Supervised graph classification for chiral quantum walks

Artem Kryukov,[1,2] Roman Abramov,[3] Leonid E. Fedichkin,[4] Alexander Alodjants,[3] and Alexey A. Melnikov[2,4,*]

[1]*Moscow Institute of Physics and Technology, 141700 Moscow Region, Russia*

[2]*Terra Quantum AG, 9400 Rorschach, Switzerland*

[3]*ITMO University, 197101 Saint Petersburg, Russia*

[4]*Valiev Institute of Physics and Technology, Russian Academy of Sciences, 117218 Moscow, Russia*

(Received 17 April 2021; accepted 6 January 2022; published 14 February 2022)

Particle transport in quantum systems, which can be modeled by quantum walks on graphs, demonstrates a faster propagation advantage over the corresponding transport in classical systems. As known from several graph examples, achieving further advantages is possible by adding directional control in quantum walks. One way to introduce directional bias is via time-reversal symmetry breaking that can be achieved with chiral quantum walks, where complex phases are added to the edge weights. However, it is not known for which complex phases values and on which graphs quantum transport can be enhanced. Therefore, the classification of graph properties remains an open problem. Here we tackle this graph classification problem with a graph convolutional neural network trained on a set of simulated examples. We find that chiral quantum-walk dynamics leads to almost always faster transport on hypercubes compared to nonchiral dynamics. We connect our paper to physical implementations of quantum walks in superconducting qubits and optical waveguides. Our results open the possibility and flexibility of experimental implementations in demonstrating quantum-walk advantage.

DOI: 10.1103/PhysRevA.105.022208

I. INTRODUCTION

A quantum walk [1–5], an analog to a classical random walk [6–8], is a stepwise process of walker state changes defined on a graph. Both quantum walks and classical random walks appear broadly in nature [9–13], hence making quantum- and random-walk process simulation significant [14–18]. In addition to simulation nature, there are widespread applications [19] in randomized algorithms [20–22], search algorithms [23–26], machine learning [27–29], quantum gates design [30], and universal models for quantum computing [31,32].

Compared to random walks, quantum walks are fundamentally different. First, due to the superposition principle, a quantum walker's state can represent a superposition of several states. Second, quantum walkers have no path defined: only a measurement probabilistically determines a walker position on the graph. Due to these differences, one can get a significant advantage in quantum transport properties on graphs. However, one can also experience a quantum disadvantage because of the lack of directional control on graphs. The directional control cannot be easily implemented because of the time-reversal symmetric evolution of quantum systems [33]. Nonetheless, there is a possibility to break the time-reversal symmetry on graphs with complex edge weights. Quantum-walk dynamics on graphs with complex weights is known as chiral quantum walks [34–36].

The existence of quantum advantage in particle transport depends on several factors: graph topology, initial and target vertices positions, measurement type, and decoherence rates. As was shown in Refs. [37,38], machine learning can be of great help in finding classifying factors. Machine learning is becoming increasingly more useful in automation of problem solving in quantum physics research [39,40]. New reinforcement learning algorithms, inspired by the success of automated designs [41], were shown to be capable of designing new quantum experiments [42–44], and of discovering new concepts in physics from observed data [45,46]. In addition to designing new experiments, machine learning helps in designing new quantum algorithms [47] and protocols [48]. Then, machine learning is used to realize these experiments and protocols in quantum devices, by autonomously learning how to control [49–52], error correct [53–55], and measure quantum devices [56].

In this paper, we consider the chiral quantum walk, which gives many more possibilities to control the transport due to the continuous spectrum of phases attached to graph edges. In general, chiral quantum walks represent one way to add direction into the graph structure. The directed graphs evoke great interest in the quantum domain due to their important applications in information science (see Refs. [57,58]). Remarkably, there exist different ways to introduce directed graphs in the case of a quantum walker. In particular, it is possible through the control of nonunitary terms in the Gorini-Kossakowski-Sudarshan-Lindblad equation [see Eq. (5)], by means of time-dependent Hamiltonians and/or Hamiltonians with *PT* symmetry [57]. In our paper complex-valued weights are added to the edges of the graphs. We study arbitrary graphs and special cases of two possible implementations of quantum walks in currently available quantum devices and train the machine learning model to predict quantum advantage.

*Corresponding author: alexey@melnikov.info

The paper is structured as follows. First, we describe chiral quantum-walk dynamics on graphs, and demonstrate several consequences of these dynamics using hypercube graphs. Next, we introduce a modified architecture of a convolutional neural network to classify graphs according to chiral quantum-walk properties of these graphs. We then demonstrate the convolutional neural network's effectiveness by considering several graph sets, including two special cases relevant for quantum walks' near-term experimental implementations. We finish the paper by summarizing the results and discuss their impact.

Chiral quantum walks and particle transfer

Continuous-time quantum walks (CTQWs) of a single particle are considered in this paper. A particle can be thought of as an excitation that hops in superposition between neighboring vertices on a graph defined by a weighted adjacency matrix $\mathcal{A}$. The quantum-walk dynamics is governed by the Hamiltonian

$$\mathcal{H} = \hbar\omega \sum_{i,j=1}^{d} \beta_{ij} |i\rangle\langle j|, \tag{1}$$

where β_{ij} are couplings between graph vertices i and j, and $\hbar\omega$ is a constant energy associated with transitions between the vertices. The coefficients β_{ij} establish a weighted edge of a graph with d vertices having a weighted adjacency matrix $\mathcal{A}$ with components

$$\mathcal{A}_{ij} = \beta_{ij}, \tag{2}$$

for all edges connecting vertices i and j. A quantum walker, which can be considered as a quantum system with d basis states $|i\rangle$, $i \in [1, d]$, is described by a state $|\psi(t)\rangle$ at the time t. The state $|\psi(t)\rangle = U(t)|\psi(0)\rangle$, where $U(t) = e^{-i\omega\mathcal{A}t}$, can be reverted to the initial state $|\psi(0)\rangle$ with the unitary $U^\dagger(t) = e^{i\omega\mathcal{A}^\dagger t}$. In case of the real-valued couplings β, the initial state is obtained with $U^\dagger(t) = U(-t)$, which corresponds to replacing t with $-t$. Therefore, this CTQW case with real edge weights corresponds to the time-symmetric evolution. The time-symmetric evolution is standard in quantum walks [1,4].

The time asymmetry can arise in chiral quantum walks due to a more general case of complex-valued couplings β. Chiral quantum walks are quantum walks on graphs with complex edge weights, introducing complex phases in the adjacency matrix. We denote the adjacency matrix with complex edge values as $\mathcal{A}_c$, which has the components

$$(\mathcal{A}_c)_{ij} = e^{i\varphi_{ij}} \beta_{ij}, \tag{3}$$

with phases φ_{ij}. One can relate the adjacency matrix to the Hamiltonian $\mathcal{H} = \hbar\omega\mathcal{A}_c$. The chiral quantum-walk dynamics allows for directional control that can contribute to quantum search [59] and to potential time-controlled quantum routers in complex quantum networks [60].

A quantum state $|\psi(t)\rangle = \sum_i^d \alpha_i(t)|i\rangle$, arising from the evolution with the Hamiltonian in Eq. (1), defines a probability distribution over the graph $p_q(t) = |\alpha_i(t)|^2$. Changing a probability distribution is one of the central problems in random walks. Specifically in the particle transfer problem, one is interested in the time it takes for a particle to stochastically move from one, initial, vertex to another, target, vertex on a given graph. Quantum particles can be faster or slower than classical particles in reaching target vertices. To perform a comparison, one simulates a probability vector $p_{\rm cl}(t)$ corresponding to a classical particle evolution, which is

$$p_{\rm cl}(t) = e^{(T-I)t} p(0). \tag{4}$$

Matrix T is the transition matrix, which is a matrix of probabilities T_{ij} for a walker to go from vertex j to vertex i. The transition matrix is obtained from the real-valued adjacency matrix $\mathcal{A}$ by normalizing all columns of $\mathcal{A}$ to one.

To cover a complete picture of quantum dynamics, we have to consider a quantum walker as an open quantum system. It relates to the discussed fundamental difference between classical random walks and quantum walks: the measurement device is needed to read out the walker's state, hence perturbing the unitary dynamics.

Measurement devices can be connected to all the vertices [61], however for particle transfer studies one needs only one measurement at the target vertex. A measurement apparatus coupled to a target vertex can be modeled as a sink vertex, to which walkers decay with a certain rate once the target vertex is reached [62]. We simulate the chiral quantum walks combining open dynamics by numerically solving the Gorini-Kossakowski-Sudarshan-Lindblad equation:

$$\frac{d\rho(t)}{dt} = -i\omega[\mathcal{A}_c, \rho(t)] + \gamma\left(L\rho(t)L^\dagger - \frac{1}{2}\{L^\dagger L, \rho(t)\}\right), \tag{5}$$

where $L = |\text{sink}\rangle\langle\text{target}|$ is a Lindbladian term that connects target and sink vertices and γ is the coupling strength. In case $\gamma = 0$, one goes back to the unitary dynamics case discussed above, and no particle will be captured.

The solution of Eq. (5) for the hypercube graph is shown in Fig. 1 for three different hypercube dimensions: $n = 2$ (square), $n = 3$ (cube), and $n = 4$ (tesseract). The transfer probabilities, which are probabilities of reaching the target vertex, are compared for classical, quantum, and chiral quantum walks. From Fig. 1 one sees that depending on the initial and target vertex positions and the hypercube dimension, different types of walks reach the $1/\log d$ transfer probability threshold.

II. CONVOLUTIONAL NEURAL NETWORK CLASSIFYING CHIRAL QUANTUM WALKS

To classify graphs according to the classical random-walk and quantum-walk properties, we design a convolutional neural network similar to the one introduced in Ref. [37]. The classical-quantum convolutional neural network (CQCNN) used in this paper has several differences compared to the original CQCNN architecture. First, the two previous implementations [37,63] were designed to deal only with real-valued unweighted adjacency matrices $\mathcal{A}$. Here we study a more general case of chiral quantum walks requiring reliable feature extraction from complex-valued weighted adjacency matrices $\mathcal{A}_c$. Second, the previous CQCNN implementations were not designed for large input matrices and high variability

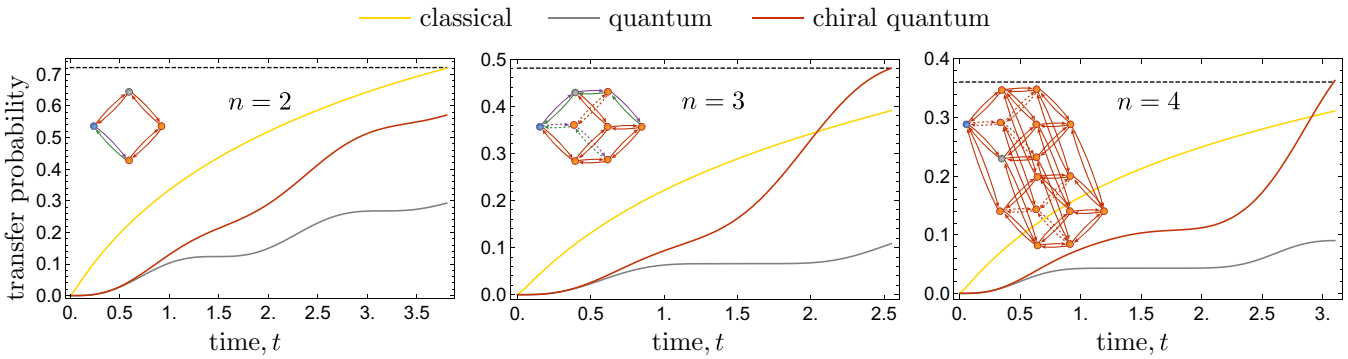

FIG. 1. Particle transfer probabilities on hypercube graphs of dimensions $n = 2$, 3, and 4. Particle transfer is performed via three different stochastic mechanisms: classical random walk (yellow), quantum walk (gray), and chiral quantum walk (orange). The transfer probability threshold is shown as a dashed line. The corresponding hypercubes are shown in the inset. Initial vertices are shown as purple, whereas targets are green.

in their size. Here, by studying hypercube graphs, we have input matrices scaling as $d = 2^n$ with the hypercube dimension n.

The CQCNN architecture designed for graph classification in this paper is shown in Fig. 2. The neural network consists of two parts: convolutional layers and classification layers. The input adjacency matrix $\mathcal{A}_c$ comes in the form of two real-valued matrices $\mathrm{Re}\mathcal{A}_c$ and $\mathrm{Im}\mathcal{A}_c$. The total number of activated input neurons is $2d^2$ with normalized neuron values in the region from -1 to 1. The input is then passed through five edge-to-vertex (ETV) filters, each of the size d^2, to perform convolution. The ETV filter $\mathrm{ETV}^{(m)}$ performs the following operation on an input matrix I to produce an output vector o with components:

$$o_i^{\mathrm{ETV}^{(m)}}(I) = \sum_{k=1,k\neq i}^{n} \left(\mathrm{ETV}_{ik}^{(m)}\right) I_{ik} + \left(\mathrm{ETV}_{ki}^{(m)}\right) I_{ki}. \qquad (6)$$

The filter aims to extract features using information about the adjacent edges' connectivity for every given vertex. The five ETV filters' output consists of ten vectors of dimension d, which are then flattened and passed through two fully connected layers of neurons with rectified linear unit activation functions to allow for nonlinear classification functions. The hidden layer consists of five neurons, independently of d. The final output is a value of two neurons y_1 and y_2 that are converted to an output class as $\mathrm{argmax}(y_1, y_2)$.

We next apply the described CQCNN network to different datasets. First, two special cases of hypercubes that are relevant for experimental implementation are considered. The first case is linked to the qubit quantum register implementation, whereas the second case is relevant for photonic implementation. We then consider a general case of arbitrary hypercube graphs with chirality, followed by an example of random graph sets.

A. Chiral quantum walks on hypercubes with qubits

The first special case of chiral weighted hypercube graphs under consideration is defined by an underlying symmetry appearing in the binary encoding of vertices [18]. One can embed an n-dimensional hypercube into an n-dimensional space, where each axis has exactly two positions for the vertices. One can label these positions as 0 and 1. This case is graph-

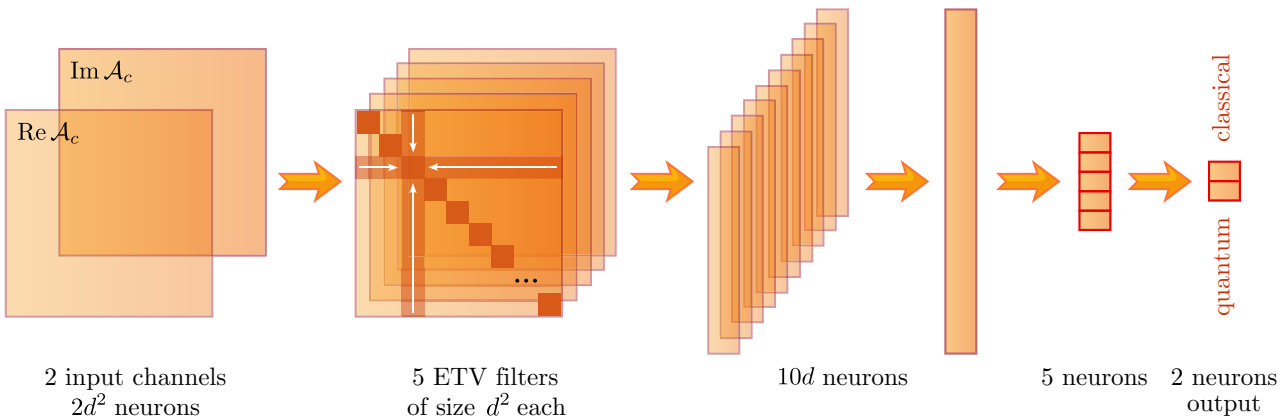

FIG. 2. Schematic representation of a CQCNN implemented as a binary classifier. The neural network consists of convolutional layers and fully connected layers. The CQCNN takes a graph adjacency matrix $\mathcal{A}_c$ input and outputs a prediction of which walker is faster on a given graph: quantum or classical. The total number of neurons is specified for an input graph with d vertices. This CQCNN architecture is fixed for all the supervised learning tasks studied in the paper.

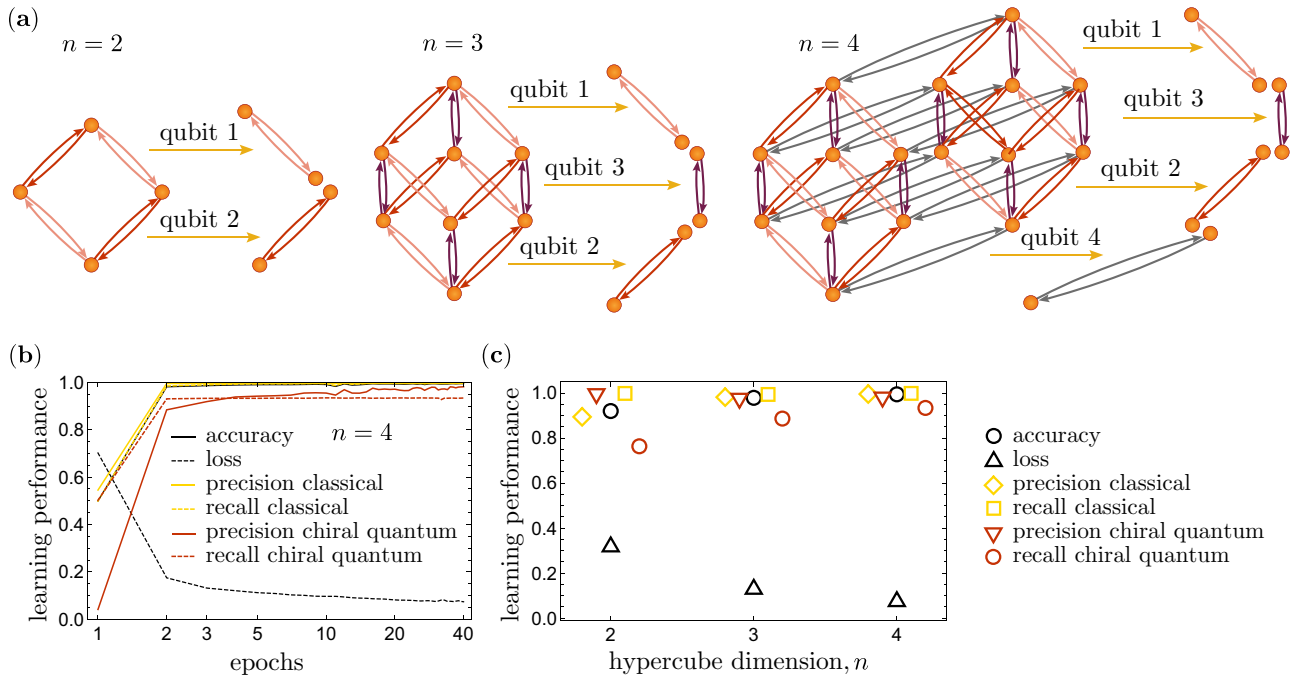

FIG. 3. Chiral quantum walks on hypercubes with a binary encoding symmetry. (a) Encoding a state of a quantum walker on 2^n vertices in n qubits. Examples of $n = 2$, 3, and 4 are shown. The same colors represent edges related to each other due to the binary encoding symmetry. (b) Learning curves of the CQCNN on the chiral hypercube graphs with the above symmetry for $n = 4$ (tesseract). The average over 100 independent parallel CQCNN runs is shown. (c) Learning performance after 40 epochs, with the batch size equal to the total training dataset, shown for different hypercube dimensions. The average over 100 independent parallel CQCNN runs is shown.

ically illustrated in Fig. 3(a). Edges, for which the connected vertices have the same binary encoding up to a single bit flip, are assigned the same weight $e^{i\varphi_k}$ as they correspond to the symmetric edges shown in Fig. 3(a). For this special case, the adjacency matrix of the hypercube (hc) can be decomposed in the following way:

$$\mathcal{A}_c^{\mathrm{hc}} = \sum_{i=0}^{n-1} I_2^{\otimes i} \otimes \mathcal{A}_c^{\mathrm{qubit}\,i} \otimes I_2^{\otimes n-1-i}, \tag{7}$$

where I_2 is a two-dimensional identity matrix, $\mathcal{A}_c^{\mathrm{qubit}\,i}$ is a two-vertex adjacency matrix corresponding to an arbitrary Hamiltonian acting on a qubit i. This symmetry and the consequential simplification provide a possibility to implement quantum walks in the space of qubits. Moreover, these quantum walks can be simulated on a quantum computer efficiently with n qubits for $d = 2^n$ dimensional graphs. Successful implementation of quantum walks on up to 25-dimensional hypercubes was demonstrated on the IBM Q quantum computer [18]. The quantum transfer speedup was demonstrated using a time-reversal symmetric quantum walk; however, the same mapping can be used to simulate a chiral quantum walk.

To determine if there is a quantum advantage in the chiral quantum-walk case, one has to simulate the walk for given amplitudes and locations of a given initial and target vertices. In addition to this, a sink vertex described in Eq. (5) introduces a difference in the dynamics making it even harder to estimate the potential quantum advantage. For that reason, we next use the CQCNN to classify graphs according to their quantum advantage.

To achieve an accurate prediction on quantum advantage, we first generate sets of 1000 graphs with random edge phases from zero to 2π (chosen uniformly) for each of the $2^n - 1$ possible target vertices for a fixed initial vertex 1. These sets are generated for $n = 2$, 3, and 4 separately by numerically solving Eq. (5). We use 70% of examples in the datasets for CQCNN training, and 30% are used in a test set. We use the batch size equal to the total dataset, known as the batch mode, and minibatches of size 100 to compute the loss and perform back-propagation. These datasets are used to train and test 100 independent CQCNN networks, each contributing to an averaged learning performance result shown in Figs. 3(b) and 3(c).

The learning performance is evaluated by four metrics: accuracy, loss, precision, and recall. The loss that corresponds to the cross entropy loss function estimates the quality of the machine learning model on the training set, whereas the accuracy, which is defined as a fraction of the correctly classified graphs from the test set, estimates the quality of the model on the test set. Precision and recall shed light on the performance on the test set further: recall is the fraction of correct predictions in a particular class (classical or quantum), and precision is the fraction of predictions in a particular class (classical or quantum) that turned out to be correct.

As seen in Fig. 3(c), the performance measures on test graphs have similarities: the learning done on graphs with $n = 2$, 3, and 4 demonstrates an average performance over 0.9 for classification accuracy with the performance being best in case of $n = 4$. An improvement of performance with increasing n, given the same dataset size, is explained by the increase in the number of edges less affected by the measure-

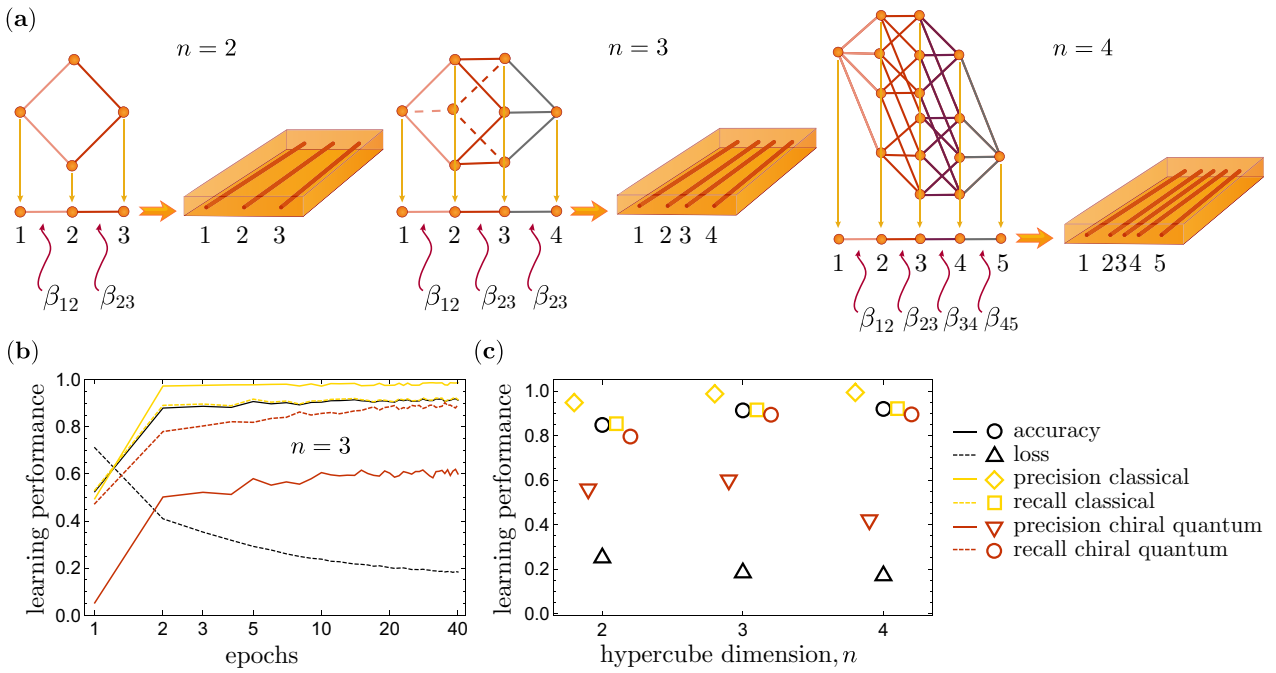

FIG. 4. Quantum walks on weighed hypercubes with a Hamming distance symmetry. (a) Encoding a state of a quantum walker on 2^n vertices in $n+1$ optical waveguides. Examples of $n = 2$, 3, and 4 are shown. The same colors represent edges with the same Hamming distance relative to the leftmost initial vertex. (b) Learning curves of the CQCNN on the weighted hypercube graphs with the above symmetry for $n = 3$ (cube). The average over 100 independent parallel CQCNN runs is shown. (c) Learning performance after 40 epochs, with the batch size equal to the total training dataset, shown for different hypercube dimensions. The average over 100 independent parallel CQCNN runs is shown.

ment device. The phase value can be ignored for the edges where the particle evolves coherently, which is learned by the neural network. Details on the learning performance dynamics are shown in Fig. 3(b) for $n = 4$. We achieve a nearly perfect classification performance only after three epochs, each using the full training dataset.

B. Quantum walks on weighted hypercubes in optical waveguides

The second experimentally relevant dataset that we consider uses the symmetry appearing relative to the initial vertex. On hypercube graphs vertices with the same Hamming distance from the initial vertex have the same transfer probability. In this symmetry case, shown in Fig. 4(a), edges connecting vertices with the same Hamming distance have the same weight. Figure 4(a) demonstrates that a quantum-walk space can be significantly reduced from 2^n vertices to $n+1$ vertex in the mapped space. There is a change in hopping amplitudes β_{ij} associated with this mapping. The mapping is well suited for photonic quantum walks, which stand out from other experiments. In photonic experiments, it is quite natural to implement single-particle and noninteracting multiparticle continuous-time quantum walks. This can be done via coherent propagation of photons in optical waveguide arrays, where the distance between waveguides controls the amplitudes β_{ij} [64–66].

To find out if there exists a quantum advantage in a weighted hypercube of considered symmetry, one has to simulate the walk for given weights, and locations of given initial and target vertices. As in the previous section, the sink vertex described in Eq. (5) introduces a difference in probability distributions which is hard to estimate in practice. For that reason, we next use the CQCNN to classify graphs according to their quantum advantage.

We have generated sets of 1000 graphs with random edge weights from 0 to 1 (chosen uniformly) for each of the $2^n - 1$ possible target vertices for a fixed initial vertex 1. These sets are generated for $n = 2$, 3, and 4 separately by numerically solving Eq. (5). We use 70% of examples in the datasets for CQCNN training, and 30% are used in a test set. These datasets are used to train and test 100 independent CQCNN networks, each contributing to an averaged learning performance result shown in Figs. 4(b) and 4(c). We use the batch size equal to the total dataset, known as the batch mode, and minibatches of size 100 to compute the loss and perform backpropagation. From the results in Fig. 4(c), where the learning is done on graphs with $n = 2$, 3, and 4, we obtain the classification accuracy above 0.85 for all n, with the performance being best in case of $n = 4$. An improvement of performance with increasing n, given the same dataset size, is interesting and highlights that CQCNN learns more from larger graphs. Details on the learning performance dynamics are shown in Fig. 4(b) for $n = 3$.

C. Chiral quantum walks on arbitrary hypercubes and random graphs

We finish our results with the most general case studied in this paper: the case of arbitrary chiral quantum-walk dynamics

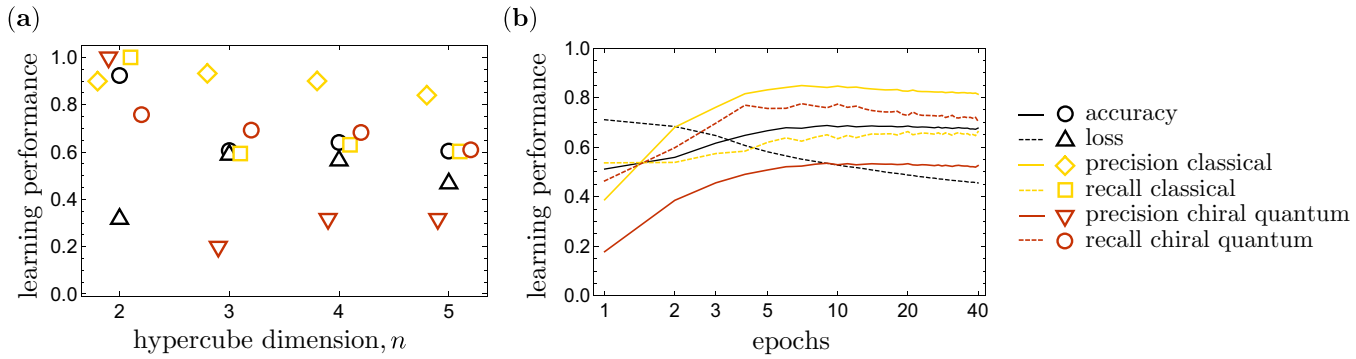

FIG. 5. (a) Learning results for the chiral quantum walks on hypercube graphs of dimension $n = 2$, 3, 4, and 5. Training is performed during 40 epochs, with the batch size equal to the total training dataset. The average over 100 independent parallel CQCNN runs is shown. (b) Learning results for the chiral quantum walks on random graphs with $d = 15$ vertices. Training is performed during 40 epochs, with the batch size equal to the total training dataset. The average over 100 independent parallel CQCNN runs is shown.

on hypercubes, and dynamics of random graphs. The learning results are shown in Fig. 5 for hypercubes (a) and random (b) graphs. Random graphs are uniformly sampled from the set of connected graphs. We generated datasets of 1000 graphs with random edge phases from 0 to 2π (chosen uniformly) for each of the $2^n - 1$ possible target vertices for a fixed initial vertex 1. The displayed classification results are worse than in the experimentally relevant special hypercube cases in previous sections. The performance decreases with n, which is explained by the limited availability of the training data. Given that the number of different graphs grows as factorial with n and d, 1000 training examples are becoming less sufficient for larger graphs. Nonetheless, the classification accuracy surpasses the random guess significantly, making it possible to use the new CQCNN architecture for chiral quantum walks on arbitrary graphs.

III. DISCUSSION

Chiral quantum walks, which introduce time-symmetry breaking, are studied in this paper. Introducing direction-dependent transitions on graphs opens a possibility to further speed up quantum transport. It is, however, difficult to find out if a given graph topology, a given configuration of edge phases, and given target vertices would allow observing a quantum advantage.

In this paper, we introduced a supervised learning methodology to study quantum transport of chiral quantum walks. We optimized the architecture and significantly reduced network parameters compared to the previous versions of graph convolutional neural networks for quantum-walk classification. The presented development of the graph neural network approach is designed to predict the performance of a quantum circuit, or an underlying quantum system, given its graph structure. We tested our machine learning algorithm on various datasets, both special cases of experimental implementation relevance and general cases of hypercube and random graphs. First, we have observed that chiral quantum walks have improved quantum particle transfer on hypercubes. Second, we achieved the classification accuracy of 90–100% in the case of quantum walks with qubits, and 80–90% accuracy in the case of quantum walks in optical waveguides. Finally, our graph classifier learned by achieving an accuracy of 60–70% on general hypercube cases, and random graphs with up to 15 vertices. Given the relatively small size of the training set, our results demonstrate the practical possibility of predicting chiral quantum-walk transport advantage on arbitrary graphs. This paper paves the way for faster particle transport leading to more efficient energy transport and better quantum information processing.

ACKNOWLEDGMENTS

Results of Secs. II A and II B were obtained with financial support from Russian Foundation for Basic Research Grant No. 19-52-52012 MHT_a, and from Priority 2030 program. Results of Sec. II C were obtained at the Valiev Institute of Physics and Technology of Russian Academy of Sciences (RAS), and supported by Grant No. FFNN-2022-0016 of the Ministry of Science and Higher Education of Russia for Valiev Institute of Physics and Technology of RAS.

[1] Y. Aharonov, L. Davidovich, and N. Zagury, Quantum random walks, Phys. Rev. A **48**, 1687 (1993).

[2] J. Kempe, Quantum random walks: An introductory overview, Contemp. Phys. **44**, 307 (2003).

[3] N. Konno, Quantum walks, in *Quantum Potential Theory* (Springer, New York, 2008), pp. 309–452.

[4] S. E. Venegas-Andraca, Quantum walks: A comprehensive review, Quantum Inf. Process. **11**, 1015 (2012).

[5] K. Kadian, S. Garhwal, and A. Kumar, Quantum walk and its application domains: A systematic review, Comput. Sci. Rev. **41**, 100419 (2021).

[6] K. Pearson, The problem of the random walk, Nature (London) **72**, 342 (1905).

[7] L. Lovász, Random walks on graphs: A survey, in *Combinatorics: Paul Erdõs is Eighty* (János Bolyai Math. Soc., Budapest, 1996), Vol. 2, pp. 353–397.

[8] D. Aldous and J. Fill, Reversible Markov chains and random walks on graphs (2002, unpublished), unfinished monograph recompiled 2014, https://www.stat.berkeley.edu/~aldous/RWG/book.pdf.

[9] M. Kac, Random walk and the theory of Brownian motion, Am. Math. Mon. **54**, 369 (1947).

[10] F. Bartumeus, M. G. E. da Luz, G. M. Viswanathan, and J. Catalan, Animal search strategies: A quantitative random-walk analysis, Ecology **86**, 3078 (2005).

[11] D. Brockmann, L. Hufnagel, and T. Geisel, The scaling laws of human travel, Nature (London) **439**, 462 (2006).

[12] G. S. Engel, T. R. Calhoun, E. L. Read, T.-K. Ahn, T. Mančal, Y.-C. Cheng, R. E. Blankenship, and G. R. Fleming, Evidence for wavelike energy transfer through quantum coherence in photosynthetic systems, Nature (London) **446**, 782 (2007).

[13] M. Mohseni, P. Rebentrost, S. Lloyd, and A. Aspuru-Guzik, Environment-assisted quantum walks in photosynthetic energy transfer, J. Chem. Phys. **129**, 174106 (2008).

[14] H. Gould, J. Tobochnik, and W. Christian, *An Introduction to Computer Simulation Methods* (Addison-Wesley, New York, 1988), Vol. 1.

[15] M. B. Cohen, J. Kelner, J. Peebles, R. Peng, A. Sidford, and A. Vladu, Faster algorithms for computing the stationary distribution, simulating random walks, and more, in *Proceedings of the 57th Annual Symposium on Foundations of Computer Science* (IEEE, New Brunswick, NJ, 2016), pp. 583–592.

[16] J. A. Izaac and J. B. Wang, pyCTQW: A continuous-time quantum walk simulator on distributed memory computers, Comput. Phys. Commun **186**, 81 (2015).

[17] S. Apers, A. Sarlette, and F. Ticozzi, Simulation of quantum walks and fast mixing with classical processes, Phys. Rev. A **98**, 032115 (2018).

[18] A. K. Madhu, A. A. Melnikov, L. E. Fedichkin, A. Alodjants, and R.-K. Lee, Quantum walk processes in quantum devices, arXiv:2012.14386 (2020).

[19] L. Fedichkin and F. Meshchaninov, Analysis and applications of quantum walks, J. Math. Sci. **252**, 104 (2020).

[20] R. Motwani and P. Raghavan, *Randomized Algorithms* (Cambridge University, New York, 1995), Chap. 6, p. 127.

[21] C. Gkantsidis, M. Mihail, and A. Saberi, Random walks in peer-to-peer networks: Algorithms and evaluation, Perform. Eval. **63**, 241 (2006).

[22] A. Ambainis, Quantum walks and their algorithmic applications, Int. J. Quantum Inform. **1**, 507 (2003).

[23] G. Oshanin, K. Lindenberg, H. S. Wio, and S. Burlatsky, Efficient search by optimized intermittent random walks, J. Phys. A **42**, 434008 (2009).

[24] A. M. Childs and J. Goldstone, Spatial search by quantum walk, Phys. Rev. A **70**, 022314 (2004).

[25] S. Chakraborty, L. Novo, A. Ambainis, and Y. Omar, Spatial Search by Quantum Walk is Optimal for Almost All Graphs, Phys. Rev. Lett. **116**, 100501 (2016).

[26] T. Osada, B. Coutinho, Y. Omar, K. Sanaka, W. J. Munro, and K. Nemoto, Continuous-time quantum-walk spatial search on the bollobás scale-free network, Phys. Rev. A **101**, 022310 (2020).

[27] M. Szummer and T. Jaakkola, Partially labeled classification with Markov random walks, in *Advances in Neural Information Processing Systems*, edited by T. G. Dietterich, S. Becker, and Z. Ghahramani (MIT, Cambridge, MA, 2002), Vol. 14, pp. 945–952.

[28] L. Grady, Random walks for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. **28**, 1768 (2006).

[29] G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel, Quantum Speedup for Active Learning Agents, Phys. Rev. X **4**, 031002 (2014).

[30] A. A. Melnikov and L. E. Fedichkin, Quantum walks of interacting fermions on a cycle graph, Sci. Rep. **6**, 34226 (2016).

[31] A. M. Childs, Universal Computation by Quantum Walk, Phys. Rev. Lett. **102**, 180501 (2009).

[32] A. M. Childs, D. Gosset, and Z. Webb, Universal computation by multiparticle quantum walk, Science **339**, 791 (2013).

[33] M. Szegedy, Quantum speed-up of Markov chain based algorithms, in *45th Annual Symposium on Foundations of Computer Science* (IEEE, New York, 2004), pp. 32–41.

[34] Z. Zimboras, M. Faccin, Z. Kadar, J. D. Whitfield, B. P. Lanyon, and J. Biamonte, Quantum transport enhancement by time-reversal symmetry breaking, Sci. Rep. **3**, 2361 (2013).

[35] D. Lu, J. D. Biamonte, J. Li, H. Li, T. H. Johnson, V. Bergholm, M. Faccin, Z. Zimborás, R. Laflamme, J. Baugh, and S. Lloyd, Chiral quantum walks, Phys. Rev. A **93**, 042302 (2016).

[36] J. Turner and J. Biamonte, Topological classification of time-asymmetry in unitary quantum processes, J. Phys. A **54**, 235301 (2021).

[37] A. A. Melnikov, L. E. Fedichkin, and A. Alodjants, Predicting quantum advantage by quantum walk with convolutional neural networks, New J. Phys. **21**, 125002 (2019).

[38] C. Moussa, H. Calandra, and V. Dunjko, To quantum or not to quantum: towards algorithm selection in near-term quantum optimization, Quantum Sci. and Technol. **5**, 044009 (2020).

[39] V. Dunjko and H. J. Briegel, Machine learning & artificial intelligence in the quantum domain: A review of recent progress, Rep. Prog. Phys. **81**, 074001 (2018).

[40] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[41] M. Krenn, M. Malik, R. Fickler, R. Lapkiewicz, and A. Zeilinger, Automated Search for New Quantum Experiments, Phys. Rev. Lett. **116**, 090405 (2016).

[42] A. A. Melnikov, H. Poulsen Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel, Active learning machine learns to create new quantum experiments, Proc. Natl. Acad. Sci. USA **115**, 1221 (2018).

[43] A. A. Melnikov, P. Sekatski, and N. Sangouard, Setting Up Experimental Bell Tests with Reinforcement Learning, Phys. Rev. Lett. **125**, 160401 (2020).

[44] T. Menke, F. Häse, S. Gustavsson, A. J. Kerman, W. D. Oliver, and A. Aspuru-Guzik, Automated design of superconducting circuits and its application to 4-local couplers, npj Quantum Inf. **7**, 49 (2021).

[45] R. Iten, T. Metger, H. Wilming, L. del Rio, and R. Renner, Discovering Physical Concepts with Neural Networks, Phys. Rev. Lett. **124**, 010508 (2020).

[46] H. Poulsen Nautrup, T. Metger, R. Iten, S. Jerbi, L. M. Trenkwalder, H. Wilming, H. J. Briegel, and R. Renner, Operationally meaningful representations of physical systems in neural networks, arXiv:2001.00593 (2020).

[47] L. Cincio, Y. Subaşı, A. T. Sornborger, and P. J. Coles, Learning the quantum algorithm for state overlap, New J. Phys. **20**, 113022 (2018).

[48] J. Wallnöfer, A. A. Melnikov, W. Dür, and H. J. Briegel, Machine learning for long-distance quantum communication, PRX Quantum **1**, 010301 (2020).

[49] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, Reinforcement Learning with Neural Networks for Quantum Feedback, Phys. Rev. X **8**, 031084 (2018).

[50] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Reinforcement Learning in Different Phases of Quantum Control, Phys. Rev. X **8**, 031086 (2018).

[51] I. A. Luchnikov, S. V. Vintskevich, D. A. Grigoriev, and S. N. Filippov, Machine Learning Non-Markovian Quantum Dynamics, Phys. Rev. Lett. **124**, 140502 (2020).

[52] F. Schäfer, M. Kloc, C. Bruder, and N. Lörch, A differentiable programming method for quantum control, Mach. Learn.: Sci. Technol. **1**, 035009 (2020).

[53] H. Poulsen Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, Optimizing quantum error correction codes with reinforcement learning, Quantum **3**, 215 (2019).

[54] A. Valenti, E. van Nieuwenburg, S. Huber, and E. Greplova, Hamiltonian learning for quantum error correction, Phys. Rev. Research **1**, 033092 (2019).

[55] R. Sweke, M. S. Kesselring, E. P. van Nieuwenburg, and J. Eisert, Reinforcement learning decoders for fault-tolerant quantum computation, Mach. Learn.: Sci. Technol. **2**, 025005 (2020).

[56] G. Liu, M. Chen, Y.-X. Liu, D. Layden, and P. Cappellaro, Repetitive readout enhanced by machine learning, Mach. Learn.: Sci. Technol. **1**, 015003 (2020).

[57] J. A. Izaac, J. B. Wang, P. C. Abbott, and X. S. Ma, Quantum centrality testing on directed graphs via *pt*-symmetric quantum walks, Phys. Rev. A **96**, 032305 (2017).

[58] K. Wang, Y. Shi, L. Xiao, J. Wang, Y. N. Joglekar, and P. Xue, Experimental realization of continuous-time quantum walks on directed graphs and their application in PageRank, Optica **7**, 1524 (2020).

[59] T. G. Wong, Quantum walk search with time-reversal symmetry breaking, J. Phys. A **48**, 405303 (2015).

[60] Y. Liu and D. L. Zhou, Quantum state transfer along a ring with time-reversal asymmetry, Phys. Rev. A **91**, 052318 (2015).

[61] A. A. Melnikov, L. E. Fedichkin, R.-K. Lee, and A. Alodjants, Deep neural networks classifying transfer efficiency in complex networks, in *Proceedings of the Opto-Electronics and Communications Conference* (IEEE, Taipei, Taiwan, 2020), pp. 1–3.

[62] A. A. Melnikov, L. E. Fedichkin, and A. Alodjants, On training a classifier of hitting times for quantum walks, AIP Conf. Proc. **2241**, 020029 (2020).

[63] A. A. Melnikov, L. E. Fedichkin, R.-K. Lee, and A. Alodjants, Machine learning transfer efficiencies for noisy quantum walks, Adv. Quantum Technol. **3**, 1900115 (2020).

[64] M. Will, S. Nolte, B. N. Chichkov, and A. Tünnermann, Optical properties of waveguides fabricated in fused silica by femtosecond laser pulses, Appl. Opt. **41**, 4360 (2002).

[65] H. B. Perets, Y. Lahini, F. Pozzi, M. Sorel, R. Morandotti, and Y. Silberberg, Realization of Quantum Walks with Negligible Decoherence in Waveguide Lattices, Phys. Rev. Lett. **100**, 170506 (2008).

[66] A. A. Melnikov, A. P. Alodjants, and L. E. Fedichkin, Tunneling in double-layer optical waveguides as quantum walks on graphs, Proc. Steklov Inst. Math. **313**, 142 (2021).