Quantum walks on graphs representing the firing patterns of a quantum neural network

Maria Schuld,* Ilya Sinayskiy, and Francesco Petruccione
Quantum Research Group, School of Chemistry and Physics,
University of KwaZulu-Natal Durban, KwaZulu-Natal, 4001, South Africa
and National Institute for Theoretical Physics (NITheP), KwaZulu-Natal, 4001, South Africa
(Dated: April 2, 2014)

Quantum walks have been shown to be fruitful tools in analysing the dynamic properties of quantum systems. This article proposes to use quantum walks as an approach to Quantum Neural Networks (QNNs). QNNs replace binary McCulloch-Pitts neurons with a qubit in order to use the advantages of quantum computing in neural networks. A quantum walk on the firing states of such a QNN is supposed to simulate central properties of the dynamics of classical neural networks, such as associative memory. It is shown that a biased discrete Hadamard walk derived from the updating process of a biological neuron does not lead to a unitary walk. However, a Stochastic Quantum Walk between the global firing states of a QNN can be constructed and it is shown that it contains the feature of associative memory. The quantum contribution to the walk accounts for a modest speed-up in some regimes.

PACS numbers: 03.67.Lx, 03.65.Yz, 87.18.Sn

I. INTRODUCTION

Quantum walks, the quantum equivalent of classical random walks, became a booming research field in the last decade [1–3]. Based on the theory of Markov chains, classical random walks study the evolution of the probability distribution of an abstract walker's position on a graph. The positions or vertices of the graph are connected by edges symbolising transition probabilities. In each step, the walker makes a random decision (often described by a coin toss) to which adjacent position to jump. Quantum walks, in which the walker's position on the graph can be a superposition and the decision process is simulated by a 'quantum coin' such as the Hadamard transformation, show a surprisingly different behaviour to classical random walks. Quantum walks have been formulated as discrete [4] or continuous [5] walks, and led to new versions such as the semi-classical Stochastic Quantum Walk [6] or the Open Quantum Walk [7].

The potential of quantum walks is based on their fruitful application for quantum computing just like classical walks lead to efficient classical algorithms [8, 9]. An important application of quantum walks has been found in the newly emerging field of quantum biology [10]. Evidence suggests that photosynthetic plants use nontrivial quantum effects such as superposition and interference for energy transport in their light-harvesting complexes [11, 12]. The trajectory of an excitation 'jumping' between molecular 'sites' from the antenna to the reaction center can thereby be modeled with the

formalism of quantum walks [13–16].

The success story of quantum biology is a motivation to reaccess questions of quantum dynamics in another important biological system: the brain. In 2006, Christof Koch and Klaus Hepp wrote in their *Nature* contribution entitled 'Quantum Mechanics in the Brain' [17]: "The critical question [...] is whether any components of the nervous system - a 300° Kelvin tissue strongly coupled to its environment - display macroscopic quantum behaviours, such as quantum entanglement, that are key to the brain's function." Besides a number of controversial theories on the 'quantum brain' [18, 19], there has been no evidence for nontrivial quantum effects in the nervous system yet. On the contrary, the macroscopic nature of signal transmission between neural cells seems to make quantum coherence impossible [20]. However, the intersection of neuroscience and quantum physics has been accessed from the perspective of computational science. In the last two decades, various Quantum Neural Network (QNN) models [21–28] have been proposed. Although not claiming to be realistic quantum models of biological neural networks, these proposals explore alternative ways of computation using both the advantages of quantum computing and neural computing.

This article follows the perspective of QNN research and investigates a new approach to introduce quantum physics into neural networks by making use of the theory of quantum walks. The sites of the quantum walk symbolise the firing patterns of a neural network consisting of simplified binary neurons with the states 'active' and 'resting'. A firing pattern is given by a binary string encoding which neuron of a network is firing and which is resting. The current position of the walker represents the network's firing state. A quantum

^{*} schuld@ukzn.ac.za

walker is of course able to be in a superposition of firing patterns. We show that a discrete quantum walk, in which a Hadamard-like biased coin is successively flipped to decide on the firing state of single neurons clashes with the framework of unitary quantum walks. To simulate a neural network's dissipative dynamics, we therefore need to focus on quantum walks that incorporate decoherence. A continuous Stochastic Quantum Walk on the hypercube, obtaining basic features of the brain's property of associative memory (i.e. retrieving a memorised pattern upon an imperfect initial pattern), is implemented and analysed. It can be shown that under certain conditions, the quantum part of the walk accounts for a modest speed-up of the walk. results serve as an example of the application of quantum walks to obtain specific dynamics of a quantum system.

The paper has the following structure: Section II and III very briefly introduce into the necessary theoretical background of quantum walks as well as Quantum Neural Networks. Section IV gives an idea of how quantum walks can be constructed in the context of QNNs, and explains the reason for the failure of the most intuitive way. A more mature approach is presented. The conclusion (Section V) offers a discussion including a way forward for the use of quantum walks in QNN research.

II. A BRIEF INTRODUCTION TO QUANTUM WALKS

On any graph of n vertices a Markov chain can be defined. A Markov chain is a sequence of events that is governed by a stochastic process in which the results of a timestep only depend on the results of the previous timestep [29]. Markov chains are described by a stochastic matrix $M(n \times n, \mathbb{R})$ with $\sum_{j=1}^{n} m_{ij} = 1$ and entries m_{ij} representing the weight of the directed edge going from vertex i to j (see Fig. 1). These weights can be interpreted as a transition rate from site i to j. Repeatedly applying M to a n-dimensional stochastic vector $\vec{\pi}$ with $\sum_{l=1}^{n} \pi_{l} = 1$ evolves an initial probability distribution through discrete time steps. The probability of being at vertex i changes according to [5]

$$\frac{d\pi_i}{dt} = -\sum_j M_{ij}\pi_j(t). \tag{1}$$

Markov chains on regular undirected graphs result in a stationary probability distribution π_s which is independent of the initial state [4]. The time it takes to reach the stable distribution is called the *mixing time*.

Markov chains with equal probability to jump from site i to any of the d sites adjacent to i are also known as random walks on a graph [30]. Random walks are based on the idea of an abstract walker who in each step tosses a d-dimensional coin to choose one of d possible

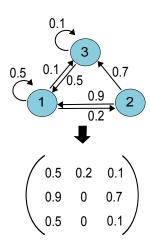


FIG. 1. (Colour online) Any weighed graph defines a Markov chain represented by a stochastic matrix.

directions at random. Random walks have been proven to be powerful tools in constructing efficient algorithms in computer science (for references see [8]).

In the quantum equivalent of random walks, a quantum walker walks between sites by changing its position state $|x\rangle \in \{|1\rangle,...,|n\rangle\}$. The difference to classical walks is twofold: First, the various paths are realised in a superposition and thus interfere with one another, and a measurement 'collapses' the paths into a current position. Second, the dynamics have to preserve the squared amplitude vector instead of the stochastic vector to preserve the total probability. This means that the evolution has to be unitary, or in the most general case of an open system, a completely positive trace preserving map [31, 32].

The unitarity of quantum walks furthermore implies that the evolution is reversible. Quantum walks therefore do not have a stationary probability distribution π_s as classical random walks do. However, it turns out that taking an average over the probability distribution over states $|x\rangle$,

$$\bar{P}_T(x) = \frac{1}{T} \sum_{t=0}^{T-1} |\langle x | \psi(t) \rangle|^2,$$
 (2)

leads to a stable distribution \bar{P}_s [4]. Quantum walks received a fair amount of attention and have been the topic of extensive reviews, books and attempts of implementations [1–3, 33, 34]. The reasons are that first, quantum walks show markedly different features than their classical counterparts. Second, quantum walks were in some cases able to outperform classical walks [5, 9, 31, 35, 36].

Two versions of quantum walks were established and exist in parallel: the discrete [4] and continuous

time quantum walk [5, 37]. The bridge between the two has finally been shown in [38]. An important development was also the exploration of decoherence in quantum walks [2]. Recently, an interesting version of the continuous quantum walk with decoherence has been introduced [6]. So called Stochastic Quantum Walks obey a Gorini-Kossakowski-Lindblad-Sudarshan (GKLS) type master equation [39, 40] that consists of a coherent as well as an incoherent part. These three versions will be important in the application further down and shall therefore be briefly presented.

Discrete quantum walks

In a discrete quantum walk, the 'walker' is associated with a wave function describing a quantum system with states $|\psi\rangle = |c\rangle \otimes |i\rangle \in \mathcal{H}_c \otimes \mathcal{H}_i$. The Hilbert space \mathcal{H}_i has a basis $\{|0\rangle,...,|n\rangle\}$ (n may be countable infinite) that represent sites or vertices on which the walk takes place. The Hilbert space \mathcal{H}_c with basis $|1\rangle,...,|d_i\rangle$ is a 'coin' space that denotes the current state of a coin 'tossed' to decide which direction to take next. Note that usually only regular graphs are considered and d is independent of the current position $|i\rangle$. The discrete walk then follows two substeps in each step $t \to t+1$, executed by a coin and a shift operator:

$$|\psi_{t+1}\rangle = \hat{S}(\hat{C}\otimes 1) |\psi_t\rangle.$$

First, the coin is 'tossed' by applying \hat{C} to the coin space. By that, the coin state is put into a superposition. Second, the conditional shift operator \hat{S} shifts the state to the r'th adjacent site if the outcome of the coin is $|r\rangle$ $(r \in \{1, ..., d\})$.

The most well-known coin is the Hadamard transformation that works on the two-dimensional basis $\{|a\rangle\,,|b\rangle\}$ as

$$\hat{H}|a\rangle = \frac{1}{\sqrt{2}}(|a\rangle + |b\rangle), \quad \hat{H}|b\rangle = \frac{1}{\sqrt{2}}(|b\rangle - |a\rangle).$$
 (3)

The minus in the second equation indicates that even the Hadamard transformation is not completely unbiased and denotes at the same time the fundamental difference to classical walks, as it is the source of interference [41].

Continous quantum walks

In 2001, Andrew Childs, Edward Farhi and Sam Gutman published a continuous version of the quantum walk. Their idea is based on the equivalence between Eq. (1) and the Schrödinger equation for state $|\psi(t)\rangle$

$$i\frac{d}{dt}\langle i| \psi(t)\rangle = \sum_{j}\langle i| H| j\rangle\langle j| \psi(t)\rangle \tag{4}$$

While Eq. (1) preserves $\sum_{l=1}^{n} \pi_l = 1$, the Schrödinger equation (4) makes sure that $\sum_{i} |\langle i| \psi(t) \rangle|^2 = 1$ is fulfilled. The difference between the two evolutions is simply the imaginary unit in the latter [5]. Comparing

both equations, one can see that the stochastic matrix M of the classical Markov chain is replaced by the Hamiltonian H of the quantum system. H consequently equals the weighed adjacency matrix of the graph. To obtain a hermitian and thus symmetric operator (as the entries are real numbers), the graph needs to be undirected.

Stochastic Quantum Walks

A number of contributions investigate what happens if decoherence is introduced into quantum walks [2, 42]. Decoherence destroys the quantum property of coherent states and drives the dynamics into the classical regime. In some cases this can lead to preferred dynamics [2]. An interesting proposal for a decohered continuous quantum walk has recently been introduced [6]. A so called Stochastic Quantum Walk (SQW) evolves a density matrix according to the GKLS master equation [39, 40]

$$\dot{\rho} = -i\kappa [H, \rho] - \gamma \sum_{k} \left(\frac{1}{2} L_{k}^{\dagger} L_{k} \rho + \frac{1}{2} \rho L_{k}^{\dagger} L_{k} - L_{k} \rho L_{k}^{\dagger} \right). \tag{5}$$

Note that here and in the following, \hbar is set to one. The parameters κ and γ define the influence of the two terms on the right side. The sum term describes the stochastic evolution, in which L_k denote Lindblad jump operators that decohere the quantum state. The first term is the usual Schrödinger quantum evolution as known from the von Neumann equation. The Hamiltonian represents a (weighed) adjacency matrix as in the continuous walk given in Eq. (4). In this fashion, the "evolution among vertices happens through coherences developed by a Hamiltonian" [6] while the system is constantly exposed to decoherence. Thus, both advantages of dissipation and coherence can be used and transitions from one to the other studied. We will implement a version of the Stochastic Quantum Walk in Section IVb.

III. NEURAL NETWORKS AND THE CONCEPT OF THE 'QURON'

To get an understanding of what is meant by the 'firing patterns of a QNN', we need to briefly introduce into some fundamentals of computational neuroscience as well as the basic concept of a QNN.

Neural networks are computational systems inspired by the biological neural networks forming our brain. The brain is believed to compute information by carrying electric signals, so called action potentials, along the membranes of interconnected neural cells [43, 44]. The algorithmic dynamics of biological neural networks are defined by the connection strengths between neurons as well as by the activation function of a neuron due to the input signals from other neurons. Information is encoded in the global firing pattern of the neural network.

It turns out that important properties of the brain,

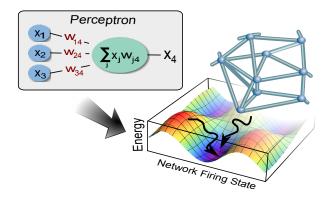


FIG. 2. (Colour online) Illustration of a perceptron, a mathematical model of the neural activation mechanism with neurons x_1, x_2, x_3, x_4 and connection strengths w_{14}, w_{24}, w_{34} . In a recurrent Hopfield network, neurons with a perceptron activiation function are mutually connected as depicted in the graph structure. Such a network that stores firing patterns as minima of an energy function.

such as the computation of incomplete or imperfect input, can be retrieved by the easiest model of a neuron possible, introduced by McCulloch and Pitts as early as 1943 [45]: an active neuron firing a sequence of action potentials in a given time is represented by a '1' while a resting neuron is represented by a '0'. (for other types of neural networks, see [46]). The N neurons of a neural network can thus be described by variables $x_i \in \{0,1\}, i=1,...,N$. Each neuron x_i is assigned to a characteristic threshold θ_i . The biologically derived activation or updating function of a neuron x_i is then given by

$$x_i = \begin{cases} 1, & \text{if } \sum_{i \neq j=1}^N w_{ji} x_j \leq \theta_i, \\ 0, & \text{else,} \end{cases}$$
 (6)

where the w_{ij} , i, j = 1, ..., N are real numbers denoting the strength of the connection between neuron x_i and x_j . The according setup is called a 'perceptron' (see Fig. 2). The vector $(x_1, ..., x_N)$ is called the 'state' or firing pattern of the network. Initialising the network means to set each neuron to either 1 or 0. An update step of the global network state can either happen through a synchronous update of all neurons, through a chronological or random sequence of individual updates according to Eq. (6).

One of the milestones in artificial neural network research was John J. Hopfield's 1982 publication on a network today widely known as 'Hopfield network' [47] in which the connection strengths fulfill

$$w_{ij} = w_{ji}, \qquad w_{ii} = 0.$$

Although of a simple setup, the Hopfield model shows the powerful feature of associative memory. Associative memory is the ability to – out of a number of stored firing network states – retrieve the network state that is in the center of the dynamic basin of attraction for the input pattern. Hopfield networks thus store firing patterns as dynamic attractors. These dynamic attractors are minima of the Ising-type energy function

$$E(x_1, ..., x_N) = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} x_i x_j + \sum_{i=1}^{N} \theta_i x_i.$$
 (7)

A Hopfield network always inherits attractors from the nonlinearity of the updating process. The specific dynamics of a neural network are then solely defined by the choice of weights w_{ij} . The property $w_{ii} = 0$ makes sure that all attractors are stable states (as opposed to limit cycles of alternating states) [48]. After a finite number of updating steps, any initial state of the network will consequently end up in the 'closest' attracting network state which is then reproduced by the updating process. An important implication is that neural networks based on a step activation function do manipulate information irreversibly due to its injectivity. i.e. a state of a neural network at timestep t_{n-1} cannot be reconstructed from its state at timestep t_n . This might be different for other types of activation functions such as the sigmoid function. It is also interesting to note that in conventional neural networks, the number of neural excitations (i.e. of neurons with the state 'active') is not conserved, which is crucial for the attempt to construct quantum walks of excitations in Quantum Neural Network.

Approaches to QNNs [23, 27, 49–54] are mostly based on Hopfield-type neural networks. The basic idea of introducing quantum properties is to replace the McCulloch-Pitts neuron $x=\{0,1\}$ by a 'qubit neuron' $|x\rangle$ of the two-dimensional Hilbert space \mathcal{H}^2 with basis $\{|0\rangle\,,|1\rangle\}$. We propose to simply call this object a 'quron'. The central property of a quron is that it can be in a superposition of its two firing states with the complex amplitudes α,β

$$|x\rangle = \alpha |0\rangle + \beta |1\rangle, \quad |\alpha|^2 + |\beta|^2 = 1.$$
 (8)

The state of a network with N qurons thus becomes a quantum product state of the 2^N -dimensional Hilbert space $\mathcal{H}^{2^N} = \mathcal{H}^2_{(1)} \otimes \cdots \otimes \mathcal{H}^2_{(N)}$

$$|\psi\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_N\rangle = |x_1x_2\dots x_N\rangle.$$

These are the firing states of a QNN on which a quantum walk will be constructed in the following section.

IV. QUANTUM WALKS BETWEEN QUANTUM NEURAL NETWORK STATES

The genius of the Hopfield model lies in the fact that operations on the neuron as a local unit impose dynamics on the global network state. These global dynamics

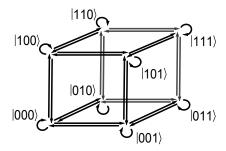


FIG. 3. The graph of a quantum walk on the firing states of a QNN has the structure of a hypercube, where the firing patterns sit at the corners.

can be understood as a classical random walk between network states,

$$(x_1,...,x_n)_{t_0} \to (x_1,...,x_n)_{t_1} \to (x_1,...,x_n)_{t_2} \to ...,$$

beginning with the initial pattern and in each step jumping to the updated network state. After a finite number of steps, the chain reproduces the stable state serving as the output of the algorithm.

Likewise, a quantum walk on the firing states of a QNN can be defined as an evolution

$$|\psi\rangle_{t_0} \to |\psi\rangle_{t_1} \to |\psi\rangle_{t_2} \to \dots,$$

following the laws of quantum mechanics. An important question is the structure of the graph underlying such a quantum walk. The vertices of the graph are given by binary strings denoting all possible firing patterns. The connectivity thus depends on the updating protocol. If all neurons are updated synchronously, each transition between firing patterns is theoretically possible and the graph is fully connected. We will concentrate on the more common case of updating single neurons at a time giving rise to the hypercube given in Fig. 3 for the N=3 dimensional case. The hypercube connects binary strings that differ only in one bitflip (in other words, they have a Hamming distance of one [56]). We add self-connections of every site to account for updates that leave the firing pattern unchanged.

The remaining part of this article will mention a very intuitive way of implementing a quantum walk on a QNN by tossing a quantum coin to decide upon the updated state of each quron, and show why it fails to lead to a discrete, unitary quantum walk. It will then present a version of a Stochastic Quantum Walk that simulates an associative memory and discuss the results.

A. Why the most intuitive version of a discrete quantum walk fails

A straight forward way to construct a QNN seems to be to replace the updating process of a neuron given in Eq. (6) by a biased Hadamard-like transformation on a two-dimensional coin state $|c\rangle = \{|0\rangle_c\,, |1\rangle_c\}$ and to flip the state of the quron depending on the outcome of the coin as done in the discrete quantum walk. To retrieve nontrivial dynamics, we would want a biased coin leading to a superposition shown in Eq. (8) in which the amplitudes $|\alpha|^2$ and $|\beta|^2$ encode a probability of the corresponding neuron to fire or to rest. This can be done by defining the firing probability $0 \le p_i \le 1$ of a neuron x_i as

$$p_i = \frac{\sum_j w_{ij} x_j + (N-1)}{2(N-1)},\tag{9}$$

and choosing

$$\alpha_i = \sqrt{1 - p_i}, \qquad \beta_i = \pm \sqrt{p_i}$$

The \pm in front of β is introduced to simulate the quantum properties of the Hadamard transformation or in other words, to introduce interference. The firing probability is nothing else than the normalised summed up signal coming from all input neurons to an output neuron. Since we sum over N-1 weighed neurons $w_{ij}x_i \in [-1,1]$, the incoming signal lies in the interval [-(N-1), (N-1)]. To obtain a positive value normalised to [0,1] representing the probability, we consequently need to 'shift' the signal to positive values and divide by the range of the interval, 2(N-1). Thus, if the incoming signal is strong, the probability for the neuron to become active is high, regardless of its prior state. Note that the thresholds θ_i of classical neurons are set to zero when dealing with gurons in the following. The updating process for guron $|x_i\rangle$ can consequently be formulated as a transformation

$$\begin{split} \hat{H} & |0\rangle_c = \sqrt{1 - p_i} & |0\rangle_c + \sqrt{p_i} |1\rangle_c \,, \\ & \hat{H} & |1\rangle_c = \sqrt{1 - p_i} & |0\rangle_c - \sqrt{p_i} |1\rangle_c \,. \end{split} \tag{10}$$

This is slightly different from the biased Hadamard transformation used in biased quantum walks [2, 57–59],

$$\hat{H} |0\rangle_{c} = \sqrt{p_{i}} |0\rangle_{c} + \sqrt{1 - p_{i}} |1\rangle_{c},$$

$$\hat{H} |1\rangle_{c} = \sqrt{1 - p_{i}} |0\rangle_{c} - \sqrt{p_{i}} |1\rangle_{c}, \quad (11)$$

where the variable p_i denotes the probability that the coin state flips its value, so that the biased Hadamard coin depends on the history of the coin state. This small difference leads to a problem in the implementation of the quantum walk proposed here, since the coin (Eq. 10) is nonunitary. In fact, this property is not surprising since it stems from the dissipative dynamics of the Hopfield network, in which information of the former state of the neuron does not feed into the updating process (due to $w_{ii} = 0$). A direct application of coherent quantum walks onto neural dynamics is thus not trivial. As a conclusion, quantum walks including decoherence must be considered to incorporate dissipation.

B. A Stochastic Quantum Walk on the hypercube

We want to propose another type of quantum walk that is not derived from the neural updating mechanism, but still obtains the Hopfield network's dynamics of associative memory. Hence, our goal is to introduce a Stochastic Quantum Walk on a hypercube graph that ends up in one of two 'attracting firing states' closer to the initial state in terms of Hamming distance.

The hypercube of dimension N is given by a set of vertices \mathcal{V}^{2^N} as 'corners', representing the density matrices $|x_1,...,x_N\rangle_i\langle x_1,...,x_N|=|i\rangle\langle i|,i=1,...,2^N$ (compare Fig. 4). The quantum state $|x_1,...,x_N\rangle_i$ is thereby the i-th basis state of a QNN of 2-level qurons as introduced above, and the shorthand $|i\rangle$ is used to reduce notation. In the hypercube, two vertices $|i\rangle\langle i|$ and $|j\rangle\langle j|$ are connected by an edge if their respective network states differ by one quron's state or not at all, or in other words, the Hamming distance $d_H(i,j)$ between the two states $|i\rangle$ and $|j\rangle$ is one or zero. The hypercube graph's adjacency matrix is consequently given by

$$H_{ij} = \begin{cases} a_{ij}, & \text{if } d_H(i,j) = 0, 1, \\ 0, & \text{else.} \end{cases}.$$

For now we set $a_{ij} = 1$ [60]. We introduce sinks by removing the edges leading to/from the vertices that represent the patterns we want to memorise. For simplicity we shall consider the example of only two 'sink states', but this case can easily be generalised. Removing the edges is necessary since once the walker arrived at a sink, it is supposed to be trapped with no possibility to leave. Since in continuous quantum walks the adjacency matrix is equivalent to the hamiltonian, the graph structure needs to be undirected to ensure the hermiticity of the hamiltonian. This is why by removing the edges leading out of the sinks, we have to remove the edges leading to the sinks at the same time. The graph structure of the coherent part of the stochastic quantum walk is sketched in Fig. 4 on the left.

The dissipative part of the GKLS master equation (5) can be written with the help of jump operators L_k . We use these jump operators to account for the 'directed' part of the walk. Each edge $i \leftrightarrow j$ between vertices $|i\rangle \langle i|$ and $|j\rangle \langle j|$ is assigned with a jump operator $L_k = L_{i \to j} = |j\rangle \langle i|$ where $|j\rangle \langle j|$ is the vertex closer or equal to a sink state. If both vertices sharing an edge have the same distance to a sink, no jump operator is attributed to that edge. This setup creates a flow to the sink states in the dissipative part and builds a 'bridge' between the graph structure of the coherent part to the disconnected sinks. The graph structure of the dissipative part of the stochastic quantum walk is sketched in Fig. 4 on the right.

The resulting master equation for the stochastic quantum walk with the two sink states $|l\rangle \langle l|, |m\rangle \langle m|$ is then

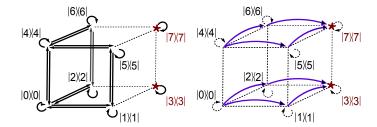


FIG. 4. (Colour online) Illustration of the construction of the graph for the Stochastic Quantum Walk described in the text. The left figure represents the coherent part, using a Hamiltonian derived from the graph's adjacency matrix. The sinks ($|7\rangle\langle7|$ and $|3\rangle\langle3|$, in red) are isolated (dashed lines indicate no connection). The right figure represents the decoherent part of the walk. Lindblad jump operators (long arrows, in purple) simulate attraction by introducing a flow towards the sinks.

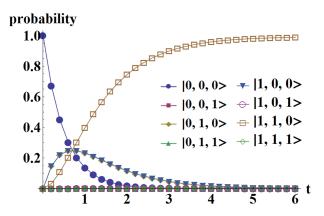


FIG. 5. (Colour online) Example of the evolution of the QNN firing states' probabilities in the Stochastic Quantum Walk on the hypercube of dimension N=3 as introduced here; with sinks at $|101\rangle\langle101|$ and $|111\rangle\langle111|$, initial state $|000\rangle\langle000|$ and $\kappa=\gamma=1$. After only two time units, the walker has a high probability to be in the desired output state. Note that time is in inverse units of γ .

given by Eq. (5) with

$$\begin{split} H &= \sum_{\langle i,j> \neq l,m} a_{ij} \left| i \right\rangle \left\langle j \right|, \\ L_k &= L_{i \to j} = \left| j \right\rangle \left\langle i \right|, \end{split}$$

where $\langle i,j \rangle = \{|i\rangle \langle i|, |j\rangle \langle j| \in \mathcal{V}^{2^N} | d_H(i,j) = 1\}$ is a pair of connected vertices, $i \to j = \{|i\rangle \langle i|, |j\rangle \langle j| \in \mathcal{V}^{2^N} | \min[d_H(j,l), d_H(j,m)] \leq \min[d_H(i,l), d_H(i,m)]\}$ denotes a pair of connected vertices in which $|j\rangle \langle j|$ is the vertex closer or equal to a sink state, and $a_{ij} = a_{ji}$.

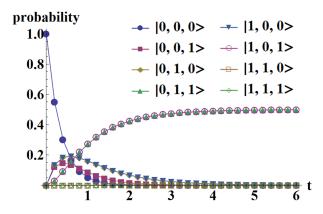


FIG. 6. (Colour online) In this example, the sink states are given by $|011\rangle\langle011|$ and $|101\rangle\langle101|$, the initial state is $|000\rangle\langle000|$ and $\kappa=\gamma=1$. Since both sink states have the same Hamming distance to the initial state, the walker has a probability of almost 0.5 to end up in either of the respective sink states. Note that time is in inverse units of γ .

C. Results

The Stochastic Quantum Walk starts at the vertex representing the initial firing pattern and propagates over the hypercube. After a time evolution of the magnitude of several time units, the walk always finds the sink closest to the initial state in terms of Hamming distance with a probability of nearly 1 (Fig. 5). The model consequently shows the basic neural network feature of associative memory. If the two sinks have the same distance, the output is an equal probability of both sink states (Fig. 6). This is an optimisation to a classical, deterministic associative memory which favours one of two states of an equal Hamming distance to the initial states.

It turns out that the dynamics of the Stochastic Quantum Walk on the hypercube are mainly influenced by the incoherent part of Eq. (5). Figure 7 shows the time until the average probability distribution reaches the correct stable state (mixing time) in dependence of the two parameters κ and γ as given in Eq. (5). One can see that the time of reaching a stable distribution only depends on κ for small values of γ , which denotes the coherent or quantum part in the stochastic walk. However, for values $\gamma < 1$, the quantum part can increase the speed of the walk by several time units. Since the quantum walk has shown to traverse the hypercube exponentially faster than a classical walk [35], the contribution of the quantum speed-up might be larger in higher dimensions. Due to the exponential growth in the dimension of the Hamiltonian, simulations in dimensions ≥ 7 require large computational resources.

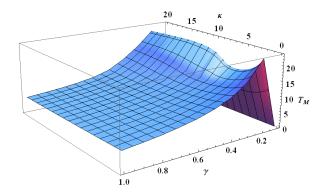


FIG. 7. (Colour online) Mixing time T_M in inverse units of γ to reach a steady state in dependence of the parameters κ and γ in the Stochastic Quantum Walk on the 4-dimensional hypercube with sink states $|1011\rangle\langle1011|,|1111\rangle\langle1111|$ and initial state $|0000\rangle\langle0000|$. The speed of the algorithm is almost independent of the quantum contribution to the walk (represented by parameter κ). However, for low values of γ , there is an optimal value of κ in terms of mixing time. Please note that if the Stochastic Quantum Walk did not show convergent behaviour, the mixing time was set to zero in order to indicate that the Quantum Associative memory did not retrieve the correct result.

V. CONCLUSIONS

This article studied some aspects of the application of quantum walks to Quantum Neural Networks. It was argued that a direct translation of the neural updating mechanism into a Hadamard-like transformation faces the problem of a nonunitary coin operator. This stems from the dissipative nature of the neural activation function and is symptomatic for the attempt to combine the attractor-like dynamics of neural networks and the linear, unitary dynamics of quantum objects. We concluded that decoherence needs to be introduced into the model. A Stochastic Quantum Walk on the hypercube was therefore constructed, and we could show its property of associative memory, an important feature of neural networks. Due to the low dependence on the quantum evolution or coherent part of the walk, this model is only under a limited perspective a candidate for a quantum walk on the firing states of a Quantum Neural Network. However, these results can be seen as a first attempt in this direction and serve as an example of the application of quantum walks to obtain certain algorithmic dynamics of quantum systems.

There are other versions of quantum walks that might be worth investigating in order to overcome the flaws presented by Stochastic Quantum Walks and coined quantum walks. For example, there are different ways to introduce decoherence into the dynamics, such as projective measurements on the coin [2, 61, 62]. In fact, measurements have been proposed by several authors searching for a QNN model to account for the nonlinear updating process of neurons in a quantum regime [49–52]. Another interesting version of quantum walks are the recently developed Open Quantum Walks [7, 63, 64]. Based on the theory of open quantum systems, Open Quantum Walks describe a walker whose internal degrees of freedom are interacting with an environment and influencing the walker's external degree of freedom. The formalism shows a striking analogy to the updating function of neurons, giving the advantage that it does not require the global coherence of QNN states as in the walks on network states investigated here. Open Quantum Walks might consequently be a natural candidate when studying possible dynamics of Quantum Neural Networks.

The underlying idea to this paper was to use the formalism of quantum walks in order to find a dynamic evolution of a Quantum Neural Network that optimises the computational properties of classical neural networks [55]. In a second step, the dynamic evolution could then be attempted to be attributed to physical processes, in the far picture possibly leading to a quantum model of biological neural networks. It can therefore be interesting to ask how the model can incorporate learning, a mechanism characteristic for neural networks. It is important to emphasize again that the nodes of the graph constructed in Fig. 4 do not represent qurons, but entire firing states of a Quantum Neural Network and the edges a_{ij} consequently do not correspond to the neural

weights w_{ij} , i, j = 1, ..., N. However, similar to Hopfield networks 'learning' a pattern by choosing appropriate weights that imprint the memory states into the energy function Eq. (7), choosing the connections a_{ij} defines the dynamics of the Quantum Associative Memory model presented here. In both cases, learning is static, i.e. done by the initial construction of the network (or the graph). To get a quantum model that includes dynamic learning it would be a fruitful perspective to construct a quantum walk that simulates the above mentioned feed-forward neural networks. Feed-forward networks are dynamically trained by so called backpropagation algorithms, where random initial weigths are repeatedly manipulated to minimise an error function comparing target outputs of a training set to real outputs calculated by the neural network [46]. Such a quantum walk would be required to reproduce feed-forward network's characteristica such as pattern recognition and could serve as an interesting continuation of the results found here.

ACKNOWLEDGEMENTS

This work is based upon research supported by the South African Research Chair Initiative of the Department of Science and Technology and National Research Foundation.

- [1] J. Kempe, Contemp. Phys. 44, 307 (2003).
- [2] V. Kendon, Math. Structures Comput. Sci. 17, 1169 (2007).
- [3] S. E. Venegas-Andraca, Quantum Inf. Processing 11, 1015 (2012).
- [4] J. K. Dorit Aharonov, Andris Ambainis and U. Vazirani, Proc. 33rd STOC (ACM), 5059 (2001), quant-ph/0012090.
- [5] A. M. Childs, E. Farhi, and S. Gutmann, Quantum Inf. Processing 1, 35 (2002).
- [6] J. D. Whitfield, C. A. Rodrí guez Rosario, and A. Aspuru-Guzik, Phys. Rev. A 81, 022323 (2010).
- [7] S. Attal, F. Petruccione, and I. Sinayskiy, Phys. Lett. A 376, 1545 (2012).
- [8] C. Moore and A. Russell, in Randomization and Approximation Techniques in Computer Science (Springer, 2002) pp. 164–178.
- [9] A. Ambainis, Int. J. Quantum Inf. 1, 507 (2003).
- [10] P. Ball, Nature **474**, 272 (2011).
- [11] G. S. Engel, T. R. Calhoun, E. L. Read, T.-K. Ahn, T. Mančal, Y.-C. Cheng, R. E. Blankenship, and G. R. Fleming, Nature 446, 782 (2007).
- [12] G. Panitchayangkoon, D. Hayes, K. A. Fransted, J. R. Caram, E. Harel, J. Wen, R. E. Blankenship, and G. S. Engel, Proc. Natl. Acad. Sci. 107, 12766 (2010).
- [13] A. Ishizaki and G. R. Fleming, Proc. Natl. Acad. Sci. 106, 17255 (2009).
- [14] S. Hoyer, M. Sarovar, and K. B. Whaley, New J. Phys. 12, 065041 (2010).

- [15] M. Mohseni, P. Rebentrost, S. Lloyd, and A. Aspuru-Guzik, J. Chem. Phys. 129, 174106 (2008).
- [16] P. Rebentrost, M. Mohseni, I. Kassal, S. Lloyd, and A. Aspuru-Guzik, New J. Phys. 11, 033003 (2009).
- [17] C. Koch and K. Hepp, Nature **440**, 611 (2006).
- [18] S. Hameroff and R. Penrose, Phys. Life Rev. 403 (2013).
- [19] W. J. Freeman and G. Vitiello, J. Phys. A 41, 304042 (2008).
- [20] M. Tegmark, Phys. Rev. E **61**, 4194 (2000).
- [21] M. Andrecut and M. Ali, Int. J. Mod. Phys. C 13, 75 (2002).
- [22] M. Altaisky, arXiv preprint quant-ph/0107012 (2001).
- [23] S. Gupta and R. Zia, J. Comput. System Sci. 63, 355 (2001).
- [24] E. C. Behrman, V. Chandrashekar, Z. Wang, C. Belur, J. E. Steck, and S. Skinner, arXiv preprint quantph/0202131 (2002).
- [25] L. Fei and Z. Baoyu, in Neural Networks and Signal Processing, 2003. Proceedings of the 2003 International Conference on, Vol. 1 (IEEE, 2003) pp. 539–542.
- [26] R. Zhou, H. Wang, Q. Wu, and Y. Shi, Int. J. Theor. Phys. 51, 705 (2012).
- [27] W. Oliveira, A. J. Silva, T. B. Ludermir, A. Leonel, W. R. Galindo, and J. C. Pereira, in Neural Networks, 2008. SBRN'08. 10th Brazilian Symposium on (IEEE, 2008) pp. 147–152.
- [28] G. Tóth, C. S. Lent, P. D. Tougaw, Y. Brazhnik, W. Weng, W. Porod, R.-W. Liu, and Y.-F. Huang, arXiv preprint cond-mat/0005038 (2000).

- [29] W. Feller, An introduction to probability theory and its applications (John Wiley & Sons, 2008).
- [30] Some authors extend the random walk model to biased probabilities [2, 57–59]. These so called 'biased random walks' are nothing else than Markov chains introduced here. We will use the more general definition.
- [31] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, arXiv:quant-ph/0012090 (2002).
- [32] M. A. Nielsen and I. L. Chuang, Quantum computation and quantum information (Cambridge university press, 2010).
- [33] J. Wang and K. Manouchehri, *Physical Implementation of Quantum Walks* (Springer, 2013).
- [34] B. C. Travaglione and G. J. Milburn, Phys. Rev. A 65, 032310 (2002).
- [35] J. Kempe, arXiv preprint quant-ph/0205083 (2002).
- [36] N. Shenvi, J. Kempe, and K. B. Whaley, Phys. Rev. A 67, 052307 (2003).
- [37] E. Farhi and S. Gutmann, Phys. Rev. A 58, 915 (1998).
- [38] F. W. Strauch, Phys. Rev. A **74**, 030301 (2006).
- [39] G. Lindblad, Commun. Math. Phys. 48, 119 (1976).
- [40] V. Gorini, A. Frigerio, M. Verri, A. Kossakowski, and E. Sudarshan, Rep. Math. Phys. 13, 149 (1978).
- [41] The reason why a separate coin space is introduced is that the unitarity condition is not fulfilled by most examples of walks (a detailed argumentation can be found in [9]).
- [42] T. A. Brun, H. A. Carteret, and A. Ambainis, Phys. Rev. A 67, 052317 (2003).
- [43] D. Purves, Neuroscience, 3rd ed. (Sinauer, 2008).
- [44] P. Dayan and L. F. Abbott, Theoretical neuroscience, Vol. 31 (MIT press Cambridge, MA, 2001).
- [45] W. S. McCulloch and W. Pitts, B. Math. Biol. 5, 115 (1943).
- [46] M. I. Rabinovich, P. Varona, A. I. Selverston, and H. D. Abarbanel, Rev. Mod. Phys. 78, 1213 (2006).
- [47] J. J. Hopfield, Proc. Natl. Acad. Sci. 79, 2554 (1982).
- [48] R. Rojas, Neural Nets: A Systematic Introduction (Springer-Verlag New York Incorporated, 1996).

- [49] S. C. Kak, Adv. Imag. Elect. Phys. 94, 259 (1995).
- [50] M. Peruš, Neural Netw. World 10, 1001 (2000).
- [51] T. Menneer and A. Narayanan, Department of Computer Science, University of Exeter, United Kingdom, Technical Report 329, 1995 (1995).
- [52] M. Zak and C. P. Williams, Int. J. Theor. Phys. 37, 651 (1998).
- [53] E. C. Behrman, J. E. Steck, and S. R. Skinner, in Neural Networks, 1999. IJCNN'99. International Joint Conference on, Vol. 2 (IEEE, 1999) pp. 874–877.
- [54] F. Li, S. Zhao, and B. Zheng, in Signal Processing, 2002 6th International Conference on, Vol. 2 (IEEE, 2002) pp. 1267–1270.
- [55] M. Schuld, I. Sinayskiy, and F. Petruccione, "The quest for a quantum neural network," (2014), in preparation.
- [56] R. W. Hamming, Bell Syst. Tech. J. 29, 147 (1950).
- [57] T. D. Mackay, S. D. Bartlett, L. T. Stephenson, and B. C. Sanders, J. Phys. A 35, 2745 (2002).
- [58] M. Štefaňák, T. Kiss, and I. Jex, New J. Phys. 11, 043027 (2009).
- [59] P. Ribeiro, P. Milman, and R. Mosseri, Phys. Rev. Lett. 93, 190503 (2004).
- [60] The weights a_{ij} can be chosen to be biased so that edges further away from both sinks get a lower value than those close to a sink. This gives another speed-up, especially in high dimensions.
- [61] T. A. Brun, H. A. Carteret, and A. Ambainis, Phys. Rev. A 67, 032304 (2003).
- [62] V. Kendon and B. Tregenna, Phys. Rev. A 67, 042315 (2003).
- [63] I. Sinayskiy and F. Petruccione, Quantum Inf. Process. 11, 1301 (2012).
- [64] M. Bauer, D. Bernard, and A. Tilloy, Phys. Rev. A 88, 062340 (2013).