\phi}{2}\right)|1\rangle, \quad (32)$$

$$|\tilde{x}\rangle = \cos\left(\frac{\gamma}{2}\right)|0\rangle - \sin\left(\frac{\gamma}{2}\right)|1\rangle, \quad (33)$$

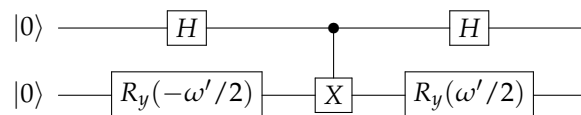
where θ can be set to zero without loss of generality such that $|x_0\rangle = |0\rangle$. In this case, the initial state used to perform the distance-based quantum classification has identical class and flag registers; therefore, a single qubit suffices to encode both.

$$\frac{1}{2}|0\rangle(|0\rangle|\tilde{x}\rangle + |1\rangle|x_0\rangle) + \frac{1}{2}|1\rangle(|0\rangle|\tilde{x}\rangle + |1\rangle|x_1\rangle). \quad (34)$$

The ratio t between the probabilities of measuring $|0\rangle$ and $|1\rangle$ in the class register is given by

$$t = \frac{P(|y\rangle = |0\rangle)}{P(|y\rangle = |1\rangle)} = \frac{\cos^2(\gamma/4)}{\cos^2((\gamma - \phi)/4)}. \quad (35)$$

The circuit implementing the reduced quantum distance-based classifier with two qubits is shown below. The second register corresponds to the $|y\rangle$ (class) qubit, while the first acts as an ancilla.



Here, ω' must be determined to reproduce the quantum algorithm described in Section 3.2. The portion of the quantum circuit responsible for state preparation comprises all gates except the final Hadamard gate. The state preparation stage of the circuit yields

$$\frac{1}{\sqrt{2}}(\cos(\omega'/2)|11\rangle - \sin(\omega'/2)|10\rangle + |00\rangle). \quad (36)$$

After the last Hadamard gate, the final state $|\psi\rangle$ is then

$$|\psi\rangle = \frac{1}{2}(-\cos(\omega'/2)|11\rangle + \cos(\omega'/2)|01\rangle + (1 + \sin(\omega'/2))|10\rangle + (1 - \sin(\omega'/2))|00\rangle). \quad (37)$$

After post-selecting $|0\rangle$ in the first register, we obtain the state $|\psi'\rangle$

$$|\psi'\rangle = \frac{1}{2\sqrt{p'_{\text{acc}}}}(\cos(\omega'/2)|01\rangle + (1 - \sin(\omega'/2))|00\rangle), \quad (38)$$

where

$$p'_{\text{acc}} = \frac{1 - \sin(\omega'/2)}{2}. \quad (39)$$

Classifying y based on the second qubit register, we obtain

$$\frac{P'(|y\rangle = |0\rangle)}{P'(|y\rangle = |1\rangle)} = \frac{(1 - \sin(\omega'/2))^2}{\cos^2(\omega'/2)}. \quad (40)$$

To obtain the same distribution probability as the quantum distance-based classifier, we impose the fact that the above equation is also equal to t . This gives us

$$\frac{(1 - \sin(\omega'/2))^2}{\cos^2(\omega'/2)} = t, \quad (41)$$

which, solving for ω' (see Appendix A), gives us

$$\omega' = 4 \arctan \left(\frac{1 - \sqrt{t}}{1 + \sqrt{t}} \right). \quad (42)$$

4. Distance-Based Quantum Classifier in the OQW Model

In this section, we implement the quantum distance-based classifier proposed by Schuld [18,19] in the open quantum walk model of quantum computation.

4.1. Implementing the Circuit in the Framework

We now show that the open quantum walk can be used to implement the quantum classifier described above. We restrict ourselves to the minimal two-feature implementation, since our goal is to provide a proof of principle that the algorithm can be feasibly implemented and executed within this framework. To this end, we consider the simplest possible scenario: two classes and two features. The minimal circuit executing this classifier is presented in Section 3.2 and consists of two qubits and three layers of gates. Within the OQW framework, this circuit is implemented through the following unitaries acting on the two qubits:

$$U_1 = H \otimes R_y(-\omega'/2), \quad (43)$$

$$U_2 = \text{CNOT}, \quad (44)$$

$$U_3 = H \otimes R_y(\omega'/2). \quad (45)$$

and is represented by a graph with four nodes, as depicted in Figure 4. Typically, each additional circuit layer is implemented in the OQW model by introducing one extra node together with its associated unitary, so that the graph size increases linearly with the circuit depth. Note that the internal degrees of freedom of the walker are represented by two qubits, and the graph itself is also encoded in two qubits; therefore, a total of four qubits are required for this implementation. The complete set of transition operators is given by $\{B_0^0 = \sqrt{\lambda}I, B_0^1 = \sqrt{\omega}U_1, B_1^0 = \sqrt{\lambda}U_1^\dagger, B_1^1 = \sqrt{\omega}U_2, B_2^1 = \sqrt{\lambda}U_2^\dagger, B_2^2 = \sqrt{\omega}U_3, B_3^2 = \sqrt{\lambda}U_3^\dagger, B_3^3 = \sqrt{\omega}I\}$.

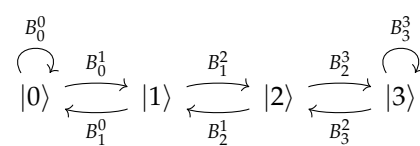


Figure 4. Open quantum walk representing the quantum classifier circuit.

To determine how many steps are required for the OQW to reach steady state, we initialize the walker in the state $|00\rangle\langle 00| \otimes |00\rangle\langle 00|$. The first two qubits represent the system that enters the circuit, while the last two qubits encode the node, labeled from 0 to 3. Since the result of the computation is always found in the last node, we first evaluate the detection probability at node 3 as a function of the number of OQW iterations n for different values of ω , as shown in Figure 5. From these results, we observe that 10 iterations are sufficient for the algorithm to converge.

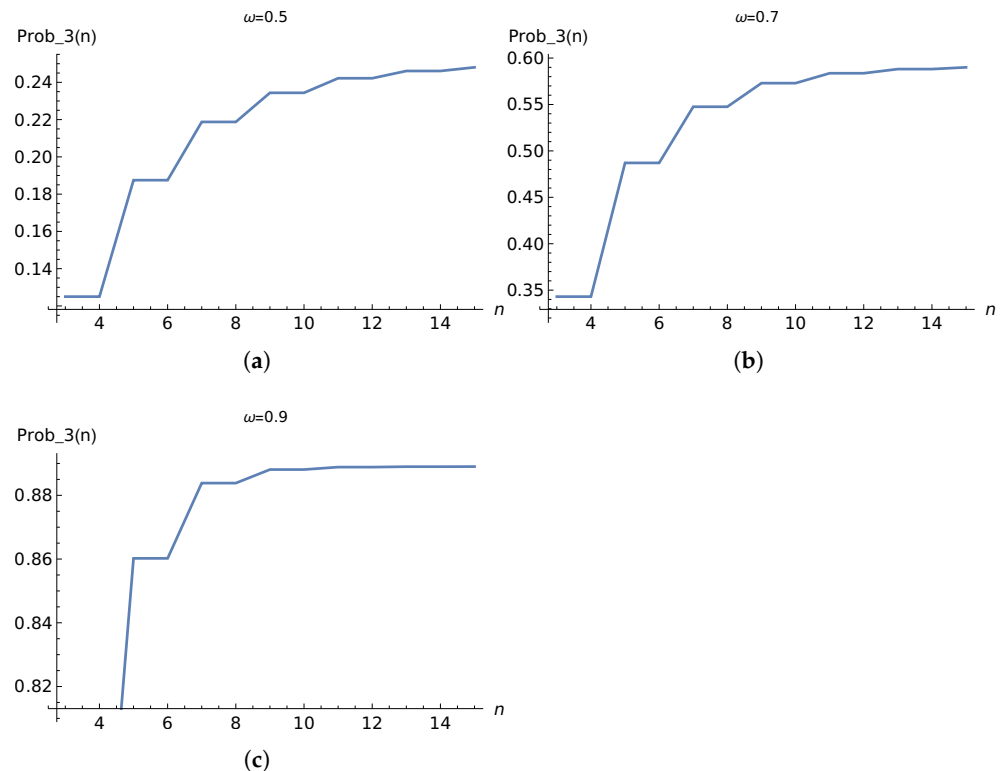


Figure 5. Detection probability of finding the walker in node 3 $P_3(n)$ versus the number of iterations n for different values of probability ω . (a) $\omega = 0.5$; (b) $\omega = 0.7$; (c) $\omega = 0.9$.

Alternatively, one could employ the analytical estimate derived in Ref. [7] for $\omega > 1/2$ to obtain

$$n_{\text{iterations}}(\omega) = \frac{N}{2\omega - 1}, \quad (46)$$

which in our case reduces to

$$n_{\text{iterations}}(\omega = 0.7) = 10 \quad (47)$$

$$n_{\text{iterations}}(\omega = 0.9) = 5. \quad (48)$$

In what follows, we use 10 iterations for all three values of ω to approach the steady state. However, it should be noted that these additional steps increase the probability of detection at the last node by only about 2% for $\omega = 0.9$. This improvement can be achieved with minimal computational cost as the underlying graph is small. In Figure 5, the detection probability of finding the walker in the last node is shown for different values of ω . In Figure 6, we can see the evolution of probabilities for $\omega = 0.7$. Also note that, although we are using a graph of size 4, Equation (46) is fully general and shows that the convergence time grows linearly with the graph size.

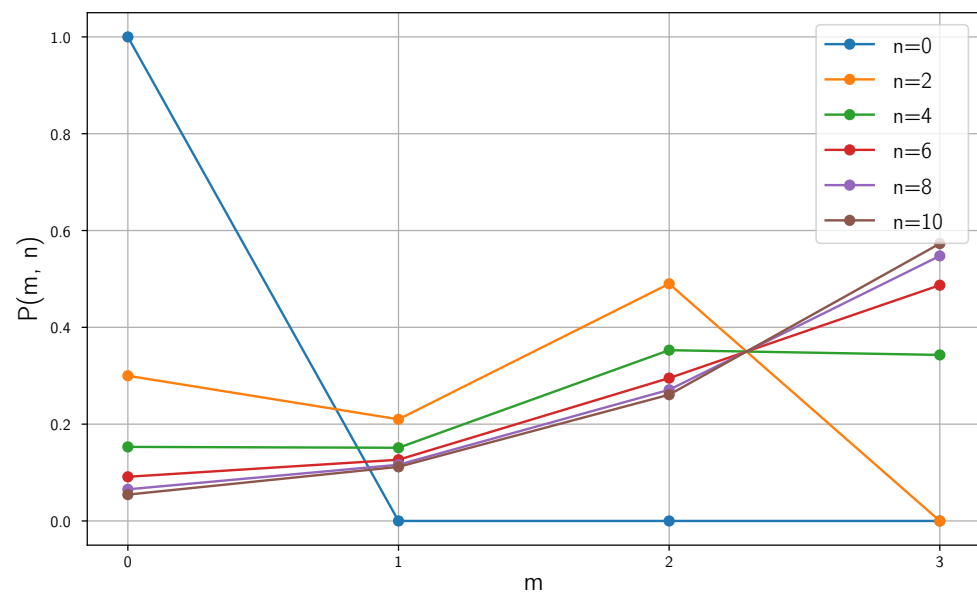


Figure 6. Evolution of probabilities in the linear OQW with $N = 4$ and $\omega = 0.7$. Distributions are shown at time steps $n = 0, 2, \dots, 10$.

4.2. Simulation Results

Now we can implement the classifier. We will consider the Iris dataset and will analyze only the setosa and versicolor classes and the width and length of the sepal feature. In this way, our dataset will be formed by a 2-dimensional vector and will contain 100 vectors. We will work with a standardized and normalized dataset.

To implement the circuit presented in the previous section, we essentially need to determine ω' . To this end, we randomly select two vectors, each belonging to one of the two distinct classes, and a third vector corresponding to the unknown point. These three data points are represented by

$$|x_0\rangle = |0\rangle, \quad (49)$$

$$|x_1\rangle = \cos(\phi/2)|0\rangle + \sin(\phi/2)|1\rangle, \quad (50)$$

$$|\tilde{x}\rangle = \cos(\gamma/2)|0\rangle + \sin(\gamma/2)|1\rangle. \quad (51)$$

For example, for Iris samples 34, 75, and 13, the standardized and normalized data vectors are given by $x_0 = (0.999807, 0.0196469)$, $x_1 = (-0.275974, 0.961165)$, and $\tilde{x} = (-0.194006, -0.981000)$. The state x_0 is taken as the reference, and in the rotated reference frame it becomes $x_0 = (0, 1)$, $x_1 = (-0.966402, -0.257036)$, and $\tilde{x} = (0.976999, -0.213242)$. This corresponds to the rotation angles $\phi = -8.90487$ and $\gamma = -3.57138$, leading to $\omega' = 1.06501$.

We then run the quantum classifier for 2000 triples of points, and the probability of correctly identifying the class of the unknown point is shown in Table 1 for different values of the parameter ω , averaged over 10 iterations. The table shows the probability of success P_{succ} in obtaining the correct class, the probability $P_{\text{error}}(1|2)$ of predicting class 1 when the true class is 2 and the probability $P_{\text{error}}(2|1)$ of predicting class 2 when the true class is 1.

Note that the probability P_{succ} is independent of the parameter ω because of the post-selection procedure. Nevertheless, ω still shapes the system's dynamics: it controls the convergence rate and, in the long-time regime, the population accumulated at the final node.

Hence, although ω does not change the classification outcome, it does impact the runtime of the model by modulating how quickly the computation reaches the post-selected event.

Table 1. Success probability and conditional error rates of the classifier as a function of ω . Here, P_{succ} is the probability of predicting the correct class; $P_{\text{error}}(1|2)$ is the probability of predicting class 1 when the true class is 2 and $P_{\text{error}}(2|1)$ vice versa.

ω	P_{succ}	$P_{\text{error}}(1 2)$	$P_{\text{error}}(2 1)$
1	91.3%	7.9%	0.8%
0.8	89.5%	0.5%	1.0%
0.5	90.4%	8.4%	1.2%

5. Conclusions

We have demonstrated that the quantum distance-based classifier can be faithfully implemented within the open quantum walk (QW) model for dissipative quantum computation and that its performance is independent of the environmental parameters ω and λ . By deriving analytical expressions for the classifier's correctness and post-selection success probability, we provided a rigorous connection between the quantum and classical distance-based frameworks. These results show that QWs can serve as a viable platform to realize basic quantum machine learning algorithms in a dissipative setting.

Our simulations confirm that the QW implementation is feasible and reproduces the expected classification behavior for two-feature datasets, such as the Iris benchmark, thereby providing the proof of principle we originally aimed to demonstrate. However, it is important to emphasize that we did not modify the algorithm; rather, we embedded it in the QW model in order to demonstrate its feasibility. It is also important to note that although there are implementations of algorithms in the QW framework, such as the Fourier transform [2] and Page-Rank [5], the literature on such implementations is still scarce. We do not compare our results with other implementations of this algorithm, such as Schuld's [18], because Schuld's approach is circuit-based, whereas in our model the unitaries are implemented via interaction with the environment. A direct comparison is therefore not entirely fair, since the two settings correspond to different computational models.

Finally, although we have employed a restricted QW model in which all Kraus operators are proportional to unitaries, extending this framework to include non-unitary Kraus operators may enable more efficient dissipative algorithms. These directions represent promising avenues for future theoretical and experimental research.

Author Contributions: Conceptualization, G.P., F.P. and N.K.B.; Formal analysis, P.L.M.; Investigation, G.P. and N.K.B.; Writing—original draft, P.L.M.; Writing—review & editing, G.P., F.P. and N.K.B.; Supervision, N.K.B.; Project administration, N.K.B. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brazil (CAPES)—Finance Code 001. N.K.B. acknowledges financial support from CNPq Brazil (442429/2023-1) and FAPESP (Grant 2021/06035-0). The authors acknowledge the financial support of the National Institute of Science and Technology for Applied Quantum Computing through CNPq process No. 408884/2024-0.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

OQW Open Quantum Walk

Appendix A

Here we show, for completeness, that Equation (42) follows from Equation (41). We first prove that, for any $x \in \mathbb{R}$ for which the expressions are well defined,

$$\frac{1 - \tan(x/2)}{1 + \tan(x/2)} = \frac{1 - \sin x}{\cos x}. \quad (\text{A1})$$

Indeed,

$$\begin{aligned} \frac{1 - \sin x}{\cos x} &= \frac{1 - \sin(2 \cdot \frac{x}{2})}{\cos(2 \cdot \frac{x}{2})} \\ &= \frac{1 - 2 \sin(\frac{x}{2}) \cos(\frac{x}{2})}{\cos^2(\frac{x}{2}) - \sin^2(\frac{x}{2})} \\ &= \frac{\sec^2(\frac{x}{2}) - 2 \tan(\frac{x}{2})}{1 - \tan^2(\frac{x}{2})} \\ &= \frac{1 + \tan^2(\frac{x}{2}) - 2 \tan(\frac{x}{2})}{1 - \tan^2(\frac{x}{2})} \\ &= \frac{(1 - \tan(\frac{x}{2}))^2}{1 - \tan^2(\frac{x}{2})} \\ &= \frac{1 - \tan(\frac{x}{2})}{1 + \tan(\frac{x}{2})}. \end{aligned} \quad (\text{A2})$$

The second equality uses the double-angle formulas for sin and cos; the third is obtained by multiplying numerator and denominator by $\sec^2(x/2)$; the fourth uses $\sec^2(x/2) = 1 + \tan^2(x/2)$; and the remaining steps follow from straightforward algebra. Therefore, Equation (41) implies

$$\frac{1 - \sin(\omega'/2)}{\cos(\omega'/2)} = \frac{1 - \tan(\omega'/4)}{1 + \tan(\omega'/4)} = \sqrt{t}, \quad (\text{A3})$$

from which we may isolate $\tan(\omega'/4)$ to obtain

$$\tan(\omega'/4) = \frac{1 - \sqrt{t}}{1 + \sqrt{t}}. \quad (\text{A4})$$

This is equivalent to Equation (42) (up to the choice of branch and 2π periodicity).

References

1. Verstraete, F.; Wolf, M.M.; Cirac, J.I. Quantum computation and quantum-state engineering driven by dissipation. *Nat. Phys.* **2009**, *5*, 633–636. [\[CrossRef\]](#)
2. Sinayskiy, I.; Petruccione, F. Efficiency of open quantum walk implementation of dissipative quantum computing algorithms. *Quantum Inf. Process.* **2012**, *11*, 1301–1309. [\[CrossRef\]](#)
3. Marshall, J.; Venuti, L.C.; Zanardi, P. Classifying quantum data by dissipation. *Phys. Rev. A* **2019**, *99*, 032330. [\[CrossRef\]](#)
4. Attal, S.; Petruccione, F.; Sabot, C.; Sinayskiy, I. Open Quantum Random Walks. *J. Stat. Phys.* **2012**, *147*, 832–852. [\[CrossRef\]](#)
5. Dutta, S. Discrete-time open quantum walks for vertex ranking in graphs. *Phys. Rev. E* **2025**, *111*, 034312. [\[CrossRef\]](#) [\[PubMed\]](#)

6. Lardizabal, C.F.; Souza, R.R. On a Class of Quantum Channels, Open Random Walks and Recurrence. *J. Stat. Phys.* **2015**, *159*, 772–796. [\[CrossRef\]](#)
7. Maciel, P.L.; Bernardes, N.K. Dynamics and computation in linear open quantum walks. *Phys. Rev. A* **2025**, *112*, 032627. [\[CrossRef\]](#)
8. Konno, N.; Yoo, H.J. Limit Theorems for Open Quantum Random Walks. *J. Stat. Phys.* **2013**, *150*, 299–319. [\[CrossRef\]](#)
9. Attal, S.; Guillotin-Plantard, N.; Sabot, C. Central Limit Theorems for Open Quantum Random Walks and Quantum Measurement Records. *Ann. Henri Poincaré* **2015**, *16*, 15–43. [\[CrossRef\]](#)
10. Lloyd, S.; Garnerone, S.; Zanardi, P. Quantum algorithms for topological and geometric analysis of data. *Nat. Commun.* **2016**, *7*, 10138. [\[CrossRef\]](#)
11. Schuld, M.; Killoran, N. Quantum machine learning in feature Hilbert spaces. *Phys. Rev. Lett.* **2019**, *122*, 040504. [\[CrossRef\]](#)
12. Cong, I.; Choi, S.; Lukin, M.D. Quantum convolutional neural networks. *Nat. Phys.* **2019**, *15*, 1273–1278. [\[CrossRef\]](#)
13. Korkmaz, U.; Türkpençe, D.; Akıncı, T.Ç.; Şeker, S. A thermal quantum classifier. *Quantum Inf. Comput.* **2020**, *20*, 969–986. [\[CrossRef\]](#)
14. Brito, E.B.; de Paula Neto, F.M.; Bernardes, N.K. Quantum classifier based on open quantum systems with amplitude information loading. *Quantum Inf. Process.* **2024**, *23*, 339. [\[CrossRef\]](#)
15. Blank, C.; Park, D.K.; Rhee, J.K.K.; Petruccione, F. Quantum classifier with tailored quantum kernel. *npj Quantum Inf.* **2020**, *6*, 41. [\[CrossRef\]](#)
16. Pérez-Salinas, A.; Cervera-Lierta, A.; Gil-Fuster, E.; Latorre, J.I. Data re-uploading for a universal quantum classifier. *Quantum* **2020**, *4*, 226. [\[CrossRef\]](#)
17. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer: New York, NY, USA, 2009. [\[CrossRef\]](#)
18. Schuld, M.; Fingerhuth, M.; Petruccione, F. Implementing a distance-based classifier with a quantum interference circuit. *EPL* **2017**, *119*, 60002. [\[CrossRef\]](#)
19. Neumann, N.M.P. Classification Using a Two-Qubit Quantum Chip. In *Computational Science—ICCS 2021; Lecture Notes in Computer Science*; Paszynski, M., Kranzlmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A., Eds.; Springer: Cham, Switzerland, 2021; Volume 12747. [\[CrossRef\]](#)
20. Turkpence, D.; Akıncı, T.Ç.; Şeker, S. A steady state quantum classifier. *Phys. Lett. A* **2019**, *383*, 1410–1418. [\[CrossRef\]](#)
21. Zhang, J.; Li, Z.; He, R.; Zhang, J.; Wang, B.; Li, Z.; Niu, T. Interactive Quantum Classifier Inspired by Quantum Open System Theory. In *Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; IEEE: New York, NY, USA, 2021; pp. 1–7. [CrossRef]*
22. Venegas-Andraca, S.E. Quantum walks: A comprehensive review. *Quantum Inf. Process.* **2012**, *11*, 1015–1106. [\[CrossRef\]](#)
23. Aharonov, Y.; Davidovich, L.; Zagury, N. Quantum random walks. *Phys. Rev. A* **1993**, *48*, 1687. [\[CrossRef\]](#)
24. Portugal, R. *Quantum Walks and Search Algorithms*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2018.
25. Childs, A.M. Universal Computation by Quantum Walk. *Phys. Rev. Lett.* **2009**, *102*, 180501. [\[CrossRef\]](#)
26. Lovett, N.B.; Cooper, S.; Everitt, M.; Trevers, M.; Kendon, V. Universal quantum computation using the discrete-time quantum walk. *Phys. Rev. A* **2010**, *81*, 042330. [\[CrossRef\]](#)
27. Breuer, H.; Petruccione, F. *The Theory of Open Quantum Systems*; Oxford Academic: Oxford, UK, 2010.
28. Behrends, E. *Introduction to Markov Chains*; Springer: Cham, Switzerland, 2000.
29. Kampen, N.G.V. *Stochastic Processes in Physics and Chemistry*, 3rd ed.; North-Holland: Amsterdam, The Netherlands, 2007.
30. Gardiner, C.W. *Handbook of Stochastic Methods*, 2nd ed.; Springer: Cham, Switzerland, 1985.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.