Quantum link prediction in complex networks

Moutinho, João P.; Melo, André; Coutinho, Bruno; Kovács, István A.; Omar, Yasser

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Quantum link prediction in complex networks

João P. Moutinho,[1,2,*] André Melo,[3] Bruno Coutinho,[2] István A. Kovács,[4,5,6] and Yasser Omar[1,7,8]

[1]*Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal*
[2]*Instituto de Telecomunicações, 1049-001 Lisboa, Portugal*
[3]*Kavli Institute of Nanoscience, Delft University of Technology, 2628 CD Delft, The Netherlands*
[4]*Department of Physics and Astronomy, Northwestern University, Evanston, 60208 Illinois, USA*
[5]*Northwestern Institute on Complex Systems, Northwestern University, Evanston, 60208 Illinois, USA*
[6]*Department of Network and Data Science, Central European University, 1051 Budapest, Hungary*
[7]*Portuguese Quantum Institute, 1049-001 Lisboa, Portugal*
[8]*Centro de Física e Engenharia de Materiais Avançados (CeFEMA), Physics of Information and Quantum Technologies Group, 1049-001 Lisboa, Portugal*

Predicting new links in physical, biological, social, or technological networks has a significant scientific and societal impact. Path-based link prediction methods utilize the explicit counting of even- and odd-length paths between nodes to quantify a score function and infer new or unobserved links. Here, we propose a quantum algorithm for path-based link prediction using a controlled continuous-time quantum walk to encode even and odd path-based prediction scores. Through classical simulations on a few real networks, we confirm that the quantum walk scoring function performs similarly to other path-based link predictors. In a brief complexity analysis we identify the potential of our approach in uncovering a quantum speedup for path-based link prediction.

I. INTRODUCTION

From genes and proteins that govern our cellular function, to our everyday use of the Internet, Nature and our lives are surrounded by interconnected systems [1]. Network science aims to study these complex networks and provides a powerful framework to understand their structure, function, dynamics, and growth. Studies in network science typically have a substantial computational component, borrowing tools from graph theory to extract relevant information about the underlying system. With the advent of quantum computation, a natural question to ask is which problems in network science can be explored with this new computing paradigm and what benefits it can yield. This question can be interpreted in at least two different ways. First, there is a large body of work in quantum algorithms for graph theoretical problems, some examples being Refs. [2–5], which may have their own applications in network science problems. However, network science algorithms often look for specific patterns or organizing principles based on empirical observations from the real underlying systems, which may warrant the development of problem-specific quantum algorithms. This constitutes a novel research direction, different from the development of more general graph-theoretical algorithms. Previous connections have been made between quantum phenomena and complex networks, both by using quantum tools to study complex networks and by using complex network tools to study quantum systems [10]. Nevertheless, to our knowledge, potential quantum speedups for network science problems have not been addressed.

In this work we propose a quantum algorithm to the problem of link prediction in complex networks using continuous-time quantum walks (CTQW) [11,12] inspired by popular path-based methods and discuss potential quantum speedups over classical algorithms. The objective in link prediction is to identify unknown connections in a network [13–18]. For example, in social networks, we aim to predict which individuals will develop shared friendships, professional relations, exchange of goods and services, or others [13,14]. In biological networks, the main focus is the issue of data incompleteness, which hinders our understanding of complex biological function. For example, in protein-protein interaction (PPI) networks link prediction methods have already proven to be a valuable tool in mapping out the large amount of missing data [19,20]. While there are many approaches to the problem of link prediction [18], such as using machine learning techniques [21,22], stochastic block models [23], or studying global perturbations [24] other methods focused on simple topological features like paths of different length between nodes quantified by powers of the adjacency matrix. Path-based methods are simple but remarkably popular and have been shown to be competitive with other approaches in networks of various types [18,25–27]. In our work we show that quantum walks can be used as a natural encoding of path-based link prediction in the development of a quantum algorithm.

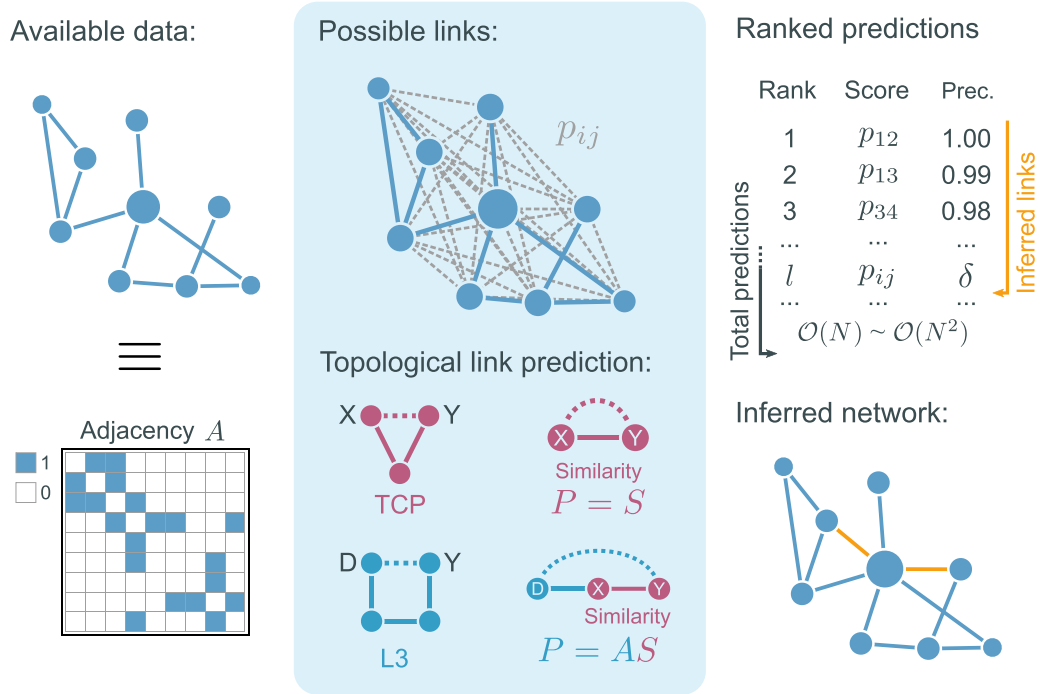*Corresponding author: joao.p.moutinho@tecnico.ulisboa.pt

FIG. 1. Classical path-based link prediction. Link prediction methods take as input in a complex network with a corresponding adjacency matrix A. Each method then associates a prediction value p_{ij}, or *score*, to every pair of nodes $\{i, j\}$, such that a higher value p_{ij} correlates to a higher probability of the link $\{i, j\}$ appearing. This requires assumptions about the organizing principles of each network. Predictions based on the triangle closure principle (TCP) rely on similarity between nodes, represented as a matrix S, a common assumption about connections in social networks. This can be quantified in the simplest case as $P \sim S \sim A^2$, counting paths of length 2 between pairs of nodes. As an alternative, proteins often connect to others that are similar to their neighbors, but not necessarily similar to themselves, quantified for example as $P \sim AS \sim A^3$, counting paths of length 3 between nodes [19]. Most classical link prediction methods output all nonzero entries of matrix P, organized in a ranked list of scores from highest to lowest, where the relevant top l predictions are those where the precision is above a user-determined threshold δ. The relevant predictions are considered as new inferred links, represented in yellow, while the rest are discarded.

II. CLASSICAL PATH-BASED LINK PREDICTION

Link prediction methods take as input a graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes with size $N = |\mathcal{V}|$ and $\mathcal{E}$ is the set of undirected links, often described by the adjacency matrix $A \in \{0, 1\}^{N \times N}$,

$$A_{ij} = A_{ji} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{E}, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

and output a matrix of predictions $P \in \mathbf{R}^{N \times N}$ where each entry p_{ij} is a score value quantifying the likelihood of a link existing between nodes i and j (see Fig. 1). Each method computes P differently, depending on the assumptions made about the network and its emergent topological features. Most path-based methods are based on the triadic closure principle (TCP), assuming that two nodes are more likely to connect the more similar they are [17,19]. Given a matrix $S \in \mathbf{R}^{N \times N}$, quantifying the similarity between any two nodes, predictions based on TCP assume that

$$P = S. \tag{2}$$

Similarity is often quantified based on the number of shared connections, i.e., paths of length two between two nodes, which can be computed as $S = A^2$. A possible generalization is to consider a linear combination of even powers of A. It has been shown that, despite its dominant use in biological

networks, the TCP approach is not valid for most protein pairs [19]. Instead, in Ref. [19], a link prediction method (L3) was proposed without the assumption that node similarity correlates with direct connectivity. L3 is based on the assumption that a potential new link (i, j) relies on i being similar to the existing neighbors of j. In matrix form, predictions based on the L3 paradigm may be computed by extending the similarity matrix S one step over the adjacency matrix

$$P = AS, \tag{3}$$

as illustrated in Fig. 1. Considering the simple case of $S = A^2$, the authors of Ref. [19] defined P based on $AS = A^3$, with an added degree of normalization. Their results showed the L3 method significantly outperforms other TCP-based methods in the prediction of protein-protein interactions. At the same time, the LO method was proposed in Ref. [28], which represents P as a linear combination of odd powers of A, also showing significant improvements over TCP-based methods. Other follow-up studies proposed different L3-based methods [29] and further studied the application of L3- and TCP-based methods [25,30], concluding that L3-based methods perform well in various network categories.

Our quantum approach takes inspiration from both these paradigms, utilizing even (TCP) and odd (L3-like) powers of A. One of the main reasons why link prediction may prove

Quantum link prediction circuit:
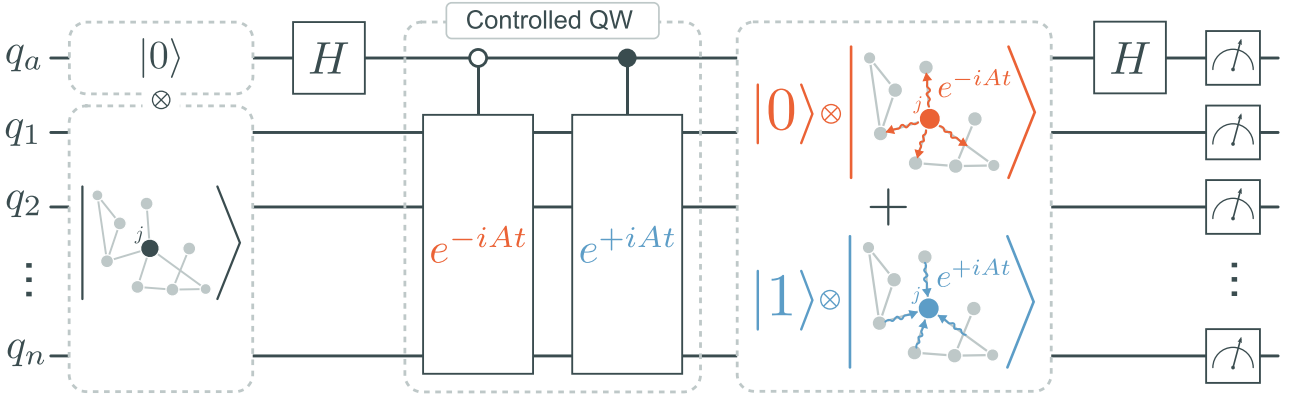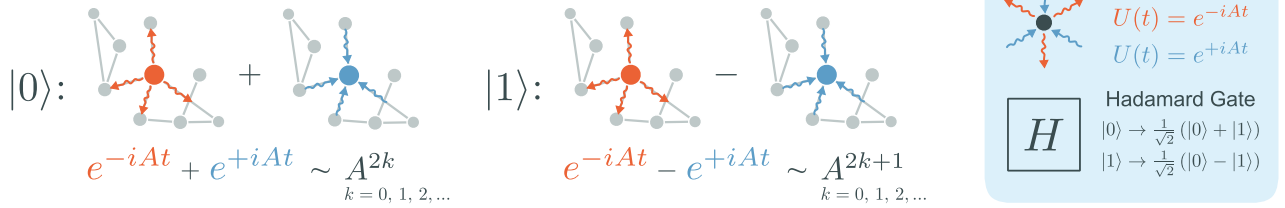

Ancilla qubit measurement:

FIG. 2. Quantum link prediction (QLP) circuit. The algorithm requires a total of $n = \log_2(N)$ qubits to encode each of the N nodes as a basis state and an extra ancilla qubit q_a to perform controlled operations. The ancilla qubit is initialized to $|0\rangle$ and the remaining n qubits are initialized to some basis state $|j\rangle$ corresponding to a node in the network. The first Hadamard gate creates a superposition of $|0\rangle$ and $|1\rangle$ in q_a. A controlled QW, represented here by two operators, applies e^{-iAt} to the remaining n qubits if $q_a = |0\rangle$ and e^{+iAt} if $q_a = |1\rangle$. Given that q_a is in a superposition of $|0\rangle$ and $|1\rangle$, the controlled QW creates a superposition of the two evolutions entangled to the ancilla qubit. A second Hadamard gate applied to the ancilla qubit mixes the two subspaces together and creates an interference between the two quantum walks. Finally, all qubits are measured. The measurement of q_a collapses the network to one of two possible cases, imposing either a sum or subtraction of the two conjugate evolutions, which encodes even powers of A (even predictions) for $q_a = |0\rangle$ and odd powers of A (odd predictions) for $q_a = |1\rangle$ [Eqs. (6) and (7)]. The measurement of the remaining n qubits returns a bit string marking a node i, which together with the initial node j forms a sample of a link (i, j) [Eq. (12)].

suitable to be tackled with a quantum computer is the realization that, in practice, we are not interested in knowing the scores of all pairs of nodes, but we simply wish to know which ones have the highest score up to a certain cutoff threshold, as illustrated in Fig. 1. By encoding the prediction scores in the amplitudes of a quantum superposition and performing quantum measurements on the system, the predictions with the highest score will be naturally sampled with higher probability, which can potentially be advantageous compared to the classical case of explicitly computing all scores as long as the quantum superposition can be efficiently prepared. We proceed now in Sec. III with the description of the quantum method and discuss in Sec. IV the expected resource complexity and show example comparisons with classical path-based methods.

III. QUANTUM LINK PREDICTION

We now describe our method for quantum link prediction (QLP), which we summarize at the end. We base our approach on a CTQW [11,12], where the Hilbert space of the quantum walker is defined by the orthonormal basis set $\{|j\rangle\}_{j \in \mathcal{V}}$, with each $|j\rangle$ corresponding to a localized state at a node j. We consider the Hamiltonian of the evolution as the adjacency

matrix of the graph, A. In Fig. 2 we show the main structure of the QLP circuit using a qubit representation. In the simplest case, we require $n = \log_2 N$ qubits to add a binary label to each of the N nodes, hereafter marked by the subscript n, and we consider an extra ancilla qubit q_a that doubles the Hilbert space of the quantum walk, such that any node j has two associated basis states

$$|0\rangle_a |j\rangle_n \text{ and } |1\rangle_a |j\rangle_n. \tag{4}$$

For an initial state $|\psi_j(0)\rangle = |0\rangle_a |j\rangle_n$, the first step in the circuit of Fig. 2 is to apply an Hadamard gate to q_a, which creates the superposition

$$\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)_a |j\rangle_n. \tag{5}$$

A conditional CTQW is then applied which evolves the $q_a = |0\rangle$ subspace with e^{-iAt} and the $q_a = |1\rangle$ subspace with e^{+iAt}. Finally, a second Hadamard gate is applied to q_a to interfere with the two quantum walks in the computational basis,

leading to the state

$$|\psi_j(t)\rangle = |0\rangle_a \left(\frac{e^{-iAt} + e^{iAt}}{2}\right)|j\rangle_n$$
$$+ |1\rangle_a \left(\frac{e^{-iAt} - e^{iAt}}{2}\right)|j\rangle_n. \tag{6}$$

To make the connection with the link prediction more evident, we rewrite the previous expression as

$$|\psi_j(t)\rangle = |0\rangle_a \left(\sum_{k=0}^{+\infty} c_{\text{even}}(k,t) A^{2k}\right)|j\rangle_n$$
$$+ i|1\rangle_a \left(\sum_{k=0}^{+\infty} c_{\text{odd}}(k,t) A^{2k+1}\right)|j\rangle_n, \tag{7}$$

where we replaced the exponential terms with their respective power series and defined the time-dependent coefficients as

$$c_{\text{even}}(k,t) = (-1)^k t^{2k}/(2k)!, \tag{8}$$

$$c_{\text{odd}}(k,t) = (-1)^{k+1} t^{2k+1}/(2k+1)!. \tag{9}$$

A detailed calculation leading to Eq. (7) can be found in Appendix A. Given some initial node j, Eq. (7) describes the state that is created following the QLP circuit, before measurement. This state has two entangled components, one with a linear combination of even powers of A for $q_a = |0\rangle$ and another with odd powers of A for $q_a = |1\rangle$. The time t of the quantum walk defines the linear weights and acts as a hyperparameter in the model. This describes the unitary part of the protocol. To obtain relevant predictions from this state we must perform repeated measurements on the system to draw multiple samples, as we now describe.

The first step is to measure q_a, yielding $|0\rangle$ or $|1\rangle$ and collapsing the state of the remaining qubits to

$$|\psi_j(t)\rangle_n^{\text{even}} \propto \left(\sum_k c_{\text{even}}(k,t) A^{2k}\right)|j\rangle, \text{ or} \tag{10}$$

$$|\psi_j(t)\rangle_n^{\text{odd}} \propto \left(\sum_k c_{\text{odd}}(k,t) A^{2k+1}\right)|j\rangle, \tag{11}$$

respectively, where we omitted the normalization. This effectively selects whether the link sampled will be drawn from a distribution encoding even or odd powers of A. The last step is then to measure the remaining qubits, yielding a bit string corresponding to a sample of some node i with probability

$$p_{ij}^{\text{even}} \propto \left|\langle i|\left(\sum_{k=0}^{+\infty} c_{\text{even}}(k,t) A^{2k}\right)|j\rangle\right|^2,$$

$$p_{ij}^{\text{odd}} \propto \left|\langle i|\left(\sum_{k=0}^{+\infty} c_{\text{odd}}(k,t) A^{2k+1}\right)|j\rangle\right|^2, \tag{12}$$

which, together with the initial node j, forms a sample of a link (i, j). The values p_{ij}^{even} and p_{ij}^{odd} encode the prediction scores of the link (i, j), but these cannot be directly extracted from the algorithm. Instead, what this algorithm allows is the repeated sampling of these distributions, yielding pairs of nodes (i, j) with probability proportional to p_{ij}^{even} or p_{ij}^{odd}.
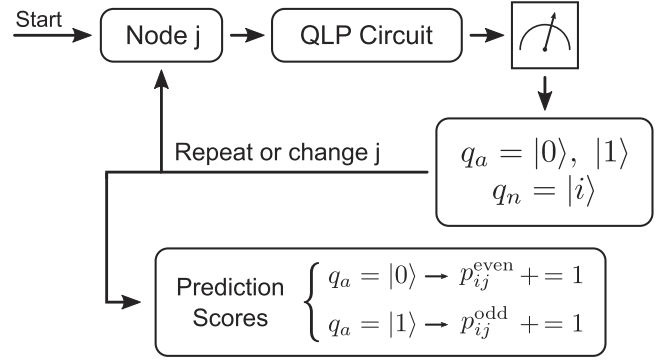

FIG. 3. QLP algorithm. Starting with an initial node j, the QLP circuit samples a node i corresponding to an even or odd prediction of a link (i, j) according to the value measured in q_a. Repeating this procedure for each node j allows the larger values of the even and odd predictions scores p_{ij} to be approximated.

This is analogous to sampling entries (i, j) from the matrix of prediction scores P with probability proportional to $|P_{ij}|^2$. As discussed in Sec. II, predictions coming from even or odd powers of A are typically useful in different types of networks. For a given network application of QLP, whether each sample obtained corresponds to an even or odd prediction depends on the value measured in the ancilla qubit and this postselection can only be done probabilistically [31]. This is a potential sampling overhead, as unwanted predictions need to be discarded. Another overhead to consider is the possibility of sampling the initial node, or to sample already existing links, given the contribution of the identity I in p_{ij}^{even} and A in p_{ij}^{odd}, which must also be discarded. As stated, QLP uses a linear combination of powers of A weighted by the time t. As already mentioned, a classical prediction method with a linear combination of odd powers of A was presented in Ref. [28], which was shown to sometimes improve the prediction precision compared to the original L3 method from Ref. [19] by also fitting an additional model parameter. Another popular link prediction method is the Katz index [32], which uses a linear combination of all powers of A.

We can now summarize the QLP algorithm, as illustrated in Fig. 3. First, an initial state $|\psi_j(0)\rangle = |0\rangle_a|j\rangle_n$ is prepared for a node j in the network. Second, the QLP evolution leading to Eq. (7) is performed for a specific time t. Finally, the ancilla and node qubits are measured to obtain a sample of a link (i, j) corresponding to an even or odd prediction, and the procedure is repeated. The number of samples that output a certain link (i, j) will follow the distributions described by Eq. (12), and thus represent a score for link (i, j). Once predictions associated with node j are sufficiently characterized, the procedure can be repeated for other relevant nodes in the network.

IV. RESULTS AND DISCUSSION

A. Complexity analysis

To identify a potential quantum advantage, we briefly discuss how link prediction scales on a classical computer. Complex networks are typically sparse [1] with the
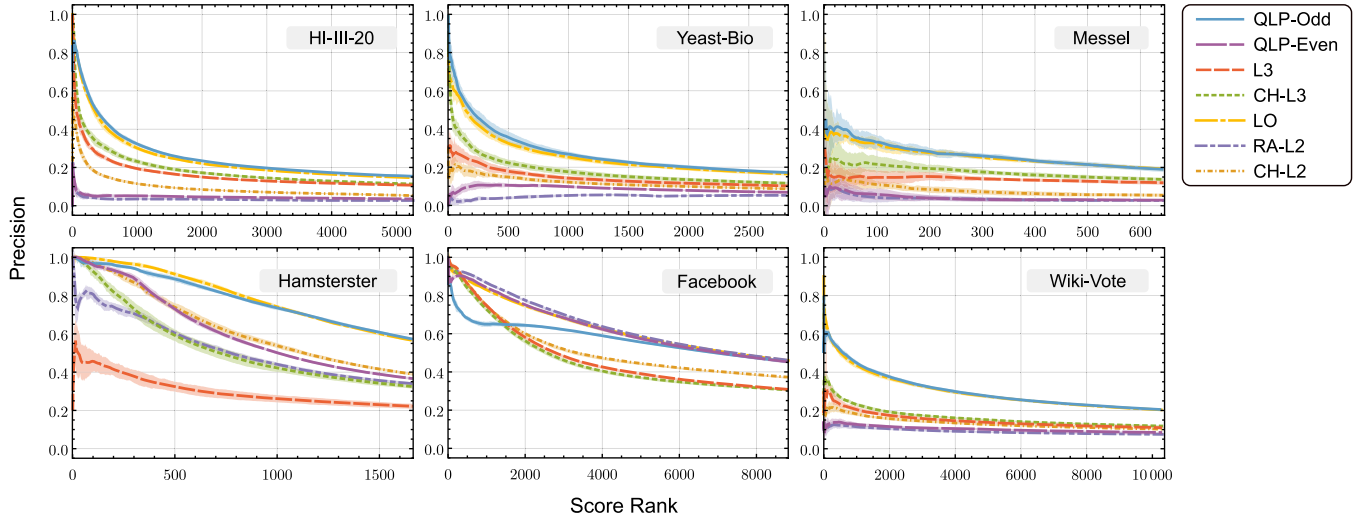
FIG. 4. Computational cross-validation: Precision. Cumulative precision over the list of ranked scores for each network, averaged over a ten-fold cross validation procedure. The shaded regions correspond to the standard deviation. Details on the precision and score rank metrics are described in the text. The networks used correspond to the PPI networks HI-III-20, the most recent PPI mapping of the human interactome [20], Yeast-Bio, a PPI network of a yeast organism [35], Messel, a food web [36], Hamsterster [37] and Facebook [38], two online social networks, and Wiki-Vote, a vote network between users for adminship of Wikipedia [39]. For comparison, we implemented five classical link prediction methods: the L3 method [19], the LO method [28], the CH-L3 method [29], and two even power methods, RA-L2 (resource allocation) [40], and CH-L2 [29,41]. The dataset parameters characterizing each network are shown in Table II and the values selected for the optimal parameters t in the QLP method and α in the LO method are shown in Table III.

average degree much smaller than the total number of nodes, $k_{av} \ll N$, and as such there are $O(N^2)$ potentially missing links. The general case of computing all possible scores leads to a classical complexity of at least $O(N^2)$. Different methods scale differently depending on the assumptions made about the solution. For example, the scaling of simple length-2 based methods is $O(N\langle k^2 \rangle)$ and the scaling of L3 [19] is upper bounded by $O(N\langle k^3 \rangle)$, where $\langle k^n \rangle$ is the average of the nth power of the degrees (see Appendix B). These methods do not calculate a score for every possible missing link, only for those corresponding to nodes at distance 2 or 3. However, other methods also surpass the $O(N^2)$ scaling, as is the case

of LO [28] that uses a matrix inversion to represent a linear combination of odd powers of A, scaling approximately with $O(N^{2.4})$, and is one of the best performing classical methods tested. Complex networks can easily reach sizes of up to millions or billions of nodes, consider, for example, online social and e-commerce networks, or the neuronal network in the human brain [33]. Improving these scalings may thus be decisive in the application of link prediction methods to larger networks in the future.

To provide an estimate for the complexity of implementing QLP on a quantum computer, there are a few things to consider. First, we comment on the implementation of the

TABLE I. Computational cross-validation: AUC. Area under the curve (AUC) performance metrics for the datasets in Fig. 4. The metrics were computed over the full set of potential predictions for each dataset in a ten-fold cross-validation procedure. Each value corresponds to the mean AUC over the ten iterations. In bold are the best values for both QLP and the classical methods tested, confirming that QLP performs similarly to other path-based link prediction methods in the AUC-ROC and AUC-PR metrics.

	Dataset	QLP-Even	QLP-Odd	LO	L3	CH-L3	RA-L2	CH-L2
AUC-ROC	HI-III-20	0.786	**0.909**	0.879	**0.917**	0.917	0.655	0.655
	Yeast-Bio	0.878	**0.894**	0.852	**0.905**	0.904	0.738	0.738
	Messel	0.635	**0.887**	0.880	**0.891**	0.890	0.641	0.649
	Hamsterster	**0.971**	0.964	0.952	0.965	**0.966**	0.962	0.962
	Facebook	**0.995**	0.994	0.988	0.991	0.991	**0.995**	0.994
	Wiki-Vote	0.878	**0.904**	0.898	**0.905**	0.905	0.858	0.859
AUC-PR	HI-III-20	0.006	**0.081**	0.074	0.042	0.049	0.005	0.013
	Yeast-Bio	0.014	**0.093**	0.082	0.038	0.049	0.013	0.024
	Messel	0.008	**0.104**	0.104	0.051	0.062	0.007	0.013
	Hamsterster	0.341	**0.568**	0.574	0.131	0.280	0.284	0.365
	Facebook	**0.429**	0.392	0.427	**0.444**	0.334	0.262	0.257
	Wiki-Vote	0.0287	**0.112**	0.111	0.026	0.037	0.043	0.047

e^{-iAt} unitary, representing the CTQW used to obtain each link prediction sample. For this purpose, the most relevant results are related to the quantum simulation of d-sparse matrices, meaning that A has at most d entries in any given row. A state-of-the-art result [34] shows that in that scenario implementing e^{-iAt} scales as $\tilde{O}(dt\|A\|_{\max})$, omitting logarithmic factors, where t is the time interval of the evolution and $\|A\|_{\max}$ is the maximum entry in absolute value. In our case, $d = k_{\max}$, the maximum degree of the network, and $\|A\|_{\max} = 1$, which allows us to write the complexity of implementing e^{-iAt} as $\tilde{O}(k_{\max}t)$. Second, we comment on the time t. As mentioned earlier, t is a hyperparameter in the model which determines how each power of A is weighted for the predictions. A large value of t would lead higher powers of A to be more heavily weighted. This is not the case for typical link prediction methods, where the most relevant contributions are typically from A^2 or A^3, irrespective of the network size. In our simulations we found the optimal value of t to change from network to network, however, it seems to do so independently of N (see Table III). For these reasons, we believe it is reasonable to disregard the contribution of t to the complexity. Finally, for each application of the circuit from Fig. 2 a link prediction sample associated with a node j is obtained. Then, assuming a repetition of the process to obtain samples for every node leads to a factor of N in the complexity, and for each node j a sufficient number of samples s_j is required to characterize the predictions associated with it. If we consider that the missing links have been removed randomly from the network, each node j will have a number of missing links proportional to its observed degree k_j. However, the relation between s_j and k_j is highly network-dependent, as it depends on how well the quantum walk method represents the underlying truth of the missing links. In practice, we can leave s_j as a free parameter, as ultimately the number of samples to obtain would be decided by the user.

In summary, assuming a repetition of the QLP method for each node in a network of N nodes, with an average number of samples per node of $s_{\mathrm{av}} = \frac{1}{N}\sum_{j=1}^{N} s_j$, and each sample requiring the implementation of e^{-iAt} with a cost $\tilde{O}(k_{\max})$ for some constant t, the final complexity estimate for QLP is

$$\tilde{O}(Ns_{\mathrm{av}}k_{\max}). \tag{13}$$

The most meaningful complexity comparison we can make is between methods that make similar assumptions. In that sense, both QLP and LO assume the solution is a linear combination of powers of the adjacency matrix, and as we will see in the next section, these methods are often the best performing. Here we can see that QLP has a potential quantum speedup given the polynomially lower dependence on N but with an extra $s_{\mathrm{av}}k_{\max}$ factor. Relating $k_{\max}$ to N can be done through γ as $k_{\max} \propto N^{\frac{1}{\gamma-1}}$, where γ is the exponent in the power-law degree distribution of a scale-free network, which is typically in the range $2 < \gamma \leqslant 4$ [1]. For these values the dependence is always sublinear, approaching linearity as $\gamma \to 2$, as in this regime the network tends to form larger and larger hubs. For $\gamma = 3$, for example, our estimate for the scaling of QLP in scale-free networks is $\tilde{O}(N^{3/2}s_{\mathrm{av}})$, a potential polynomial speedup over the LO method. Comparing QLP to simple length-2 and length-3 based methods is less straightforward,

as the difference is solely based on the degree factors and the number of samples for QLP.

B. Cross-validation tests

1. Description of metrics

In Fig. 4 we show the cumulative precision over the score rank for each method and network tested. The score rank represents the ordered list of scores: the top score has score rank 0, the second best has score rank 1, and so on. The cumulative precision tracks the ratio of correct predictions to total predictions over all previous score ranks. For example, a precision of 0.8 at score rank 9 means that out of the ten top predictions occupying score rank 0 through 9, eight of them were correct. For each iteration of the ten-fold cross validation procedure 10% of the links were randomly removed and the remainder used as input to the link prediction methods, leading to a different cumulative precision curve for each iteration. In the plots we show the average cumulative precision ± one standard deviation over the ten iterations. Since the networks tested have different sizes and densities the total number of predictions that may be considered relevant varies. We chose to cutoff the figure at a score rank of $0.05 \times N \times k_{\mathrm{av}}$, which is sufficient to show a drop in the precision over the score rank while still focusing on the precision of the best predictions occupying the first ranks.

In Table I we show the area under the curve (AUC) for the receiver-operating characteristic cruve (AUC-ROC) and the precision-recall curve (AUC-PR), which are standard benchmark metrics in machine learning for predictive models.

2. Discussion

In Fig. 4 we compare the prediction precision of QLP with classical path-based link prediction methods using the standard link prediction benchmark of cross validation on a selection of networks from different fields. In Table I we further compare the methods using two standard AUC metrics, as described. A summarized description of the classical methods used can be found in Ref. [18]. We compared against three odd-power methods, L3 [19], LO [28], and CH-L3 [29], and two even-power methods, RA-L2 [40] and CH-L2 [29,41]. These are the state-of-the-art in local and global link prediction indices based on path counting. The scores used for QLP were an exact calculation of the distributions in Eq. (12) by classically computing the time-evolution operator of the quantum walk, as described in Appendix A. The networks used and respective parameters are summarized in Table II. For each network, we selected the time t that maximizes the prediction precision by removing 10% of the links from the training set in the first iteration of the cross validation, selecting both a value that maximizes the precision of the even component as well as one that maximizes the odd component, detailed in Table III. As shown in Fig. 4 and Table I, we confirm that QLP matches the typical performance of classical path-based link prediction methods tested in terms of prediction precision as well as standard AUC metrics for a range of real-life complex networks [20,35–39], as expected. In most cases, we observe that both QLP-Odd and LO stand out as the best performing methods, a result which further affirms the

TABLE II. Datasets and respective network parameters. The parameters listed are the following: Network parameters: $|V|$ is the number of nodes; $|E|$ is the number of links; k_{av} is the average degree; ρ is the network density; d_{max} is the maximum distance between any two nodes; d_{av} is the average distance between any two nodes; C is the average clustering coefficient.

	Network	Ref.	$|V|$	$|E|$	k_{av}	ρ	d_{max}	d_{av}	C
Fig. 4	HI-III-20	[20]	8275	52569	12.589	1.59×10^{-3}	12	3.844	5.92×10^{-2}
	Yeast-Bio	[35]	4885	28270	11.161	2.29×10^{-3}	10	3.603	1.20×10^{-1}
	Messel	[36]	700	6444	18.326	2.61×10^{-2}	6	2.632	1.04×10^{-1}
	Hamsterster	[37]	2426	16631	13.711	5.65×10^{-3}	10	3.589	5.38×10^{-1}
	Facebook	[38]	4039	88234	43.691	1.08×10^{-2}	8	3.693	6.06×10^{-1}
	Wiki-Vote	[39]	7115	103689	29.147	3.98×10^{-3}	7	3.248	1.41×10^{-1}
Fig. 5	Arabidopsis	[45]	4865	11374	4.493	9.24×10^{-4}	14	5.180	9.82×10^{-2}
	Pombe	[35]	1929	3700	3.397	1.76×10^{-3}	14	4.671	6.37×10^{-2}
	AS Routes	[46]	6474	13895	3.884	6.00×10^{-4}	9	3.705	2.52×10^{-1}
	Citeseer	[47]	3264	4536	2.779	8.518×10^{-4}	28	9.315	1.45×10^{-1}
	Cora	[47]	2708	5429	4.010	1.44×10^{-3}	19	6.311	2.41×10^{-1}
	P2P-Gnutella	[46]	10876	39994	7.355	6.78×10^{-4}	10	4.622	6.22×10^{-3}
Fig. 6	Screen 1	[20]	4643	16447	6.970	1.64×10^{-3}	12	4.094	5.13×10^{-2}
	Screen 2	[20]	4177	11644	5.467	1.31×10^{-3}	13	4.284	4.33×10^{-2}
	Screen 3	[20]	3807	10245	5.268	1.31×10^{-3}	15	4.456	4.16×10^{-2}
	Screen 4	[20]	3082	5685	3.655	1.19×10^{-3}	14	5.370	1.51×10^{-2}
	Screen 5	[20]	2712	4496	3.277	1.21×10^{-3}	15	5.560	1.25×10^{-2}
	Screen 6	[20]	3128	5981	3.774	1.21×10^{-3}	16	5.361	1.54×10^{-2}
	Screen 7	[20]	3508	7910	4.486	1.28×10^{-3}	15	5.465	9.73×10^{-3}
	Screen 8	[20]	3383	7533	4.436	1.31×10^{-3}	16	5.555	1.20×10^{-2}
	Screen 9	[20]	3404	7712	4.512	1.33×10^{-3}	15	5.520	9.40×10^{-3}

case that there can be advantages in including higher-order powers of the adjacency matrix in the predictions [28]. While optimizing over t allows us to show the maximum precision of the method, we note that even with hand-picked values the method tends to perform well in comparison with other path-based methods without a free parameter. Further results for the cross-validation benchmark are shown in Fig. 5 and Table IV, as well as detailed results for each of the experimental screens that contribute to the full HI-III-20 network in Fig. 6. Here we predict interactions that were obtained by independent, full experimental screens, simulating the case of real-life performance against future experiments.

V. CONCLUSION

In this work we presented a quantum algorithm for link prediction in complex networks, QLP, offering a potential quantum speedup for a practical network science problem. The inclusion of even and odd paths allows QLP to make both TCP-like and L3-like predictions, thus encompassing all types of networks where these topological patterns play a role. Our results serve as a proof of principle for potential future applications of QLP in large complex networks using quantum hardware. Recently, a 62-node network CTQW was demonstrated experimentally [42], an important first step towards this goal.

We note that, in our estimated complexity, the dependence on k_{max} comes from assuming a circuit-based simulation of the quantum walk with the d-sparse matrix model. We may argue that the existence of large hubs make complex networks a bad fit for the d-sparse matrix model. Highly connected nodes imply that some rows in the adjacency matrix are very dense, while most are sparse, and thus $d = k_{max}$ greatly overestimates the overall sparseness of the matrix. Finding a more efficient quantum simulation algorithm that directly exploits the degree distribution of complex networks would be a very important result for quantum computation applied to network science, should such a method exist. Nevertheless, the method we propose here is general to any representation of the quantum walk, for example, using an analog quantum walk implementation, as done in Ref. [42], or any future quantum simulation techniques that prove to be more efficient.

Besides the potential improvement in complexity when sampling from the quantum solution, especially in the comparison between QLP and LO, we should also note that a classical simulation of QLP relies on the diagonalization of the adjacency matrix and thus it has a comparable classical complexity to other path-based classical link prediction methods. This makes QLP easier to be further developed with a focus on immediately relevant real-world applications, while at the same time exploring other ways in which quantum features of QLP can be advantageous when quantum hardware becomes more widely available.

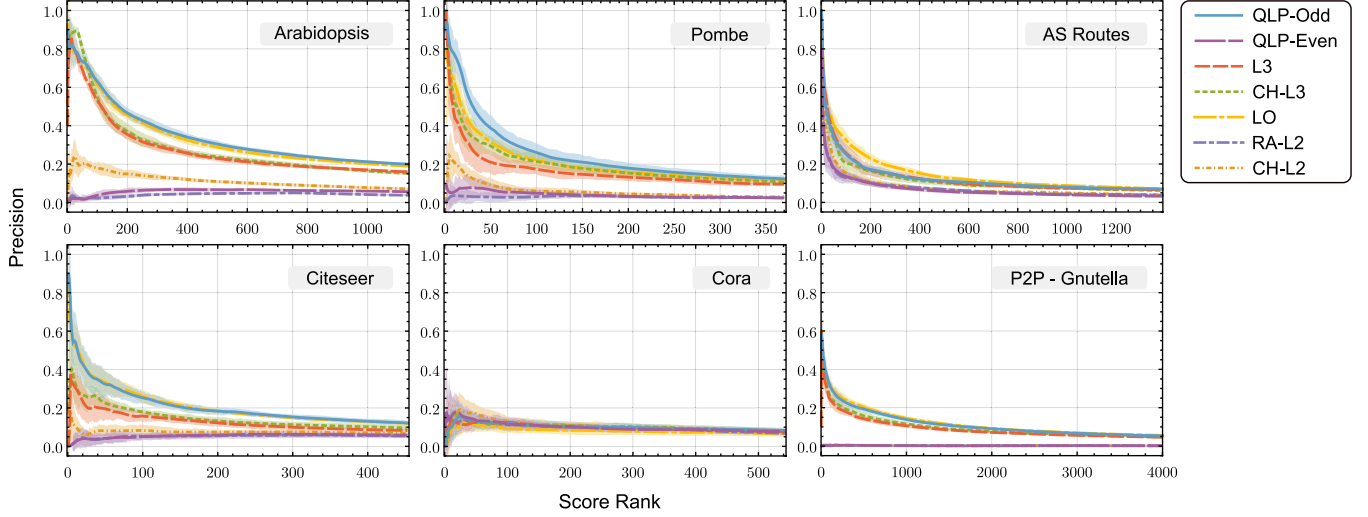Our code for QLP is available on GitHub , see Ref. [43].

FIG. 5. Extra cross-validation results: Cumulative precision over the top-ranked predictions (the top $0.05 * Nk_{av}$ scores) for each network, averaged over a ten-fold cross-validation procedure. The shaded regions correspond to the standard deviation. In each trial 10% of the links were randomly removed and the remainder were used as input to the link prediction methods. The networks used correspond to the PPI network Arabidopsis [45], Pombe [35], AS Routes [46], Citeseer [47], Cora [47], and P2P-Gnutella [46]. We note that P2P-Gnutella is a bipartite network and thus the TCP based methods QLP-Even, RA-L2 and CH-L2 have null precision. The dataset parameters characterizing each network are shown in Table II, and the values selected for the optimal 610 parameters t in the QLP method and α in the LO method are shown in Table III.

APPENDIX A: QLP METHOD

We consider the usual CTQW model, where the Hilbert space of the quantum walker is defined by the orthonormal basis set $\{|j\rangle\}_{j\in\mathcal{V}}$, each basis state $|j\rangle$ corresponding to a localized state at a node j in the network and the Hamiltonian of the evolution given by the adjacency matrix of the graph. In these conditions, the solution to the Schrödinger equation for the CTQW can be written directly as

$$|\psi(t)\rangle = e^{-iAt}|\psi(0)\rangle. \tag{A1}$$

By taking the power series of the time evolution operator we can immediately make the connection to the link prediction

$$e^{-iAt} = \sum_{k=0}^{+\infty} \frac{1}{k!}(-it)^k A^k, \tag{A2}$$

as each power A^k encodes the number of paths of length k between any two nodes in the graph. Furthermore, we note that the imaginary term i^k adds a phase to the quantum evolution that separates the sum over even powers in the real part of the evolution and the sum over odd powers in the imaginary part. To proceed we wish to separate the evolution over even powers from the evolution over odd powers, and for that it is useful to consider a qubit representation of the graph, as seen in Fig. 2 of the main text. We now define an operator $\mathrm{CQW}(t)$ corresponding to a controlled quantum walk, which applies a normal or conjugate evolution operator on the node qubits depending on the value of q_a,

$$\mathrm{CQW}(t) = |0\rangle\langle 0|_a (e^{-iAt})_n + |1\rangle\langle 1|_a (e^{+iAt})_n. \tag{A3}$$

Considering now an initial state localized at node j in the form $|\psi_j(0)\rangle = |0\rangle_a|j\rangle_n$ we start by applying an Hadamard gate on the ancilla qubit

$$H_a|0\rangle_a|j\rangle_n = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)_a|j\rangle_n, \tag{A4}$$

followed by the $\mathrm{CQW}(t)$ operator

$$\mathrm{CQW}(t)\left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)_a|j\rangle_n\right]$$
$$= \frac{1}{\sqrt{2}}(|0\rangle_a e^{-iAt}|j\rangle_n + |1\rangle_a e^{+iAt}|j\rangle_n), \tag{A5}$$

followed by a second Hadamard gate on the ancilla qubit, leading to the following expression after rearranging the terms:

$$|\psi_j(t)\rangle = \tfrac{1}{2}|0\rangle_a(e^{-iAt} + e^{iAt})|j\rangle_n + |1\rangle_a(e^{-iAt} - e^{iAt})|j\rangle_n. \tag{A6}$$

Finally, taking the power series from Eq. (A2) to replace the exponential terms we arrive at

$$|\psi_j(t)\rangle = |0\rangle_a \left(\sum_{k=0}^{+\infty} c_{\text{even}}(k,t) A^{2k}\right)|j\rangle_n$$
$$+ i|1\rangle_a \left(\sum_{k=0}^{+\infty} c_{\text{odd}}(k,t) A^{2k+1}\right)|j\rangle_n, \quad \text{(A7)}$$

with $c_{\text{even}}(k,t) = (-1)^k t^{2k}/(2k)!$ and $c_{\text{odd}}(k,t) = (-1)^{k+1} t^{2k+1}/(2k+1)!$ being time-dependent coefficients.

This procedure describes how the real and imaginary parts of the time-evolution operator can be separated on a quantum computer through an extra ancilla qubit. To simulate QLP on a conventional computer, it suffices to compute the time-evolution operator and directly extract the real and imaginary parts, as described below.

QLP on a conventional computer

In this section we describe how to directly compute the scores of QLP on a conventional computer. Consider a network described by its adjacency matrix A. Start by picking a value for the t parameter. Then numerically compute the time-evolution operator of the quantum walk

$$U(t) = e^{-iAt}, \quad \text{(A8)}$$

for example, by computing the eigenvalues and eigenvectors of A and then computing the matrix exponential. The matrix $U(t)$ is complex. The prediction scores described in the main text can then be obtained directly, in matrix form, as

$$P^{\text{even}} = |\text{Re}(e^{-iAt})|^2_{(ij)},$$
$$P^{\text{odd}} = |\text{Im}(e^{-iAt})|^2_{(ij)}, \quad \text{(A9)}$$

where $|.|^2_{ij}$ denotes the entry-wise absolute value squared. The entries of P^{even} and P^{odd} correspond to the p_{ij}^{even} and p_{ij}^{odd} values described in the main text, which can be used to rank predictions from highest to lowest score.

APPENDIX B: LINK PREDICTION COMPLEXITY

Consider a graph $G(\mathcal{V}, \mathcal{E})$ describing a complex network, where $\mathcal{V}$ is the set of nodes with size $N = |\mathcal{V}|$ and $\mathcal{E}$ is the set of undirected links. Link prediction on a classical computer requires $\frac{1}{2}N(N-1) - |\mathcal{E}|$ scores to be computed, one for each of the $\frac{1}{2}N(N-1)$ possible links, with the exception of those already present in the set of known links $\mathcal{E}$. Rewriting in terms of the average degree, $k_{\text{av}} = 2|\mathcal{E}|/N$, we have that the total number of scores scales as $\frac{1}{2}N^2 - \frac{1}{2}N(1 + k_{\text{av}})$. Real complex networks are typically sparse [1] with $k_{\text{av}} \ll N$, and thus $O(N^2)$ scores are evaluated. Taking $O(N^2)$ as an estimate for the complexity of a general classical link prediction method assumes two more things: that the method will indeed compute a score for every potential missing link, and that the cost of computing each score is $O(1)$. To analyze these assumptions, let us pick a concrete method and study its complexity.

COMMON NEIGHBOURS (CN) is one of the simplest link prediction algorithms. It quantifies the likelihood of a link existing between two nodes i and j by the number of common neighbors they share, or in other words, by the number of paths of length 2 between i and j. While we do not use CN directly in the various simulations presented in this work, we used the method of resource allocation [40] (marked as RA-L2 in the plots), which is similar to CN with the addition of a degree normalization to each score. Adding the degree normalization does not affect the complexity significantly, and so we will analyze the simpler problem of counting paths of length 2. The objective of CN is to compute

$$p_{ij} = |\Gamma(i) \cap \Gamma(j)| \quad \text{(B1)}$$

for every pair of nodes (i, j) where $|\Gamma(i) \cap \Gamma(j)| \neq 0$, with $\Gamma(x)$ being the set of nodes neighboring x. A simple algorithm to accomplish this iterates through all nodes z in the graph and adds a contribution to p_{ij} for each pair of nodes (i, j) neighboring z. Such an algorithm will visit every path of length 2 in the graph and thus its complexity will be proportional to $\sum_{i,j=1}^{N}(A^2)_{ij}$. As detailed in Ref. [44] this sum can be simplified as

$$\sum_{i,j=1}^{N}(A^2)_{ij} = \sum_{i=1}^{N} k_i^2 = N\langle k^2 \rangle, \quad \text{(B2)}$$

where $\langle k^2 \rangle$ is the average of the second power of the degrees in the graph. By assuming that the cost of accessing the graph data structure and adding the contributions to each p_{ij} is $O(1)$ we can conclude that the CN method scales as $O(N\langle k^2 \rangle)$.

COMMON NEIGHBOURS is a TCP-based method, and as discussed in the main text, it is not able to match the precision of methods based on paths of length 3 in many networks. For that reason, let us see how the complexity changes when counting paths of length 3, which is the main computational cost behind the L3 method [19]. An algorithm to count paths of length 3 can be easily built with an extension of the CN algorithm, and using the same argument as before, its complexity will be proportional to $\sum_{i,j=1}^{N}(A^3)_{ij}$. This sum is not as easy to simplify, but the authors of Ref. [44] proved the following bound for a general power of A

$$\sum_{i,j=1}^{N}(A^n)_{ij} \leqslant \sum_{i=1}^{N} k_i^n = N\langle k^n \rangle. \quad \text{(B3)}$$

With this information we can conclude that the complexity for counting paths of length 3 will be upper bounded by $O(N\langle k^3 \rangle)$.

For scale-free networks, characterized by γ, the degree exponent in the degree power-law distribution, we can analyze the moments $\langle k^n \rangle$ in terms of γ and N (see Sec. 4 of Ref. [1]). Typically, $\langle k \rangle$ (denoted as k_{av} in the rest of the text) is much smaller than $\langle k^2 \rangle$ or $\langle k^3 \rangle$. For many scale-free networks γ is between 2 and 4. As N grows, $\langle k^2 \rangle$ diverges for $2 < \gamma \leqslant 3$ and $\langle k^3 \rangle$ diverges for $2 < \gamma \leqslant 4$, while $\langle k \rangle$ does not. These divergences can be seen in the expressions below from Ref. [1], which estimated the dependence of $\langle k^n \rangle$ with N

$$\langle k^n \rangle \propto \frac{k_{\text{max}}^{n-\gamma+1} - k_{\text{min}}^{n-\gamma+1}}{n - \gamma + 1}, \quad \text{(B4)}$$

TABLE III. Hyperparameter values used for each dataset.

	Network	t (QLP-Even)	t (QLP-Odd)	α (LO)
Fig. 4	HI-III-20	3.0×10^{-6}	1.00	8.0×10^{-3}
	Yeast-Bio	7.0×10^{-1}	1.10	2.0×10^{-2}
	Messel	2.0×10^{-6}	1.20	2.0×10^{-2}
	Hamsterster	4.0×10^{-1}	1.60	3.0×10^{-2}
	Facebook	2.0×10^{-1}	1.00	7.0×10^{-3}
	Wiki-Vote	2.0×10^{-6}	1.00	4.0×10^{-3}
Fig. 5	Arabidopsis	2.0×10^{-6}	1.00	2.0×10^{-2}
	Pombe	9.0×10^{-1}	1.00	4.0×10^{-2}
	AS Routes	5.0×10^{-4}	0.60	2.0×10^{-3}
	Citeseer	5.0×10^{-5}	1.40	8.0×10^{-2}
	Cora	1.0×10^{-6}	0.03	6.0×10^{-2}
	P2P-Gnutella	-	1.30	2.0×10^{-2}
Fig. 6	Screen 1	6.0×10^{-6}	1.10	1.0×10^{-2}
	Screen 2	4.0×10^{-6}	0.90	8.0×10^{-3}
	Screen 3	6.0×10^{-5}	0.90	1.0×10^{-2}
	Screen 4	9.0×10^{-5}	0.02	1.0×10^{-2}
	Screen 5	6.0×10^{-5}	0.03	3.0×10^{-3}
	Screen 6	1.0×10^{-6}	0.03	7.0×10^{-3}
	Screen 7	1.0×10^{-6}	0.70	7.0×10^{-4}
	Screen 8	2.0×10^{-6}	0.30	4.0×10^{-4}
	Screen 9	1.0×10^{-6}	0.80	6.0×10^{-4}

which together with the relation $k_{\max} = k_{\min} N^{\frac{1}{\gamma - 1}}$ can be written as

$$\langle k^n \rangle \propto \frac{k_{\min}^{n-\gamma+1}}{n - \gamma + 1} \left(N^{\frac{n-\gamma+1}{\gamma-1}} - 1 \right). \quad \text{(B5)}$$

Out of the methods tested in this work, RA-L2 and L3 fall in the complexity categories of counting paths of length 2 and 3, respectively. CH-L2 and CH-L3 also have path counting as a base, but use a more complex structure of paths which has added complexity. LO, the best performing classical method tested, uses a matrix inversion for which the best algorithms scale roughly as $O(N^{2.4})$.

As stated in the main text, the complexity of QLP can be written as $\tilde{O}(Nk_{av}k_{\max}t)$. The previous expressions show that k_{av} remains finite for all $\gamma > 2$, while the higher-order moments can diverge. Although these expressions do not include any information about the finite value to which the moments tend when they do not diverge, complex networks are typically sparse, we may still use $k_{av} \ll N$ to quantify the differences in complexity between the methods, especially in the cases where $\langle k^2 \rangle$ and $\langle k^3 \rangle$ diverge with growing N. Furthermore, we can comment on the dependence with $k_{\max}$ coming from the d-sparse Hamiltonian simulation of the quantum walk. The relation $k_{\max} \propto N^{\frac{1}{\gamma-1}}$ [1] leads to $k_{\max} \propto N$ in the limit of $\gamma \to 2$, implying a quadratic scaling of QLP. This lower bound corresponds to an extreme case in scale-free networks and other larger values within the realistic $2 < \gamma < 4$ range reduce this dependence on N polynomially.

APPENDIX C: MODEL HYPERPARAMETERS

In Table III we present the values picked for the hyperparameter t for both the QLP-Even and QLP-Odd components, independently, and also the value of α used for the LO method [28] for each network. We omitted t for QLP-Even in P2P-Gnutella given that this is a bipartite network, and thus predictions based on even length paths have no meaning. The values were chosen by removing 10% of the links from the training set in the first iteration of the ten-fold cross-validation procedure and maximizing the prediction precision for those removed links.

One immediate observation is that the values of t for the predictions from QLP-Even and QLP-Odd are different by many orders of magnitude. From Eq. (3) in the main text we can write the prediction matrices for both components as follows:

$$P^{\text{even}} = |I - \frac{t^2}{2!}A^2 + \frac{t^4}{4!}A^4 - \frac{t^6}{6!}A^6 + \cdots|^2_{(ij)}, \quad \text{(C1)}$$

$$P^{\text{odd}} = |-tA + \frac{t^3}{3!}A^3 - \frac{t^5}{5!}A^5 + \frac{t^7}{7!}A^7 - \cdots|^2_{(ij)}, \quad \text{(C2)}$$

where $|.|^2_{(ij)}$ denotes the entry-wise absolute value squared. In both cases the first term does not contribute to the predictions. The small values of t in QLP-Even indicate that these predictions are best represented by the A^2 component in the

TABLE IV. Extra cross-validation results: AUC performance metrics for the datasets in Fig. 5, for both the receiver-operating curve (AUC-ROC) and precision-recall curve (AUC-PR). P2P-Gnutella is omitted due to a lack of computational resources to evaluate these metrics for the full network. The metrics were computed over the full set of potential predictions for each dataset in a ten-fold cross-validation procedure. Each value corresponds to the mean AUC over the ten iterations. The parameter values of t and α for the QLP and LO methods, respectively, were the same shown in Table III. In bold are the best values for both QLP and the classical methods tested.

|	Dataset	QLP-Even	QLP-Odd	LO	L3	CH-L3	RA-L2	CH-L2
AUC-ROC	Arabidopsis	0.661	**0.776**	0.792	**0.808**	**0.808**	0.574	0.574
	Pombe	0.670	**0.685**	0.727	**0.729**	**0.729**	0.563	0.563
	AS Routes	0.701	**0.725**	0.810	0.743	0.742	0.604	0.604
	Citeseer	0.638	**0.675**	0.703	**0.743**	**0.743**	0.672	0.672
	Cora	0.706	**0.781**	0.679	**0.766**	**0.766**	0.709	0.709
AUC-PR	Arabidopsis	0.007	**0.127**	**0.119**	0.006	0.015	0.092	0.097
	Pombe	0.004	**0.070**	**0.052**	0.040	0.047	0.004	0.006
	AS Routes	0.007	**0.028**	**0.029**	0.026	0.023	0.008	0.011
	Citeseer	0.014	**0.049**	**0.046**	0.029	0.032	0.014	0.017
	Cora	0.023	**0.024**	0.017	0.022	0.023	0.022	**0.024**
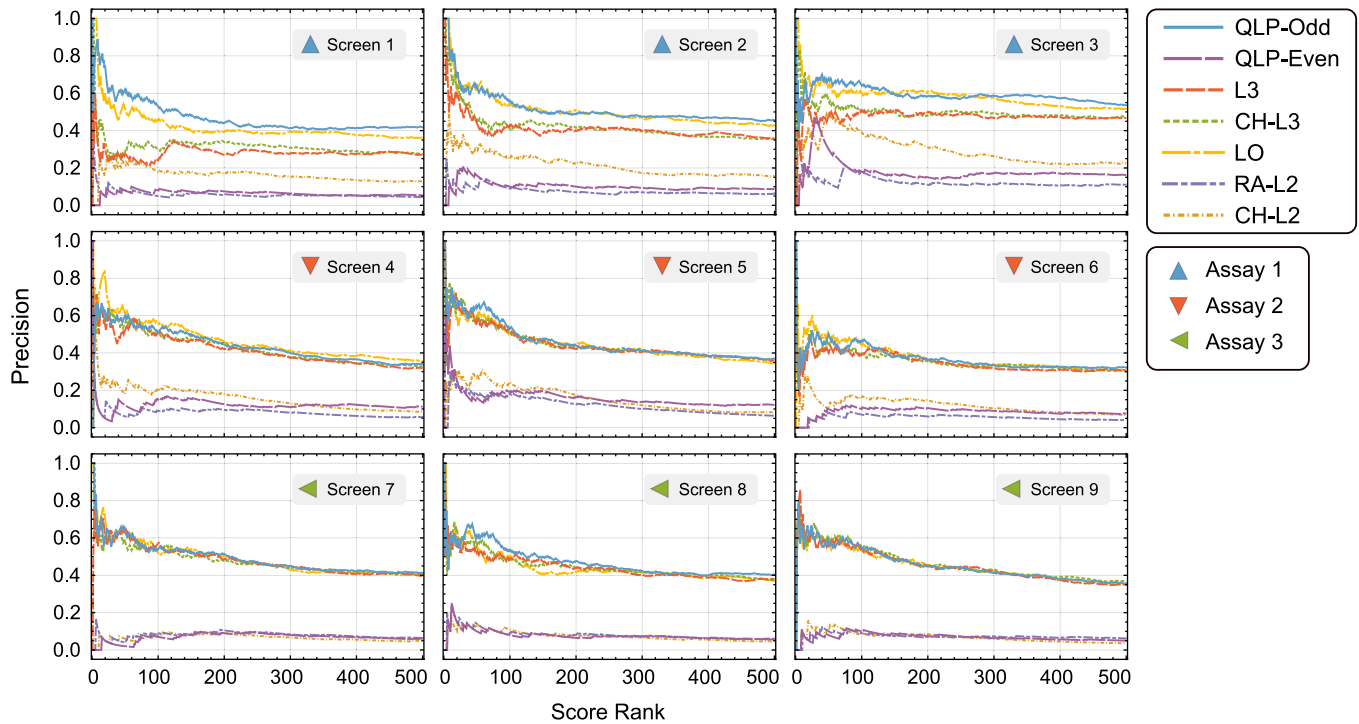
FIG. 6. Prediction of missing PPIs in the human interactome validated against experimental screens. Procedure described in Appendix D.

series, while these values observed for QLP-Odd indicate that these predictions typically benefit from the contributions of the higher-order powers of A.

APPENDIX D: VALIDATION AGAINST EXPERIMENTAL SCREENS

In Fig. 6 we validate the prediction from the QLP method against experimental protein-protein interaction screens. The results presented correspond to the cumulative precision over the top 500 ranked predictions. The dataset used consists of nine different screens over a search space of human binary PPIs using a panel of three different assay versions [20], the most recent experimental study of the human interactome network, named HI-III-20. In this study the authors presented a reference interactome map of human binary protein interac-

tions with 52 569 protein-protein interactions involving 8275 proteins. This map was generated by screening a search space of roughly 90% of the protein-coding genome a total of nine times with a panel of three different but complementary assay versions. For each of the nine plots we used the results of the respective screen as the input network to the link prediction methods and compared the predictions obtained to the PPIs identified in the remaining two screens from the same assay. For example, for the case of Screen 1, the predictions were compared with the PPIs identified in Screen 2 and Screen 3 combined. For the methods with a free parameter (QLP and LO) we randomly removed 50% of the input dataset and optimized the method to best predict the removed links by maximizing the area under the precision curve over the top 500 score ranks. The optimized parameter was then used for the results displayed.

[1] A.-L. Barabási, *Network Science* (Cambridge University Press, Cambridge, England, 2016).

[2] C. Dürr, M. Heiligman, P. Hoyer, and M. Mhalla, SIAM J. Comput. **35**, 1310 (2006).

[3] A. Ambainis and R. Špalek, *Annual Symposium on Theoretical Aspects of Computer Science* (Springer, New York, 2006), pp. 172–183.

[4] S. Chakraborty, L. Novo, A. Ambainis, and Y. Omar, Phys. Rev. Lett. **116**, 100501 (2016).

[5] S. Chakraborty, L. Novo, S. Di Giorgio, and Y. Omar, Phys. Rev. Lett. **119**, 220503 (2017).

[6] D. I. Tsomokos, Phys. Rev. A **83**, 052315 (2011).

[7] E. Sánchez-Burillo, J. Duch, J. Gómez-Gardenes, and D. Zueco, Sci. Rep. **2**, 605 (2012).

[8] M. Faccin, T. Johnson, J. Biamonte, S. Kais, and P. Migdał, Phys. Rev. X **3**, 041007 (2013).

[9] K. Mukai and N. Hatano, Phys. Rev. Res. **2**, 023378 (2020).

[10] M. Faccin, P. Migdał, T. H. Johnson, V. Bergholm, and J. D. Biamonte, Phys. Rev. X **4**, 041012 (2014).

[11] E. Farhi and S. Gutmann, Phys. Rev. A **58**, 915 (1998).

[12] J. Kempe, Contemp. Phys. **44**, 307 (2003).

[13] D. Liben-Nowell and J. Kleinberg, J. Am. Soc. Inf. Sci. **58**, 1019 (2007).

[14] P. Wang, B. W. Xu, Y. R. Wu, and X. Y. Zhou, Sci. Chin. Info. Sci. **58**, 1 (2015).

[15] I. Albert and R. Albert, Bioinformatics **20**, 3346 (2004).

[16] L. Getoor and C. P. Diehl, SIGKDD Explor. Newsl. **7**, 3 (2005).

[17] L. Lü and T. Zhou, Physica A **390**, 1150 (2011).

[18] T. Zhou, iScience **24**, 103217 (2021).

[19] I. A. Kovács, K. Luck, K. Spirohn, Y. Wang, C. Pollis, S. Schlabach, W. Bian, D.-K. Kim, N. Kishore, T. Hao *et al.*, Nat. Commun. **10**, 1240 (2019).

[20] K. Luck, D.-K. Kim, L. Lambourne, K. Spirohn, B. E. Begg, W. Bian, R. Brignall, T. Cafarelli, F. J. Campos-Laborie, B. Charloteaux *et al.*, Nature (London) **580**, 402 (2020).

[21] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, *SDM06: Workshop on Link Analysis, Counter-Terrorism and Security*, Vol. 30, (SIAM, Philadelphia, 2006), pp. 798–805.

[22] A. Ghasemian, H. Hosseinmardi, A. Galstyan, E. M. Airoldi, and A. Clauset, Proc. Natl. Acad. Sci. USA **117**, 23393 (2020).

[23] R. Guimerà and M. Sales-Pardo, Proc. Natl. Acad. Sci. USA **106**, 22073 (2009).

[24] L. Lü, L. Pan, T. Zhou, Y.-C. Zhang, and H. E. Stanley, Proc. Natl. Acad. Sci. USA **112**, 2325 (2015).

[25] T. Zhou, Y.-L. Lee, and G. Wang, Phys. A: Stat. Mech. Appl **564**, 125532 (2021).

[26] A. Muscoloni, U. Michieli, Y. Zhang, and C. V. Cannistraci, Preprints **2020**, 2020120808 (2020).

[27] A. Muscoloni and C. V. Cannistraci, Preprint **2021**, 2021050689 (2021).

[28] R. Pech, D. Hao, Y.-L. Lee, Y. Yuan, and T. Zhou, Phys. A: Stat. Mech. Appl **528**, 121319 (2019).

[29] A. Muscoloni, I. Abdelhamid, and C. V. Cannistraci, bioRxiv 346916 (2018).

[30] M. Kitsak, arXiv:2003.06665.

[31] R. Kothari, *Efficient Algorithms in Quantum Query Complexity* (University of Waterloo, Waterloo, Ontario, 2014).

[32] L. Katz, Psychometrika **18**, 39 (1953).

[33] F. A. Azevedo, L. R. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. Ferretti, R. E. Leite, W. J. Filho, R. Lent, and S. Herculano-Houzel, J. Comp. Neurol. **513**, 532 (2009).

[34] G. H. Low and I. L. Chuang, Phys. Rev. Lett. **118**, 010501 (2017).

[35] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, Nucleic Acids Res. **34**, D535 (2006).

[36] J. A. Dunne, C. C. Labandeira, and R. J. Williams, Proceedings of the Royal Society B: Biological Sciences **281**, 20133280 (2014).

[37] J. Kunegis, *Proceedings of the 22nd International Conference on World Wide Web* (ACM, New York, 2013), pp. 1343–1350.

[38] J. J. McAuley and J. Leskovec, *NIPS*, Vol. 2012, (Citeseer, University Park, PA, 2012), pp. 548–56.

[39] J. Leskovec, D. Huttenlocher, and J. Kleinberg, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (ACM, New York, 2010), pp. 1361–1370.

[40] T. Zhou, L. Lü, and Y.-C. Zhang, Eur. Phys. J. B **71**, 623 (2009).

[41] C. V. Cannistraci, G. Alanis-Lobato, and T. Ravasi, Sci. Rep. **3**, 1613 (2013).

[42] M. Gong, S. Wang, C. Zha, M.-C. Chen, H.-L. Huang, Y. Wu, Q. Zhu, Y. Zhao, S. Li, S. Guo *et al.*, Science **372**, 948 (2021).

[43] https://github.com/jpmoutinho/Quantum-Link-Prediction.

[44] M. A. Fiol and E. Garriga, Discrete Mathematics **309**, 2613 (2009).

[45] Arabidopsis Interactome Mapping Consortium *et al.*, Science **333**, 601 (2011).

[46] J. Leskovec, J. Kleinberg, and C. Faloutsos, ACM Trans. Knowl. Discov. Data **1**, 2 (2007).

[47] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, AI Magazine **29**, 93 (2008).