scientific reports

OPEN

On the complexity of quantum link prediction in complex networks

João P. Moutinho[1,2✉], Duarte Magano[1,2] & Bruno Coutinho[2]

Link prediction methods use patterns in known network data to infer which connections may be missing. Previous work has shown that continuous-time quantum walks can be used to represent path-based link prediction, which we further study here to develop a more optimized quantum algorithm. Using a sampling framework for link prediction, we analyze the query access to the input network required to produce a certain number of prediction samples. Considering both well-known classical path-based algorithms using powers of the adjacency matrix as well as our proposed quantum algorithm for path-based link prediction, we argue that there is a polynomial quantum advantage on the dependence on N, the number of nodes in the network. We further argue that the complexity of our algorithm, although sub-linear in N, is limited by the complexity of performing a quantum simulation of the network's adjacency matrix, which may prove to be an important problem in the development of quantum algorithms for network science in general.

Complex networks provide a common framework to study different complex systems[1]. Representing agents as nodes and interactions as links is a general enough description to fit many real systems, such as protein–protein interaction networks, social networks, transportation networks, electrical grids, and many others. Over the years, network science has lead to the realisation that these different systems share many common structural properties, and it is often possible to gain system-specific insights through general network-based problems and tools[2-6]. In the study of human disease, for example, the sub-field of Network Medicine has emerged from the success of network-based tools in problems such as the prediction of drug-combinations and cancer-driver genes[7-9].

One network problem with multidisciplinary applications is that of link prediction, which aims to infer new or unobserved links from a network based on its current or known topology[10-13]. In biological networks, link prediction has important applications in the identification of unknown protein–protein interactions[14], or aiding in the mapping of large scale neural networks[15,16], which better our understanding of human biology. In online social and commerce networks it can be used to suggest new friendships between users[10,12] or make product recommendations, increasing customer retention[17-19].

Link prediction is often a computationally intensive task. To make informed predictions, methods evaluate a certain score function over the whole set of potentially missing links to identify the few that stand out as the best predictions[13]. Recently, it has been suggested that the usage of quantum computers may help speed-up link prediction by sampling new links from a quantum walk evolution encoding the score values[20]. This result was one of the first examples of a quantum algorithm developed based on network science insights, with applications ranging from social network analysis to network medicine problems. The importance of quantum computing applied to network medicine and link prediction was further discussed in[21] and[22], respectively. Previous works in quantum walk algorithms have also tackled important network problems such as graph transversal[23,24] and marked node search[25-27].

In this work we further study the problem of link prediction and the path-based approach using Continuous-Time Quantum Walks (CTQW). First, we describe a sampling-based framework for link prediction, allowing us to make more precise resource comparisons between classical and quantum sampling-based algorithms. Previous results in classical algorithms have suggested that sampling-based algorithms may be advantageous for network-problems, including link prediction[28-30], which we can directly compare to. For this purpose, we use the standard benchmark of query complexity under a common input model.

Second, we provide an improved version of the quantum algorithm for link prediction initially put forward in Ref.[20] which allows links to be sampled globally from the network. Our proposed quantum algorithm produces link prediction samples with comparable precision to the studied classical algorithms by querying the input $\tilde{O}(k_{max})$ times per sample where k_{max} is the largest node degree in the network. In contrast, the classical algorithms studied typically require the full input to be queried for any number of samples, which naturally scales as $\tilde{O}(N)$, the number of nodes in the network. Using a scale-free model for complex networks[1] the growth of the

[1]Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal. [2]Instituto de Telecomunicações, Lisboa, Portugal. ✉email: joao.p.moutinho@tecnico.ulisboa.pt

largest degree in the network can be described as $k_{\max} \sim O\left(N^{\frac{1}{1-\gamma}}\right)$, with $\gamma > 2$. Thus, we argue that our quantum algorithm for link prediction achieves a polynomial speedup over the classical case in the input query complexity for a fixed number of samples.

Furthermore, we argue that the quantum complexity of our link prediction algorithm is limited by the complexity of performing a quantum simulation of the network's adjacency matrix. Finding an efficient quantum simulation algorithm for complex networks will have strong implications in the development of efficient quantum algorithms for network science problems in general. A first step towards this goal was recently demonstrated in Ref.[31], where an efficient quantum simulation algorithm was developed for sparse networks with a few densely connected nodes.

We organize our work as follows: in "Preliminaries" section we provide all definitions and necessary background for our work. In "Classical sampling algorithms" section we discuss relevant classical sampling-based link prediction algorithms and study their complexity. In "Quantum link prediction" section we provide our improved quantum algorithm for link prediction and study its complexity. In "Discussion and conclusions" section we further discuss our results, conclusions, and future work.

Preliminaries
Notation
We consider data organized in a simple, undirected and unweighted graph $\mathcal{G}(V, E)$, where V is the set of nodes with size $N = |V|$ and E is the set of links, with size $|E|$. For each node $v \in V$ we denote $\Gamma(v)$ as the set of nodes neighbouring v, and $\Gamma_l(v)$ as the l-th neighbour of v. The degree of v is defined as $k_v = |\Gamma(v)|$, and the average degree over the network is defined as $k_{\mathrm{av}} = 2|E|/N$. The adjacency matrix $A \in \mathbb{R}^{N \times N}$ is such that $A_{ij} = A_{ji} = 1$ if $(i, j) \in E$, and 0 otherwise.

We use the standard "big O" notation for asymptotic upper bounds. Given two functions f and g from $\mathbb{R}$ to $\mathbb{R}$ we say that $f = O(g)$ if there exists a constant C such that for any x greater than a threshold x_0 we have $f(x) < Cg(x)$. We further use the $\tilde{O}$ notation when omitting poly-logarithmic dependencies.

Link prediction
Link prediction is the general problem of inferring new or unobserved links from data organized in a networked structure. That is, given an adjacency matrix A of a complex network, we wish to select which unconnected pairs of nodes (i, j), i.e., pairs where $A_{ij} = 0$ and $i \neq j$, are the most likely to form a new link. The underlying assumption here is that there is enough information about the organizing principles of $\mathcal{G}$, or that this information can be correctly identified in its structural patterns, to make this inference with good precision. By quantifying this structural information, link prediction methods assign a prediction score p_{ij} to every unconnected pair such that the probability of a prediction being correct is proportional to p_{ij}. However, most of the information about the score distribution is ultimately discarded, as the goal is to identify which links have the highest score. As such, it may be useful to design algorithms that can efficiently sample links according to their score distribution without outputting the full distribution.

We note also that we are considering no extra structure on the input besides the information contained in the adjacency matrix, i.e., which links are connected, $A_{ij} = 1$, or which are not connected, $A_{ij} = 0$. Other definitions of link prediction may consider extra structure on the input by separating links between measured or unmeasured. A measured link is a link which is known to be connected or disconnected, and an unmeasured link is unknown. In that scenario, the objective of link prediction is to evaluate the set of unmeasured links to infer which should be 1 or 0. In this work we consider the first scenario, where all unconnected links are assumed to be potentially missing, as that is the information available in most datasets. Nevertheless, our work can be directly extended to the second scenario by restricting the class of links which constitute a useful prediction from the set of unconnected links to the set of unmeasured links.

Link prediction is a widely studied problem in network science, and several approaches exist including machine learning techniques[32,33], stochastic block models[34] or global perturbation methods[35]. A recent and comprehensive review on the topic can be found in Ref.[13]. In this work we focus on a popular class of link prediction methods where the predictions are made based on specific path structures between nodes[10,13,14,16,36–40], which have been shown to be competitive with other approaches in prediction precision[13,36,39,41].

Path-based link prediction
The overall mathematical structure of different path-based link prediction methods varies widely, but a common feature is the quantification of each prediction score p_{ij} through the number of paths of a specific length between i and j. For a given adjacency matrix A, each entry (i, j) of A^k represents the number of paths of length k between i and j. Initial results in path-based link prediction suggested that methods based on paths of length two are a simple way to quantify similarity between nodes[10,37],

$$P \sim A^2, \tag{1}$$

where P is the matrix of prediction scores p_{ij}. These methods are often associated with social networks, but have been applied to all types of networks with varying results. More recently it was shown that link prediction methods scoring links based on direct similarity do not perform well in protein–protein interaction networks due to the fact that proteins often connect based on neighbour similarity principles[14], i.e., proteins connect to proteins that have neighbours that are similar to themselves. The authors in[14] suggested that any link prediction method based on direct similarity can be extended to a neighbour similarity method by taking $P_{\mathrm{neighbour}} = A.P_{\mathrm{direct}}$. For the simple case of scoring links based on length two paths this implies that

$$P \sim A^3. \tag{2}$$

With this simple extension the authors proposed a link prediction method that was substantially better in the prediction of protein–protein interactions.

Other works have described direct similarity and neighbour similarity methods based on paths of even length and odd length. In[40] a linear optimization method was proposed to predict links based on a linear combination of odd powers of A. The quantum algorithm initially proposed in[20] uses the real and imaginary part of e^{-iAt} to represent even and odd based predictions. In these works, and also in an extensive review of path-based methods[36], path-based link prediction has been tested in a wide range of complex networks and several examples have been identified where either direct similarity or neighbour similarity methods tend to perform better.

Our focus in this work will be to further explore the quantum walk representation of direct and neighbour similarity for link prediction, as initially proposed in Ref.[20], providing a more efficient algorithm to implement it, and compare it to the simplest classical representations given by A^2 and A^3 in terms of query complexity. To do so, we will use a link prediction sampling framework, described next.

Sampling path-based predictions

As discussed, the link prediction problem assumes that there is some information contained in the topological structure of the graph that can be used to infer which links are missing from the network, e.g. the number of paths of different length between nodes, computed through powers of A. Let us consider a general prediction method with an associated prediction matrix P obtained as a function of the adjacency matrix, $P = f(A)$, which quantifies structural information that is useful for link prediction. Note that P does not represent the ground-truth of the missing links, it represents only the scores p_{ij} the method uses to infer which links are more likely to appear. Typically, P is computed explicitly, and the scores are used to rank the predictions from best to worst. Instead, we wish to study algorithms that take as input the adjacency matrix A of a given network and output samples of links (i, j) following the distribution of scores p_{ij} for a given $f(A)$. We write the probability of sampling a link (i, j) following $f(A)$ as

$$\mathcal{P}[(i,j)|f] = \frac{|f(A)_{ij}|^q}{\|f(A)\|_q^q}, \tag{3}$$

with $\|.\|_q$ being the $L_{p,q}$ matrix norm for $p = q$, ensuring $\sum_{ij} \mathcal{P}[(i,j)] = 1$. As we will see, the normalization is method dependent. The classical algorithms we consider sample from $f(A) = A^2$ and $f(A) = A^3$ normalized by the $L_{1,1}$ norm[28,29], and the quantum algorithm we present samples from a distribution normalized by the $L_{2,2}$ norm.

One important thing to note is that the typical forms of $f(A)$ used not only contain information about missing links, but also about existing ones. For example, when counting the number of paths of length 3 between all pairs of nodes, the entries $(A^3)_{ij}$ encode this information irrespectively of A_{ij} being 0 or 1. However, for the purpose of link prediction, the only useful predictions are those for which $A_{ij} = 0$ and $i \neq j$, i.e., predictions corresponding to new links between distinct nodes. This is an important detail of the link prediction problem, as it will condition the results obtained from any algorithm sampling from $f(A)$ directly.

For any link prediction method, we may now define the probability of sampling a *bad link*, i.e., a link that is useless for link prediction,

$$p_{B|f} \equiv \mathcal{P}[((i,j)|f) \wedge (A_{ij} = 1 \vee i = j)]. \tag{4}$$

The list of indices matching the condition $(A_{ij} = 1 \vee i = j)$ can be represented by the entries of the matrix $A + I$, which are either 0 or 1. Thus, $p_{B|f}$ can be computed by summing the probability of sampling each of these entries,

$$p_{B|f} = \sum_{i,j}^N (A+I)_{ij} \frac{|f(A)_{ij}|^q}{\|f(A)\|_q^q}. \tag{5}$$

Similarly, we can use the all-ones matrix J to represent the entries that are *useful* or *good* for link prediction through the matrix

$$G = J - (A + I). \tag{6}$$

Thus, the probability of a good sample is

$$p_{G|f} = \sum_{i,j}^N G_{ij} \frac{|f(A)_{ij}|^q}{\|f(A)\|_q^q}. \tag{7}$$

For any algorithm sampling entries based on a link-prediction method with an underlying $f(A)$, we can expect that $O(1/p_{G|f})$ samples will be required before an actual *useful* link prediction is observed. For the remainder of the text, we omit the f subscript in p_G, as it will be clear from context which method is being referenced. The number of samples needed to observe a *correct* link prediction is harder to characterize, as that is not only dependent on the structure of $f(A)$, but also on how well $f(A)$ represents the ground-truth behind the missing links in the network, which will influence the precision of the method.

Input model

We base our comparison between classical and quantum algorithms for link prediction on both having query access to a common input model, which we now describe. We consider the *general graph model* (GGM)[42,43], which allows query access to the graph $\mathcal{G} = (V, E)$ through the following operations:

(1) **Degree query** given $v \in V$, returns the degree k_v;
(2) **Neighbour query** given $v \in V$ and an integer l, returns the l-th neighbour of v if $l \leq k_v$, and $*$ otherwise;
(3) **vertex-pair query** given $u, v \in V$, returns the adjacency matrix entry A_{uv};

This general model is a combination of the bounded-degree model[44], defined by the first and second operations, and the dense graph model, defined by the third operation[45]. This type of input access model was initially used for complexity studies in graph property testing algorithms, and later adapted to quantum computation literature as it provides a framework where classical and quantum resources can be directly compared through the total number of queries to the input. In quantum computation literature these models are often referred to as the adjacency list model and adjacency matrix model, respectively[46]. A quantum extension of the general graph model can be described by defining three unitary operators $\mathcal{O}_{\text{deg}}$, $\mathcal{O}_{\text{nei}}$ and $\mathcal{O}_{\text{pair}}$[47] such that

$$\mathcal{O}_{\text{deg}}|v, 0\rangle = |v, k_v\rangle, \tag{8}$$

$$\mathcal{O}_{\text{nei}}|v, l, 0\rangle = |v, l, \Gamma_l(v)\rangle, \tag{9}$$

$$\mathcal{O}_{\text{pair}}|u, v, 0\rangle = |u, v, A_{uv}\rangle. \tag{10}$$

Our work depends on having coherent access to these oracles, such that information can be queried in a superposition, which is a standard assumption in the theoretical development of quantum algorithms. The development of QRAMs to allow such access is an active research field, and some hardware proposals have been put forward[48]. Nevertheless, their practical realization still faces significant challenges.

Finally, we consider that each of the described queries, either classical or quantum, counts as $O(1)$ in the query complexity. A classical query learns a piece of information about the input, which can be stored in a classical register and does not need to be repeated. As such, there is a trivial upper bound on the query complexity for classical algorithms: any graph problem can be solved classically with at most $O(|E|)$ queries, i.e., by accessing the whole input. This says nothing about the extra number of operations required. Besides queries to the input, we also comment on the extra number of operations required in classical algorithms and the extra number of simple gates required in the quantum algorithm. Nevertheless, we focus our comparison on the query complexities.

Classical sampling algorithms
A^2 and A^3 sampling algorithms

As discussed in Sect. 2.3 some of the most basic but popular path-based link prediction methods are based on computing A^2 and A^3 to quantify the number of paths of length 2 and 3, respectively, between pairs of nodes. Here, we study classical algorithms to sample entries from $f(A) = A^2$ and $f(A) = A^3$. Algorithms 1 and 2 accomplish exactly that, previously presented in Refs.[28] and[29], and discussed in the Supplementary Material. Our contribution here is simply their inclusion in our link prediction sampling framework and the study of their query complexity. These algorithms require an initial probability distribution to be processed and are then able to produce samples of links (i, j) with probability

$$\mathcal{P}[(i, j), A, n] = \frac{|(A^n)_{ij}|}{\|A^n\|_{1,1}} \tag{11}$$

for $n = 2$ and 3.

Data: Graph $\mathcal{G}(V, E)$ in the GGM, integer n_s

Result: n_s samples of pairs (i, j)

```
 1  foreach v ∈ V
 2  │   compute p̃_v = k_v²
 3  end
 4  normalize p_v = p̃_v / ∑_v p̃_v
 5  s = 0
 6  while s < n_s
 7  │   select v ∈ V with probability p_v
 8  │   randomly select i ∈ Γ(v)
 9  │   randomly select j ∈ Γ(v)
10  │   if i ≠ j and (i, j) ∉ E then
11  │   │   save (i, j), s = s + 1
12  │   end
13  end
14  return list of pairs (i, j)
```

Algorithm 1. A^2 Sampling, adapted from[28].

Data: Graph $\mathcal{G}(V, E)$ in the GGM, integer n_s

Result: n_s samples of pairs (i, j)

```
 1  foreach v ∈ V
 2  │   foreach u ∈ Γ(v)
 3  │   │   compute p̃_uv = k_u k_v
 4  │   end
 5  end
 6  normalize p_uv = p̃_uv / ∑_uv p̃_uv
 7  s = 0
 8  while s < n_s
 9  │   select (u, v) with probability p_uv
10  │   randomly select i ∈ Γ(u)
11  │   randomly select j ∈ Γ(v)
12  │   if i ≠ j and (i, j) ∉ E then
13  │   │   save (i, j), s = s + 1
14  │   end
15  end
16  return list of pairs (i, j)
```

Algorithm 2. A^3 Sampling, adapted from[29].

Complexity analysis

To compute the initial distributions p_v and p_{uv}, Algorithm 1 queries the degree of each node $v \in V$ in lines 1-3, having query complexity $O(N)$, and 2 queries the degree and neighbours of each node $v \in V$ in lines 1-5, having query complexity $O(|E|)$. For each query one additional operation is used to compute the entries of p_v and p_{uv}, and thus each algorithm requires an additional $O(N)$ and $O(|E|)$ operations, respectively.

Once the distributions p_v and p_{uv} are computed, Algorithms 1 and 2 can produce multiple samples of links (i, j), as described in lines 6–13 and 8–15, respectively. First, algorithm 1 samples a node v with probability p_v in line 7, and Algorithm 2 samples a link (u, v) with probability p_{uv} in line 9. This requires a processing of p_v and p_{uv} into a cumulative array, requiring $O(N)$ and $O(|E|)$ operations, respectively, and then an additional $O(\log N)$ and $O(\log |E|)$ operations to bisect the array and draw each sample, as discussed in the Supplementary Material. No additional queries to the input are required.

Finally, having sampled v or (u, v), a link (i, j) is sampled by randomly selecting nodes from the neighbourhood of v in lines 8–9 of Algorithm 1, or from the neighbourhood of (u, v) in lines 10–11 of Algorithm 2, and then checking if (i, j) is a useful sample in lines 10–12 and 12–14, respectively. We note also that this final step

takes no additional queries to the input in Algorithm 2, as the whole graph has already been queried during the processing of p_{uv}. However, for Algorithm 1, the only information queried so far is the degree of each node. Each sample requires two extra neighbourhood queries to learn i and j, and one extra vertex-pair query to learn if $(i, j) \in E$. This will lead to approximately $O(1/p_G)$ queries per useful sample, or an extra $O(n_s/p_G)$ queries in total, assuming small enough n_s. The higher the n_s the higher the chance of drawing repeated samples, which require no extra queries, and eventually converging to the maximum number of queries $O(|E|)$.

In summary, drawing n_s samples of useful links (i, j), i.e., with $A_{ij} = 0$ and $i \neq j$, following Algorithm 1 costs

$$O\left(N + \frac{n_s}{p_G}\right) \tag{12}$$

queries to the input, and takes an extra

$$O\left(N + \frac{n_s}{p_G} \log N\right) \tag{13}$$

operations. Drawing n_s samples of useful links (i, j) following Algorithm 2 costs

$$O(|E|) \tag{14}$$

queries to the input and takes an extra

$$O\left(|E| + \frac{n_s}{p_G} \log |E|\right) \tag{15}$$

operations.

The main takeaway here is that classical algorithms access the input a number of times that scales linearly with the input size N. We make this simplified statement as in complex networks the difference between $O(N)$ and $O(|E|) = O(Nk_{av})$ is often small due to the low average connectivity $k_{av} \ll N$[1]. While this cost is mostly due to the need to pre-compute p_v and p_{uv} before drawing samples, even if p_v and p_{uv} were given in the input model a similar query cost would be required to prepare the cumulative arrays for efficient bisection.

Our main objective now will be to show that a quantum algorithm can produce path-based link prediction samples using a quantum walk model with a number of input queries that is sub-linear in N.

Quantum link prediction
An improved algorithm for link prediction

Recently, in Ref.[20], a link prediction method was proposed using continuous-time quantum walks to encode predictions based on both even-length and odd-length paths. In the original work the algorithm proposed to implement this method characterizes the predictions associated with each node j separately, thus requiring N repetitions to characterize predictions over the whole network, necessarily leading to an $O(N)$ factor in the complexity.

Here we provide an improved quantum link prediction algorithm by designing it in such a way that links can be sampled globally from the network, without the need to fix an initial node. To do so, we consider a total of $2 \log_2 N + 1$ qubits: a register n with $\log_2 N$ qubits to represent each basis state $|j\rangle$ corresponding to a localized state at a node j in the network, an extra register of qubits n' with the same size as the register of node qubits n, and one ancilla register a with a single qubit. We proceed now with the description of the circuit, exemplified in Fig. 1. All qubits are initialized in the $|0\rangle$ state,
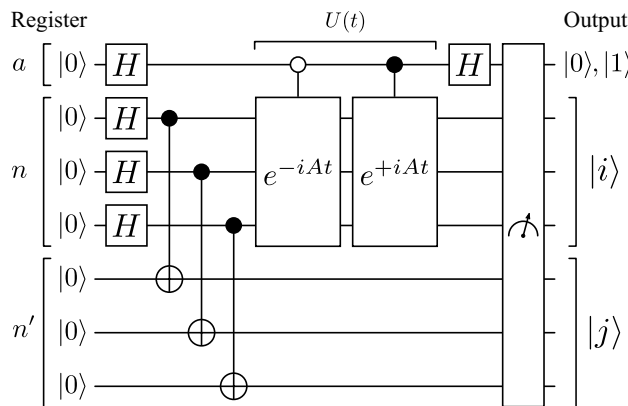
Figure 1. QLP Circuit. Example circuit to perform link prediction on a network with up to $N = 8$ nodes. The total number of gates for a network with size N is $\log_2 N + 2$ Hadamard gates and $\log_2 N$ CNOT gates, plus the cost of implementing $U(t)$, described as $(e^{-iAt})_n$ conditional on $a = |0\rangle$ and $(e^{+iAt})_n$ conditional on $a = |1\rangle$.

$$|0\rangle_a|0\rangle_n|0\rangle_{n'}, \tag{16}$$

after which Hadamard gates are applied to both the ancilla qubit and the register n leading to,

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)_a \left(\frac{1}{\sqrt{N}}\sum_{w=1}^{N}|w\rangle\right)_n |0\rangle_{n'}. \tag{17}$$

By applying a CNOT gate between each qubit in n and the respective duplicate in n', we effectively prepare the state

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)_a \frac{1}{\sqrt{N}}\sum_{w=1}^{N}|w\rangle_n|w\rangle_{n'}. \tag{18}$$

The quantum walk is now performed on register n with an ancilla-controlled operator, while the register n' remains unchanged. Consider then an operator $U(t)$ performing this quantum walk for some time t,

$$U(t) = |0\rangle\langle 0|_a \left(e^{-iAt}\right)_n I_{n'} + |1\rangle\langle 1|_a \left(e^{+iAt}\right)_n I_{n'}, \tag{19}$$

where $I_{n'}$ is the identity operator on register n'. Applying this operator to state 18 leads to

$$|\psi(t)\rangle = \frac{1}{\sqrt{2N}}\left[|0\rangle_a\sum_{w=1}^{N}\left(e^{-iAt}\right)_n|w\rangle_n|w\rangle_{n'} + |1\rangle_a\sum_{w=1}^{N}\left(e^{+iAt}\right)_n|w\rangle_n|w\rangle_{n'}\right]. \tag{20}$$

With a final Hadamard gate on register a the $|0\rangle_a$ and $|1\rangle_a$ subspaces interfere, leading to a sum of the exponential terms $\frac{1}{2}\left(e^{-iAt} + e^{iAt}\right)$ acting on register n for $q_a = |0\rangle$, and a subtraction $\frac{1}{2}\left(e^{-iAt} - e^{iAt}\right)$ acting on register n for $q_a = |1\rangle$, which we rewrite as the cosine and sine functions with the respective phase correction on the sine,

$$|\psi(t)\rangle = |0\rangle_a\left[\frac{1}{\sqrt{N}}\sum_{w=1}^{N}\cos(At)_n|w\rangle_n|w\rangle_{n'}\right] + i|1\rangle_a\left[\frac{1}{\sqrt{N}}\sum_{w=1}^{N}\sin(At)_n|w\rangle_n|w\rangle_{n'}\right]. \tag{21}$$

We note the significance of state 21, coming from the expansion of the cossine and sine in powers of A,

$$\cos(At) = \sum_{n=0}^{\infty}\frac{(-1)^n}{(2n)!}A^{2n}t^{2n}, \tag{22}$$

$$\sin(At) = \sum_{n=0}^{\infty}\frac{(-1)^n}{(2n+1)!}A^{2n+1}t^{2n+1}. \tag{23}$$

Thus, the state encodes in each subspace of the ancilla qubit a contribution of either even or odd powers of A, which represent link prediction scores for direct and neighbour similarities, respectively.

From state 21 links can be sampled by measuring all qubits in the computational basis, as we now describe. The first step is to measure the ancilla qubit, yielding $|0\rangle_a$ or $|1\rangle_a$ with probabilities

$$p^{\text{even}}(t) = \left\|\frac{1}{\sqrt{N}}\sum_{w=1}^{N}\cos(At)_n|w\rangle_n|w\rangle_{n'}\right\|^2, \tag{24}$$

$$p^{\text{odd}}(t) = \left\|\frac{1}{\sqrt{N}}\sum_{w=1}^{N}\sin(At)_n|w\rangle_n|w\rangle_{n'}\right\|^2, \tag{25}$$

respectively. By subsequently measuring the registers n and n' the state of these qubits will collapse to some $|i\rangle_n|j\rangle_{n'}$ basis state, corresponding to a sample of a link (i, j). Samples corresponding to link predictions based on even or odd paths can be post-selected depending on register a being $|0\rangle_a$ or $|1\rangle_a$, respectively. For both cases, the probability that some link (i, j) is sampled can be computed by projecting the $|0\rangle_a$ or $|1\rangle_a$ component of Eq. 21 onto $|i\rangle_n|j\rangle_{n'}$, leading to

$$p_{ij}^{\text{even}}(t) = \frac{1}{N}|\langle i|\cos(At)|j\rangle|^2, \tag{26}$$

$$p_{ij}^{\text{odd}}(t) = \frac{1}{N}|\langle i|\sin(At)|j\rangle|^2. \tag{27}$$

The values of $p_{ij}^{\text{even}}(t)$ and $p_{ij}^{\text{odd}}(t)$ represent the even and odd path-based prediction scores that are coded into the quantum walk through the power series of the cosine and sine functions, as mentioned. These are the same prediction scores obtained in the original QLP method from[20] with an extra $1/N$ normalizing factor.

From this point forward we will use these probabilities represented by the entries of the $\cos(At)$ and $\sin(At)$ matrices,

$$p_{ij}^{\text{even}}(t) = \frac{1}{N}|\cos(At)_{ij}|^2, \tag{28}$$

$$p_{ij}^{\text{odd}}(t) = \frac{1}{N}|\sin(At)_{ij}|^2, \tag{29}$$

$$p^{\text{even}}(t) = \sum_{ij}^{N} p_{ij}^{\text{even}}(t), \tag{30}$$

$$p^{\text{odd}}(t) = \sum_{ij}^{N} p_{ij}^{\text{odd}}(t), \tag{31}$$

$$p^{\text{even}}(t) + p^{\text{odd}}(t) = 1 \tag{32}$$

Data: Graph $\mathcal{G}(V, E)$, integer n_s
Result: n_s samples of pairs (i, j)
1 **while** $s < n_s$
2   apply circuit in Fig. 1
3   **if** $a = |0\rangle$, $i \neq j$ *and* $(i, j) \notin E$ **then**
4          $\vert$ save (i, j) as even prediction, $s = s + 1$
5   **end**
6   **if** $a = |1\rangle$, $i \neq j$ *and* $(i, j) \notin E$ **then**
7          $\vert$ save (i, j) as odd prediction, $s = s + 1$
8   **end**
9 **end**

Algorithm 3. QLP Sampling.

In summary, the circuit in Fig. 1 starts from an equal superposition of all nodes in the graph as in Eq. 18, evolves it according to a controlled quantum walk leading to Eq. 21, and measures a basis state $|0\rangle_a|i\rangle_n|j\rangle_{n'}$ or $|1\rangle_a|i\rangle_n|j\rangle_{n'}$ corresponding to even or odd path-based sample of a link (i, j), with probabilities given by the expressions $p_{ij}^{\text{even}}(t)$ and $p_{ij}^{\text{odd}}(t)$, respectively.

Simulations done in Ref.[20] have already shown that a classical simulation of the quantum walk process, with the probability distributions being directly used as scores for link prediction, produces results with good prediction precision when compared with other state-of-the-art classical path-based methods over a wide range of real complex networks. In the following sections we consider the improved QLP Algorithm presented here, summarized in Algorithm 3, and study how an actual quantum computing implementation would scale in terms of query access to the input to produce a fixed number of link prediction samples.

Complexity analysis

Considering now Algorithm 3 using the circuit exemplified in Fig. 1, we describe the resources required in terms of query access to the input. To produce a sample (i, j) given a network with N nodes, other than the application of $U(t)$, the circuit in Fig. 1 uses $\log_2 N + 2$ Hadamard gates and $\log_2 N$ CNOT gates. The main computational cost of the circuit is indeed the application of $U(t)$, which can be described as two applications of $e^{\pm iAt}$ conditioned on the ancilla qubit. Having an efficient implementation of e^{-iAt} in a quantum computer for some hermitian matrix A is the important problem of Quantum Simulation. It has been shown that for some classes of matrices it is possible to efficiently simulate $A \in \mathbb{R}^{N \times N}$ with $O(\text{polylog}(N))$ queries to the input[49,50]. Until recently, the networks that were known to be efficiently simulatable were typically sparse, with all nodes having at most polylog(N) connections. Real networks, however, have complex structural properties, including densely connected hubs with poly(N) connections, community structures, small-world properties, and others. Finding a single efficiently simulatable model for complex networks in general may not be possible[51]. However, motivated by complex network analysis, a recent study[31] has shown that by adding polylog(N) hub-nodes with $O(N)$ connections to an otherwise sparse network, the resulting adjacency matrix remains simulatable in $O(\text{polylog}(N))$. This toy-model for hub-sparse networks captures the important property of densely connected nodes in complex networks, and may inspire further research into the area of complex network simulation.

Let us start by describing a general query cost of implementing e^{-iAt} as requiring C queries to the input model in Sect. 2.5. Then, to produce n_s samples of useful links from the QLP algorithm, irrespectively of those being based on even or odd paths, we will need to run the circuit in Fig. 1 $O(n_s/p_G)$ times to guarantee samples are useful, leading to a total of

$$O\left(\frac{n_s}{p_G}C\right). \tag{33}$$

queries to the input. Here, the $1/p_G$ factor conditioning the sample to be useful is multiplicative in the query complexity, as quantum algorithms to simulate e^{-iAt} require the input to be queried in superposition. This information is ultimately lost when the final measurements produce each sample, meaning that for each desired sample the C queries required to simulate e^{-iAt} must be repeated.

To continue our discussion, it is useful to select a specific simulation algorithm so that we can provide a more concrete estimate for the resources of QLP in a general setting. We can consider, for example, the quantum simulation of d-sparse matrices, i.e., each row and column having at most d elements, which can always be applied to any complex network at the cost of dealing with $d \sim \text{poly}(N)$. In Ref.[50] an optimal quantum algorithm to simulate d-sparse matrices was proposed that scales as

$$C = O\big(td\|A\|_{\max} + \text{polylog}(1/\epsilon)\big) \tag{34}$$

in the number of queries, where t is the time of the evolution, $\|A\|_{\max}$ is the maximum entry of A in absolute value and ϵ is the allowed error. In the original work it is considered that $d = O(\text{polylog}(N))$, and thus it is concluded that d-sparse matrices can be efficiently simulated. In our case, analysing QLP under the d-sparse model implies $d = k_{\max}$, the maximum degree of the network, and $\|A\|_{\max} = 1$. We may then write the query complexity of QLP using the d-sparse model and disregarding the polylogarithmic factor on the error as

$$\tilde{O}\left(\frac{n_s}{p_G}k_{\max}t\right). \tag{35}$$

As mentioned, one of the main structural properties of complex networks is the existence of large hubs. For complex networks described by a scale-free model the largest node degree in the network is estimated as

$$k_{\max} = O\left(N^{\frac{1}{\gamma-1}}\right) \tag{36}$$

where γ characterizes the power-law that describes the degree distribution, $k^{-\gamma}$. Typical values of γ for real networks are in the $2 < \gamma < 4$ range[1]. Given Eq. 36, using the d-sparse model for the simulation of A representing a scale-free complex network limits QLP to be at most polynomially faster than classical algorithms in the dependence on N. Nevertheless, this is sufficient for the analysis in the remainder of this work, and the possibility of an exponential speedup remains open given any future developments on the efficient simulation of complex networks.

We have just described the resources required to produce n_s useful link samples from A^2 and A^3 using classical algorithms, and from QLP, a quantum algorithm encoding a series of even or odd powers of A, summarized in Table 1. As mentioned, the query complexity of classical methods saturates at $O(|E|)$, after which no more queries are required as the whole graph has been read to memory. To proceed with our analysis we are going to focus on the an application of QLP using the d-sparse model from Ref.[50]. Here the query complexity is multiplicative in several parameters: the number of desired useful samples n_s, the number of samples per useful sample $1/p_G$, the time of the quantum walk t, and the maximum degree of the network $k_{\max}$. We now wish to characterize the dependence on these parameters in order to comment on an achievable quantum speedup on the dependence on N, the size of the network.

$1/p_G$ is a constant overhead
To study the dependence of the query complexity of QLP on $1/p_G$ we computed this probability over time for a range of real-world complex networks[52–56] and synthetic networks. From Eqs. 7, 28 and 29, considering both even and odd-based samples, p_G can be written as a time-dependent function on the time t of the quantum walk,

Method	Queries
A^2	$O\left(N + \frac{n_s}{p_G}\right)$
A^3	$O(|E|)$
QLP (general)	$\tilde{O}\left(\frac{n_s}{p_G}C\right)$
QLP (d-sparse)	$\tilde{O}\left(\frac{n_s}{p_G}k_{\max}t\right)$

Table 1. Query complexity comparison. For QLP we consider both a general algorithm using C queries to implement e^{-iAt} as well as the algorithm from Ref.[50] for d-sparse matrices that implements e^{-iAt} with $\tilde{O}(k_{\max}t)$ queries
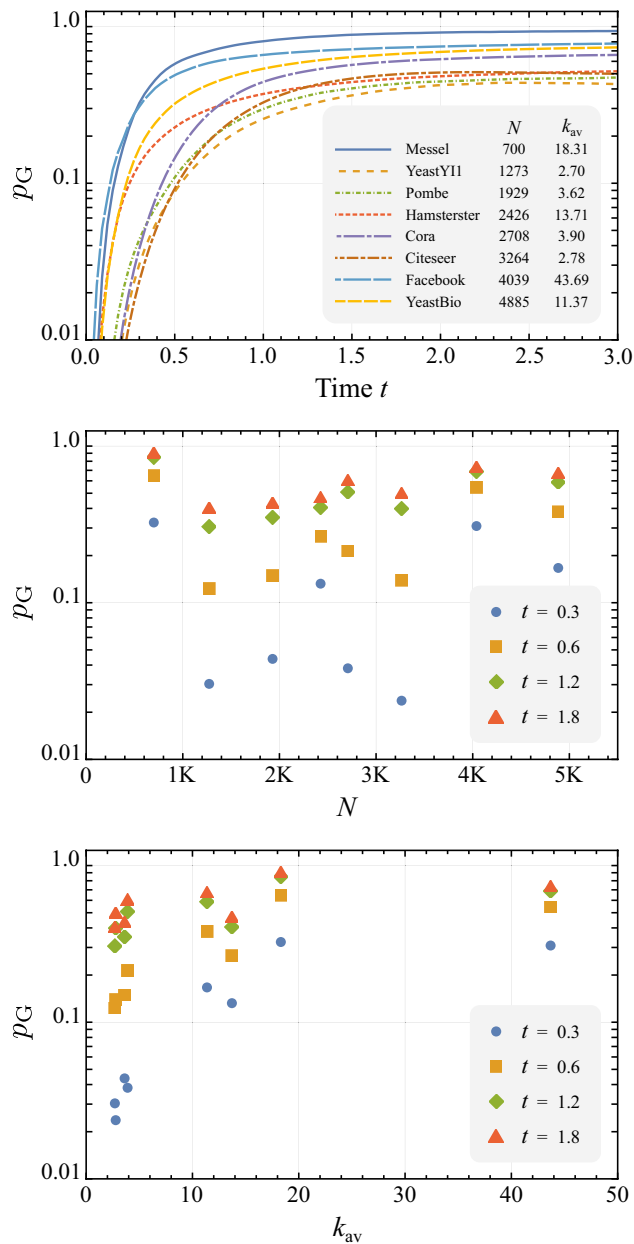
Figure 2. **Probability of sampling a useful link in QLP.** Simulating QLP on a range of datasets from real-world complex networks[52–56] we computed the total probability of obtaining a useful sample, $p_G(t)$. In the first plot, for all network sizes, the probability saturates in the 0.1 to 1.0 range for increasing t, indicating it does not decrease with N. This is more explicit in the scatter plot of p_G vs N for different values of the time t of the quantum walk. In the third plot we note there is a tendency for the useful link probability to increase with the average connectivity of the network, at least for low values of k_{av}. Overall, these results indicate the probability of sampling a useful link will typically be a constant overhead in the algorithm as N increases.

$$p_G(t) = \frac{1}{N} \sum_{i,j}^{N} G_{ij} \left(|\cos(At)_{ij}|^2 + |\sin(At)_{ij}|^2 \right), \tag{37}$$

with G the matrix of good prediction indices for a given complex network as defined in Eq. 6. In Fig. 2 we plot $p_G(t)$ for a range of real-world complex networks with different sizes and find that for all networks studied the value of $p_G(t)$ saturates in the 0.1 to 1.0 range as t increases. Through the scatter plots shown we find that $p_G(t)$ does not decrease with N and increases for small values of k_{av} after which it remains approximately constant. Similar results were observed by repeating the same analysis in three different models of synthetic networks, as shown in Figs. 1 and 2 of the Supplementary Material. Here, we found that for all three models $p_G(t)$ remains

exactly constant as N increases, being only dependent on variations of the average degree, with a similar behaviour to that observed in real-world networks.

Overall, these results indicate that the number of required samples before observing a useful sample, $O(1/p_G)$, can be considered as a constant overhead in the query complexity of QLP as N increases. How large of an overhead will depend on the network, as well the value chosen for t, as shown in Fig. 2. However, as we will see in the next section, t can typically be chosen such that $O(t/p_G)$ is a small overhead while maintaining competitive precision in the method.

t is a constant overhead

Next in our analysis is t, the time for which the quantum walk evolves over the network before each sample is obtained. This parameter influences both complexity and precision over three factors:

Factor 1 - If the value chosen is too small the quantum walk does not spread significantly over the network and thus the probability of obtaining samples that are useful for link prediction is low, increasing the overhead of $O(1/p_G)$ in the query complexity.

Factor 2 - At the same time, higher values of t imply that the quantum walk evolution must be simulated for longer, represented by the linear dependence with t in the query complexity.

Factor 3 - Finally, our goal is to do link prediction, and here t acts as a hyper-parameter in the model which determines the weight of each power of A in power series expansion of $\cos(At)$ and $\sin(At)$, which in turn influence the precision of the method.

With these three factors in mind, we wish to characterize how QLP behaves with changes in t so that we can comment on its overall effect on the resources of the method. However, given the factors described, we can no longer focus our discussion solely on computational resources, but must also discuss precision. Given that we are considering classical algorithms based on A^2 and A^3, and a quantum algorithm that approximates matrix powers through $\cos(At)$ and $\sin(At)$, we need to guarantee that any claims we make on a resource advantage also admits a competitive precision. To analyse both resources and precision, we considered the following probabilities:

- $p_G^{\text{even}}(t)$ **or** $p_G^{\text{odd}}(t)$—probability of obtaining a useful sample from either the $\cos(At)$ **or** $\sin(At)$ components of QLP.
- $p_{C|G}^{\text{even}}(t)$ **or** $p_{C|G}^{\text{odd}}(t)$—probability of obtaining a correct sample from QLP given that the sample was useful and obtained from $\cos(At)$ **or** $\sin(At)$, respectively.
- $p_{C|G}^{A2}$ **or** $p_{C|G}^{A3}$— probability of obtaining a correct sample given that the sample was useful and obtained from the respective classical algorithms for A^2 **or** A^3.

We start by commenting on $p_G^{\text{even}}(t)$ and $p_G^{\text{odd}}(t)$. These are the two contributions summing to the global probability of obtaining a useful sample from QLP $p_G(t) = p_G^{\text{even}}(t) + p_G^{\text{odd}}(t)$, as studied in the previous section,

$$p_G^{\text{even}}(t) = \frac{1}{N} \sum_{i,j}^{N} G_{ij} |\cos(At)_{ij}|^2, \tag{38}$$

$$p_G^{\text{odd}}(t) = \frac{1}{N} \sum_{i,j}^{N} G_{ij} |\sin(At)_{ij}|^2. \tag{39}$$

Consider now a matrix A' encoding the solution to the link prediction problem, i.e., $A'_{ij} = 1$ if (i, j) is a correct prediction of a missing link in A. Then, the probability that a sample is a correct prediction based on the even or odd components of QLP is given by

$$p_C^{\text{even}}(t) = \frac{1}{N} \sum_{i,j}^{N} A'_{ij} |\cos(At)_{ij}|^2, \tag{40}$$

$$p_C^{\text{odd}}(t) = \frac{1}{N} \sum_{i,j}^{N} A'_{ij} |\sin(At)_{ij}|^2. \tag{41}$$

Finally, the probabilities of obtaining a correct sample given that the sample was useful serve as a measure of precision of the method. For the quantum method, these are represented by $p_{C|G}^{\text{even}}(t)$ and $p_{C|G}^{\text{odd}}(t)$.

$$p_{C|G}^{\text{even}}(t) = \frac{p_C^{\text{even}}(t)}{p_G^{\text{even}}(t)}, \qquad p_{C|G}^{\text{odd}}(t) = \frac{p_C^{\text{odd}}(t)}{p_G^{\text{odd}}(t)}. \tag{42}$$

Similarly, $p_{C|G}^{A2}$ and $p_{C|G}^{A3}$ can be computed for each network with a respective adjacency matrix A and a link prediction solution A'.

To characterize how QLP behaves with changes in t, we selected four example datasets where we compared the evolution of the listed probabilities over time for both even and odd-power results compared to the classical case of sampling from A^2 and A^3. For each dataset we performed a 10-fold cross validation procedure, where for each of the ten iterations 10% of the links were randomly removed to build A', and the prediction methods
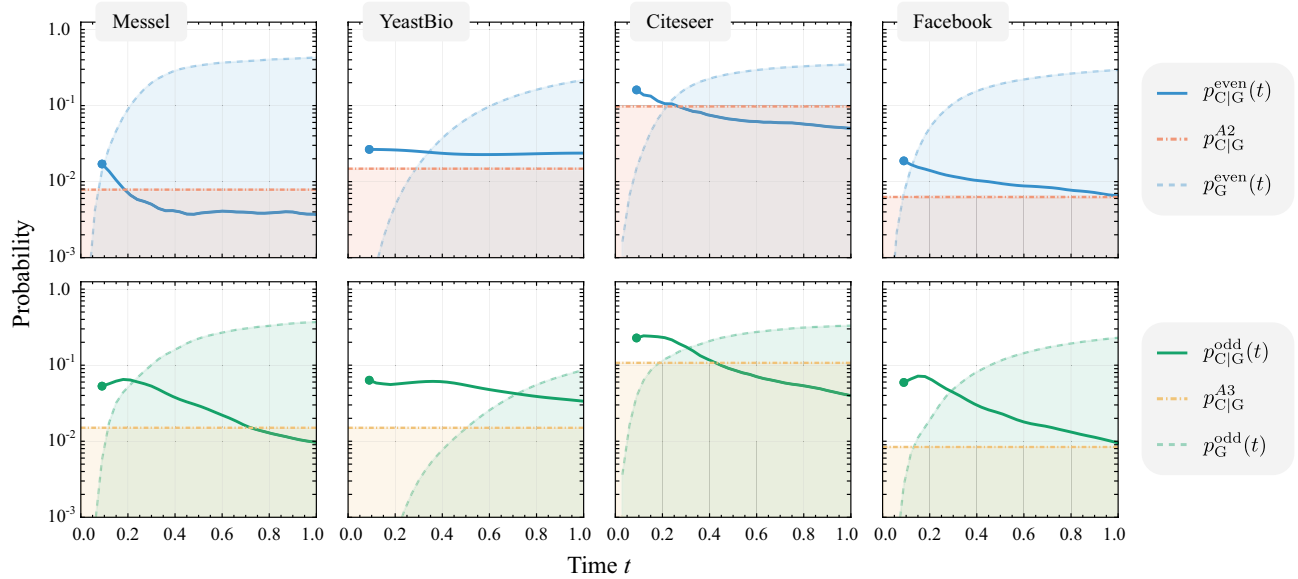
Figure 3. **Probability of sampling a correct link in QLP compared to** A^2 **and** A^3. We compare the precision of QLP with A^2 and A^3 by comparing the probability of sampling a correct prediction given that the sample was useful for four example networks. The results indicate that a value $t = O(1)$ can be chosen such that $p_{C|G}^{even}(t) \geq p_{C|G}^{A2}$ and $p_{C|G}^{odd}(t) \geq p_{C|G}^{A3}$ while maintaining a small useful sample overhead given by $1/p_G$. The plots for $p_{C|G}^{even}(t)$ and $p_{C|G}^{odd}(t)$ start at $t \approx 0.1$ to avoid the region of small p_G where the divisions in Eq. 42 are numerically unstable. Further results for more networks are provided in Fig. 3 of the Supplementary Material.

computed on the remaining 90%. In Fig. 3 we compare $p_{C|G}^{even}(t)$ with $p_{C|G}^{A2}$, superposed to $p_G^{even}(t)$, and $p_{C|G}^{odd}(t)$ with $p_{C|G}^{A3}$, superposed to $p_G^{odd}(t)$. Each result shown is an average over the ten iterations of the cross-validation procedure.

To get an intuitive reading of Fig. 3, we start by noting that in all cases it is possible to pick a $t \lesssim 1$ such that $p_G^{even}(t)$ and $p_G^{odd}(t)$ are both greater than 0.1, i.e., the sampling overhead to obtain a useful sample is small. Looking now at the precision comparison given by the $p_{C|G}$ curves, we note that for these values of t, both useful samples of the even and odd component of QLP tend to have a higher chance of being correct than those obtained from the classical algorithms for A^2 and A^3. This indicates that a value of $t = O(1)$ can typically be chosen such that QLP has competitive performance over classical sampling algorithms for A^2 and A^3 while maintaining a small sampling overhead given by $O(t/p_G)$.

Discussion and conclusions

To summarize our work, we have discussed sampling algorithms for path-based link prediction accessing the input network through the model described in Sect. 2.5, and outputting samples of links (i, j) following the distribution of scores given by a function of A. In the classical case, we have considered known algorithms to sample from A^2 and A^3 and concluded that they access the input network with a total number of queries that is linear in the number of nodes N,

$$A^2 \sim O\left(N + \frac{n_s}{p_G}\right), \quad A^3 \sim O(Nk_{av}) \tag{43}$$

In the quantum case we have presented an improved version of the QLP algorithm from Ref.[20] which can sample links globally from the network. The samples (i, j) are drawn from a score distribution following $\cos(At)$ or $\sin(At)$, which through their power-series represent even and odd powers of A, respectively, weighted by the time t of the quantum walk. Considering the d-sparse model for the quantum simulation of e^{-iAt} we estimated that QLP has a query complexity of

$$\tilde{O}\left(\frac{n_s}{p_G}k_{max}t\right), \tag{44}$$

where n_s is the total number of useful samples drawn, p_G is the probability of obtaining a useful sample, k_{max} is the maximum degree of the network, and t is the time of the quantum walk. Through numerical simulations of QLP we concluded that it is possible to draw samples with competitive precision to those obtained from A^2 and A^3 with $t/p_G \sim O(1)$. As such, considering a direct comparison to the classical sampling algorithms following the score distributions from A^2 and A^3, our final estimate for the complexity of QLP is

$$\tilde{O}(n_s k_{\max}) \tag{45}$$

in the number of queries to the input. If we consider networks following a scale-free model with a power-law degree distribution given by $k^{-\gamma}$, then

$$k_{\max} = O\left(N^{\frac{1}{\gamma-1}}\right). \tag{46}$$

Typical complex networks have a power law in the $2 < \gamma \leq 4$ range, implying that the resources described in Eq. 45 are sub-linear in N and constitute a polynomial speedup over the classical algorithms for A^2 and A^3, as long as

$$n_s < N^{\frac{\gamma-2}{\gamma-1}}. \tag{47}$$

We emphasize that the results described here are based on the quantum simulation algorithm from Ref.[50] for d-sparse matrices, which remains valid for any complex network at the cost of having $d \sim \text{poly}(N)$. Nevertheless, QLP is independent of the method used to simulate the quantum walk, and the possibility of an exponential speedup remains open given the discussion in Sect. 4.2. Depending on the structural properties found within different types of complex networks, there may be more efficient quantum simulation algorithms. For networks following the hub-sparse model described in[31], for example, the resources of QLP would scale as polylog(N), and this would be constitute an exponential speedup over a classical implementation of A^2 or A^3 sampling in these same networks.

In regards to future directions, there are few options to consider. As mentioned in the introduction, several different methods exist for link prediction, and here we focused on path-based methods which have been shown to perform reasonably well in a wide range of network types[13,36,39,41]. For path-based link prediction, although we used a quantum walk formulation, a quantum algorithm using a direct implementation of adjacency matrix powers could also be possible, for example, using the formalism of Quantum Singular Value Transformations[57]. Nevertheless, quantum algorithms for other link prediction approaches may also be developed with a potential for quantum advantage. One example would be to use quantum machine-learning techniques to adapt classical methods that learn an optimized prediction function[33]. In that case, an efficient quantum implementation of each predictor would be required.

Data availability
The datasets used in this study are available from the cited sources. The code used is available from the corresponding author on reasonable request.

References
1. Albert-László Barabási et al. *Network Science*. Cambridge University Press, 2016.
2. Barabási, Albert-László. & Albert, Réka. Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999).
3. Albert, Réka. & Barabási, Albert-László. Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**(1), 47 (2002).
4. Dorogovtsev, S. N. & Mendes, J. F. F. Advances in physics. *Evol. Netw.* **51**(4), 1079–1187 (2002).
5. Newman, Mark EJ. The structure and function of complex networks. *SIAM Rev.* **45**(2), 167–256 (2003).
6. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D.-U. Complex networks: Structure and dynamics. *Phys. Rep.* **424**(4–5), 175–308 (2006).
7. Horn, H. *et al.* Netsig: Network-based discovery from cancer genomes. *Nat. Methods* **15**(1), 61–66 (2018).
8. Cheng, F., Kovács, I. A. & Barabási, A.-L. Network-based prediction of drug combinations. *Nat. Commun.* **10**(1), 1–11 (2019).
9. Barabási, Albert-László., Gulbahce, Natali & Loscalzo, Joseph. Network medicine: A network-based approach to human disease. *Nat. Rev. Genet.* **12**(1), 56–68 (2011).
10. Liben-Nowell, David & Kleinberg, Jon. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.* **58**(7), 1019–1031 (2007).
11. Lü, Linyuan & Zhou, Tao. Link prediction in complex networks: A survey. *Phys. A Stat. Mech. Appl.* **390**(6), 1150–1170 (2011).
12. Wang, Peng, BaoWen, Xu., YuRong, Wu. & Zhou, XiaoYu. Link prediction in social networks: The state-of-the-art. *Sci. China Inf. Sci.* **58**(1), 1–38 (2015).
13. Zhou, Tao. Progresses and challenges in link prediction. *Iscience* **24**(11), 103217 (2021).
14. Kovács, I. A. *et al.* Network-based prediction of protein interactions. *Nat. Commun.* **10**(1), 1–8 (2019).
15. Yang, Yanli, Guo, Hao, Tian, Tian & Li, Haifang. Link prediction in brain networks based on a hierarchical random graph model. *Tsinghua Sci. Technol.* **20**(3), 306–315 (2015).
16. Cannistraci, C. V., Alanis-Lobato, G. & Ravasi, T. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Sci. Rep.* **3**, 1613 (2013).
17. Talasu, N., Jonnalagadda, A., Pillai, S.S.A., & Rahul, J. A link prediction based approach for recommendation systems. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp 2059–2062. IEEE, 2017.
18. Huang, Z., Li, X., & Chen, H. Link prediction approach to collaborative filtering. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, pp 141–142, (2005).
19. Daminelli, S., Thomas, J. M., Durán, C. & Cannistraci, C. V. Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks. *New J. Phys.* **17**(11), 113037 (2015).
20. Moutinho, J. P., Melo, A., Coutinho, B., Kovács, I. A. & Omar, Y. Quantum link prediction in complex networks. *Phys. Rev. A* **107**(3), 032605 (2023).
21. Maniscalco, S., Borrelli, E.-M., Cavalcanti, D., Foti, C., Glos, A., Goldsmith, M., Knecht, S., Korhonen, K., Malmi, J., & Nykänen, A., et al. Quantum network medicine: rethinking medicine with network science and quantum algorithms. arXiv preprint arXiv:2206.12405 (2022).
22. Goldsmith, M., García-Pérez, G., Malmi, J., Rossi, M.A.C., Saarinen, H., & Maniscalco, S. Link prediction with continuous-time classical and quantum walks. arXiv preprint arXiv:2208.11030, (2022).

23. Farhi, Edward & Gutmann, Sam. Quantum computation and decision trees. *Phys. Rev. A* **58**(2), 915 (1998).
24. Childs, A.M., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., & Spielman, D.A. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, pp 59–68 (2003).
25. Childs, A. M. & Goldstone, J. Spatial search by quantum walk. *Phys. Rev. A* **70**(2), 022314 (2004).
26. Chakraborty, Shantanav, Novo, Leonardo, Ambainis, Andris & Omar, Yasser. Spatial search by quantum walk is optimal for almost all graphs. *Phys. Rev. Lett.* **116**(10), 100501 (2016).
27. Apers, Simon, Chakraborty, Shantanav, Novo, Leonardo & Roland, Jérémie. Quadratic speedup for spatial search by continuous-time quantum walk. *Phys. Rev. Lett.* **129**(16), 160502 (2022).
28. Seshadhri, C., Pinar, A., & Kolda, T.G. Triadic measures on graphs: The power of wedge sampling. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp 10–18. SIAM, (2013).
29. Jha, M., Seshadhri, C., & Pinar, A. Path sampling: A fast and provable method for estimating 4-vertex subgraph counts. In *Proceedings of the 24th International Conference on World Wide Web*, pp 495–505 (2015).
30. Ballard, G., Kolda, T.G., Pinar, A., & Seshadhri, C. Diamond sampling for approximate maximum all-pairs dot-product (mad) search. In *2015 IEEE International Conference on Data Mining* pp 11–20. IEEE, (2015).
31. Magano, D., Moutinho, J.P., & Coutinho, B. On the quantum simulation of complex networks. *SciPost Physics Core.* **6**(3), 058 (2023).
32. Al Hasan, M., Chaoji, V., Salem, S., & Zaki, M. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security*, vol. 30, pp. 798–805 (2006).
33. Ghasemian, A., Hosseinmardi, H., Galstyan, A., Airoldi, E. M. & Clauset, A. Stacking models for nearly optimal link prediction in complex networks. *Proc. Natl. Acad. Sci.* **117**(38), 23393–23400 (2020).
34. Guimerà, Roger & Sales-Pardo, Marta. Missing and spurious interactions and the reconstruction of complex networks. *Proc. Natl. Acad. Sci.* **106**(52), 22073–22078 (2009).
35. Lü, L., Pan, L., Zhou, T., Zhang, Y.-C. & Stanley, H. E. Toward link predictability of complex networks. *Proc. Natl. Acad. Sci.* **112**(8), 2325–2330 (2015).
36. Zhou, Tao, Lee, Yan-Li. & Wang, Guannan. Experimental analyses on 2-hop-based and 3-hop-based link prediction algorithms. *Phys. A Stat. Mech. Appl.* **564**, 125532 (2021).
37. Zhou, Tao, Lü, Linyuan & Zhang, Yi-Cheng. Predicting missing links via local information. *Eur. Phys. J. B* **71**(4), 623–630 (2009).
38. Muscoloni, A., Abdelhamid, I., & Cannistraci, C.V. Local-community network automata modelling based on length-three-paths for prediction of complex network structures in protein interactomes, food webs and more. *bioRxiv*, pp 346916 (2018).
39. Muscoloni, A., Michieli, U., Zhang, Y., & Cannistraci, C.V. Adaptive network automata modelling of complex networks. *Preprints*, (2022).
40. Pech, Ratha, Hao, Dong, Lee, Yan-Li., Yuan, Ye. & Zhou, Tao. Link prediction via linear optimization. *Phys. A Stat. Mech. Appl.* **528**, 121319 (2019).
41. Muscoloni, A., & Cannistraci, C.V. Short note on comparing stacking modelling versus cannistraci-hebb adaptive network automata for link prediction in complex networks. *Preprints*, (2021).
42. Kaufman, Tali, Krivelevich, Michael & Ron, Dana. Tight bounds for testing bipartiteness in general graphs. *SIAM J. Comput.* **33**(6), 1441–1483 (2004).
43. Goldreich, O. *Introduction to property testing*. Cambridge University Press (2017).
44. Goldreich, O., & Ron, D. Property testing in bounded degree graphs. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pp 406–415 (1997).
45. Goldreich, Oded, Goldwasser, Shari & Ron, Dana. Property testing and its connection to learning and approximation. *J. ACM (JACM)* **45**(4), 653–750 (1998).
46. Ben-David, S., Childs, A.M., Gilyén, A., Kretschmer, W., Podder, S., & Wang, D. Symmetries, graph properties, and quantum speedups. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pp 649–660. IEEE (2020).
47. Hamoudi, Y., & Magniez, F. Quantum chebyshev's inequality and applications. arXiv preprint arXiv:1807.06456, (2018).
48. Giovannetti, Vittorio, Lloyd, Seth & Maccone, Lorenzo. Quantum random access memory. *Phys. Rev. Lett.* **100**(16), 160501 (2008).
49. Lloyd, Seth. Universal quantum simulators. *Science* **273**(5278), 1073–1078 (1996).
50. Low, G. H. & Chuang, I. L. Optimal hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.* **118**(1), 010501 (2017).
51. Childs, A.M., & Kothari, R. Limitations on the simulation of non-sparse hamiltonians. arXiv preprint arXiv:0908.4398, (2009).
52. Dunne, J. A., Labandeira, C. C. & Williams, R. J. Highly resolved early eocene food webs show development of modern trophic structure after the end-cretaceous extinction. *Proc. R. Soc. B Biol. Sci.* **281**(1782), 20133280 (2014).
53. Stark, C. *et al.* Biogrid: A general repository for interaction datasets. *Nucleic Acids Res.* **34**(1), D535–D539 (2006).
54. Kunegis, J. Konect: the koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*, pp 1343–1350 (2013).
55. Sen, Prithviraj *et al.* Collective classification in network data. *AI Mag.* **29**(3), 93–93 (2008).
56. McAuley, J. J. & Leskovec, J. Learning to discover social circles in ego networks. *NIPS* **2012**, 548–56 (2012).
57. Gilyén, A., Su, Y., Low, G.H., & Wiebe, N. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp 193–204 (2019).

Acknowledgements

Author contributions

All authors designed the goals of the research and the sampling based framework for link prediction. J.P.M. and D.M. developed the improved quantum algorithm. J.P.M. performed the calculations and simulations. All authors contributed to the writing of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-023-49906-4.

Correspondence and requests for materials should be addressed to J.P.M.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.