Discrete quantum-classical walks for link prediction

A. Marín[1], M. Soto-Gomez[3], G. Valentini[3], E. Casiraghi[3], C. Cano[4†] and D. Manzano[2†]

***Abstract*— Link prediction is a fundamental task in network science that involves estimating the likelihood of missing or future connections between nodes based on the structural patterns of a graph. While Graph Neural Networks (GNNs) have become a dominant approach, Random Walk (RW)-based methods remain highly relevant due to their lower computational demands in large-scale networks. Although recent hybrid quantum-classical walks have shown promise in exploring graph structures, existing continuous-time models are computationally expensive as they require solving high-dimensional systems of differential equations. In this paper, we propose a discrete version of a recent hybrid quantum-classical random walk algorithm that significantly improves computational efficiency by utilizing state vector representations instead of density matrices. Our method, inspired by the Quantum Jumps approach, alternates between coherent evolution and measurement-driven state collapse to generate trajectories for node embeddings. We integrate this algorithm into a link prediction pipeline and evaluate its performance across various synthetic topologies, including Grid, Power-Law, Erdös-Rényi, Barabási-Albert, and Community graphs. The experimental results demonstrate that the proposed discrete hybrid walks consistently outperform classical random walk approaches in Community and Grid topologies, showing that quantum-inspired dynamics can capture complex structural information more effectively than classical dynamics for specific topologies.**

I. Introduction

Link prediction is a fundamental problem in network science, defined as the task of estimating the likelihood of missing or future connections between two nodes based on structural patterns of the graph [1]. This problem applies to very diverse domains, from inferring connections in social networks [2], [3] to discovering unobserved protein-protein interactions in biological systems [4]. Canonical techniques to handle this problem include Common Neighbours [5] or the Adamic-Adar index [6], [2]. Graph representation learning (GRL) techniques are also used to tackle this problem [7], [8]. These methods learn graph node representations as low dimensional vector embeddings that encapsulate topological context, semantic proximity, and higher-order graph dynamics, making downstream inference more tractable and interpretable [9]. Due to the complexity of sampling big networks to infer these representations, there has been a high development of Random Walk (RW)-based models for GRL [10]. Methods such as DeepWalk or node2vec have become widely adopted in the community to extract information from big graphs by sampling local neighborhoods to analyze co-occurrence statistics [11], [12]. In contrast to network embedding models, Graph Neural Networks (GNN) methods for link prediction learn both an encoder (responsible for generating the node embeddings from the structural context and node features) and a decoder (responsible for computing the similarity between two node embeddings, and thus inferring the links) [9]. Despite having become the dominant approach for link prediction, GNNs demand higher computational resources in time and space than RW-based embedding methods, which is particularly disadvantageous for analyzing big graphs [10].

Beyond RWs, Quantum Walks (QWs) are a widely used tool to study graphs due to their propagation speed-up in comparison with their classical counterpart [13], [14], [15]. Because of this, QWs have been broadly used in several fields like searching algorithms [16], machine learning [17] or pageranking [18]. Very recently, a Hybrid Quantum-Classical Random Walk (HQCRW) has been applied as part of a GRL pipeline to the problem of community detection [19]. This hybrid methodology has been shown to simultaneously explore local and global graph structures and generate node embeddings that effectively outscore their classical counterparts for specific graph topologies. Unfortunately, this method is very expensive computationally, as it relies on density matrix representation, requiring to perform matrix multiplications of dimension $N^2 \times N^2$.

In this paper, we propose a discrete version of the HQCRW algorithm proposed in [19], that is much lighter from the computational perspective as it only implies matrix multiplication of dimension $N \times N$. We incorporate this new hybrid algorithm to a RW-based GRL pipeline to tackle the specific problem of link prediction and compare its performance with respect to baseline classical RW approaches on different graph topologies.

II. Methods

A. Pipeline for link prediction based on node embeddings and RWs

We formulate the link prediction on graphs as a two-stage pipeline grounded in traditional node-embedding GRL, where the original graph structure is first encoded into low-dimensional vector representations and subsequently exploited for downstream inference. This pipeline is shown in Fig. 1. In the embedding stage, nodes of a given graph G are mapped to continuous vectors in a Euclidean space in such a way that relevant structural properties—such as

[1]MBE. Dpt. of Electrical Engineering and Computer Science, University of Siegen, Germany. [2]QTCG. Dpt. Electromagnetism and Matter Physics and Institute Carlos I, University of Granada, Spain. [3]Dpt. di Informatica. Università degli Studi di Milano. Italy. [4]QTCG. Dpt. Computer Science and AI. University of Granada, Spain. †Corresponding authors: dmanzano@ugr.es, carloscano@ugr.es.

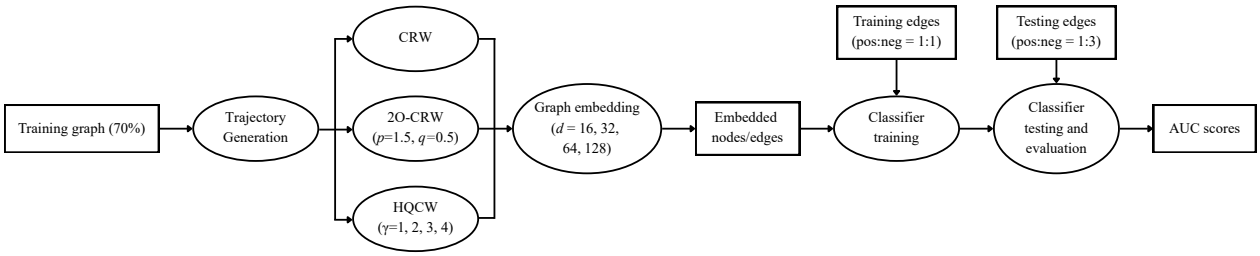

Fig. 1. Experimental evaluation pipeline.

neighborhood similarity, connectivity patterns, and higher-order proximities—are preserved. Specifically, each node in the graph G is associated with a d-dimensional vector by optimizing a skip-gram model [20] over the generated walk sequences, following the general philosophy of RW-based embedding methods such as node2vec [12]. In the inference stage, candidate links between pairs of nodes are evaluated by constructing edge-level features from their corresponding node embeddings and estimating the likelihood of edge existence using similarity measures or predictive models. Within this general framework, we propose the use of RWs and discrete-HQCRW as embedding mechanisms, and compare the performance of both for link-prediction tasks for a variety of graph topologies and sizes. Importantly, while the subsequent sections differ in the mechanism used to generate the underlying walk sequences, the remaining components of the pipeline, including the embedding learning objective and the downstream link inference strategy, remain identical across the methods considered.

B. Node embeddings with Classical Random Walks

Let $G=(V,E,w)$ denote a connected graph of N vertices M edges. $V=\{v_1,\dots,v_N\}$ is the set of vertices, $E=\{e_1,\dots,e_M\}$ is the set of edges, and $w:E\to\mathbb{R}$ assigns a weight to each edge [21]. A classical RW (CRW) on G can be modeled as a discrete-time stochastic process $\{X_t\}_{t\geq 0}$ with state space V, satisfying the Markov property. At each time step, the walker transitions from its current vertex to one of its immediate neighbors. In the simplest formulation, if the walker is located at node v_i at time t, the next position X_{t+1} is chosen uniformly among the neighbors of v_i, yielding the transition rule

$$P(X_{t+1}=u \mid X_t=v_i)=\frac{1}{\deg(v_i)},\quad u\in N(v_i),$$

where $N(v_i)$ denotes the neighborhood of v_i, and $\deg(v_i)$ its degree. Letting A be the adjacency matrix of G, with entries $A_{ij}=1$ if $(v_i,v_j)\in E$ and zero otherwise [22], the degree can be expressed as $\deg(v_i)=\sum_j A_{ij}$. This first-order random walk generates unbiased trajectories that depend solely on local connectivity and do not exploit edge weights or higher-order structural information [11].

To encode more expressive navigation patterns, the walk can be generalized to a second-order random walk (2O-CRW), where the transition probabilities depend on both the current node and the previously visited one [12]. In this setting, the probability of moving from the current node $v_i=X_t$ to a candidate neighbor $x\in N(v_i)$ is conditioned on the node $r=X_{t-1}$ visited at the previous step. For each such candidate x, an unnormalized transition weight is defined as

$$\pi_{rv_ix}=\alpha_{pq}(r,v_i,x)\,w_{v_ix},$$

where w_{v_ix} is the weight of edge (v_i,x), and α_{pq} is a bias term determined by the hop-distance[1] d_{rx} between the previous node r and the candidate node x. Specifically,

$$\alpha_{pq}(r,v_i,x)=\begin{cases}\frac{1}{p}, & d_{rx}=0,\\ 1, & d_{rx}=1,\\ \frac{1}{q}, & d_{rx}=2.\end{cases}$$

The parameters p and q control the tendency of the walk to revisit recently traversed edges or to move outward toward less explored regions of the graph. Larger values of p discourage immediate backtracking, while smaller values of q promote exploratory behavior beyond the local neighborhood [23]. The resulting transition probabilities are obtained by normalizing these scores:

$$P(X_{t+1}=x \mid X_t=v_i,X_{t-1}=r)=\frac{\pi_{rv_ix}}{\sum_{z\in N(v_i)}\pi_{rv_iz}}.$$

This second-order formulation reduces to the standard unbiased random walk when $p=q=1$, while alternative parameter choices allow the walk to smoothly interpolate between locally constrained (BFS-like) and globally exploratory (DFS-like) traversal strategies on the same graph [23].

C. Node embeddings with discrete-HQCRW

Our proposal is a discrete version of the one proposed in [19]. In this work, the hybrid model proposed is based on the Lindblad master equation [24], [25]:

$$\frac{d\rho}{dt}=-\mathrm{i}(1-\alpha)[H,\rho]+\alpha\sum_{i,j}\left(L_{i,j}\rho L_{ij}^{\dagger}-\frac{1}{2}\left\{L_{ij}^{\dagger}L_{ij},\rho\right\}\right), \tag{1}$$

where $i=\sqrt{-1}$, and $[\cdot,\cdot](\{\cdot,\cdot\})$ denote the commutator (anticommutator) of operators. The authors define the Hamiltonian H and jump operators L_{ij} using the adjacency matrix

[1]The hop-distance d_{ab} is defined as the length of the shortest path between nodes a and b.

A as $H = \sum_{ij} A_{ij} |i\rangle\langle j|$ and $L_{ij} = A_{ij} |i\rangle\langle j|$. The parameter α controls the transition from a quantum ($\alpha = 0$) to a classical ($\alpha = 1$) walk. The matrix ρ has a dimension $N \times N$ and numerically solving this equation has a complexity of $O(N^2 \times N^2)$.

These kinds of continuous-time hybrid models have been widely used to study transport properties [26] and complex networks [18]. Unfortunately, this model is computationally expensive as you have to solve a set of differential equations of N^2 variables, relying on numerical algorithms such as Runge-Kutta or Euler's [27]. In this work we propose a discrete alternative inspired by the Quantum Jumps approach [28]. This method is based on separating the dynamics generated by Eq. 1 in a coherent part, given by the first term, and an incoherent one, modeled by collapsing the state of the vector. An immediate advantage of this method is that, instead of working with density matrices (of dimension $N \times N$), we just need state vectors (dimension N).

The coherent evolution of a state vector, $|\psi\rangle$, is given by

$$|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle . \tag{2}$$

To avoid diagonalizing H we use an Euler expansion of the exponential in the form

$$e^{iH\Delta t} \simeq 1 - iH\Delta t + O(\Delta t^2), \tag{3}$$

up to first order in Δt.

Therefore, to create a trajectory, we propose to first apply the coherent evolution operator in Eq. 2 several times to update the current state of the system $\psi(t)$ and then measure the position of the walker (incoherent part of Eq.1), collapsing effectively $\psi(t)$ to one of the nodes of the graph $l \in V$. This process is repeated several times to obtain a trajectory of nodes in the graph: $l_1, l_2, ..., l_{L_{walk}}$ where L_{walk} is the length of the walk. This method is described in Algorithm 1.

Note that, for physical systems, as this approximation is of second order in Δt, it is only valid if $\Delta t \ll 1$. Given that in our case we are not modeling any real system, we can take $\Delta t = 1$. It is also important to note that any truncation of the series results in a non-unitary operator, meaning that its application does not preserve the normalization of the state vector. This is easily solved by a renormalization procedure.

D. Data generation

To perform a comparative evaluation of the performance of the different methods, we proposed a benchmark dataset composed of different synthetic graphs that vary in both size and structural topology. For this dataset, we ran the different node-embedding methods, namely embeddings based on classical RWs and based on discrete hybrid classical-quantum walks, and evaluated these embeddings for link-prediction.

The different topologies considered for the benchmark dataset are depicted in Fig. 2. The simplest topology is a two-dimensional grid (denoted `GRID`) for which the number of rows and columns was chosen as follows for a desired node size N:

Algorithm 1 Discrete HQCRW

1: Initial settings:
   $|\psi\rangle = |j\rangle$ (j: initial node for the walk)
   $steps = 0$ (counter of coherent steps)
   $jumps = 0$ (counter of jumps)
   $\gamma \in \mathbb{N}$ (Number of coherent steps before each jump)
   $\Delta t = 1$ (time step)
   $V \equiv I - iH\Delta t$ (operator coherent step)
2: **while** $steps < \gamma$ **do**
3: Apply one coherent step as:
   $|\psi\rangle = V|\psi\rangle$
   $|\psi\rangle = |\psi\rangle / ||\psi||$
4: $steps = steps + 1$
5: **end while**
6: Apply one jump as:
   Determine the probabilities of jumping from node k to node l for a time interval Δt as:
   $p_{kl} \equiv A_{lk} |\langle l|\psi\rangle|^2 \Delta t \quad (k,l) \in [1, ..., N]$.
   Normalize jump probabilities:
   $p_{kl} = p_{kl} / \sum_{k,l} p_{kl}$
   Uniformly sample a random number $r \in [0,1]$ and select a jump (k,l) such that:
   $$\sum_{(n,m)<(k,l)} p_{nm} \leq r < \sum_{(n,m)\leq(k,l)} p_{nm}$$
   where $n, m \in [1, ..., N]$.
   The state changes to $|\psi\rangle = |l\rangle$
7: Store the node l as visited
8: $jumps = jumps + 1$, $steps = 0$
9: **if** $jumps < L_{walk}$ **then**
10: Go to 2
11: **end if**

$$\text{rows} = \left\lfloor \sqrt{N} \right\rfloor , \qquad \text{columns} = \left\lceil \frac{N}{\text{rows}} \right\rceil ,$$

Power-Law graphs (`PL`) were constructed by first sampling a target degree $\deg(v_i) \equiv k_i$ for each node $v_i \in V$ from a truncated discrete power-law distribution [29],

$$P(k_i = k) = \frac{k^{-\alpha}}{\sum_{j=k_{\min}}^{k_{\max}} j^{-\alpha}}, \qquad k \in \{k_{\min}, \ldots, k_{\max}\},$$

with $\alpha = 2.1$, $k_{\min} = 5$, and $k_{\max} = 800$. Conditioned on the resulting degree sequence, edges were generated by pairing half-edges uniformly at random, by following a configuration-model wiring scheme [30].

Erdös-Rényi (`ER`), or random graphs [31], were generated by defining a probability p so that, for a graph with $|V| = N$ nodes, each unordered pair of distinct nodes (v_i, v_j), $i \neq j$, is independently connected with probability $p = \frac{\log N}{N}$. This choice of p places the graph in the connected regime with high probability as N increases, ensuring the absence of isolated components while preserving sparsity.

Barabási-Albert graphs (`BA`) [32], also known as scale-free graphs, were generated using a preferential attachment

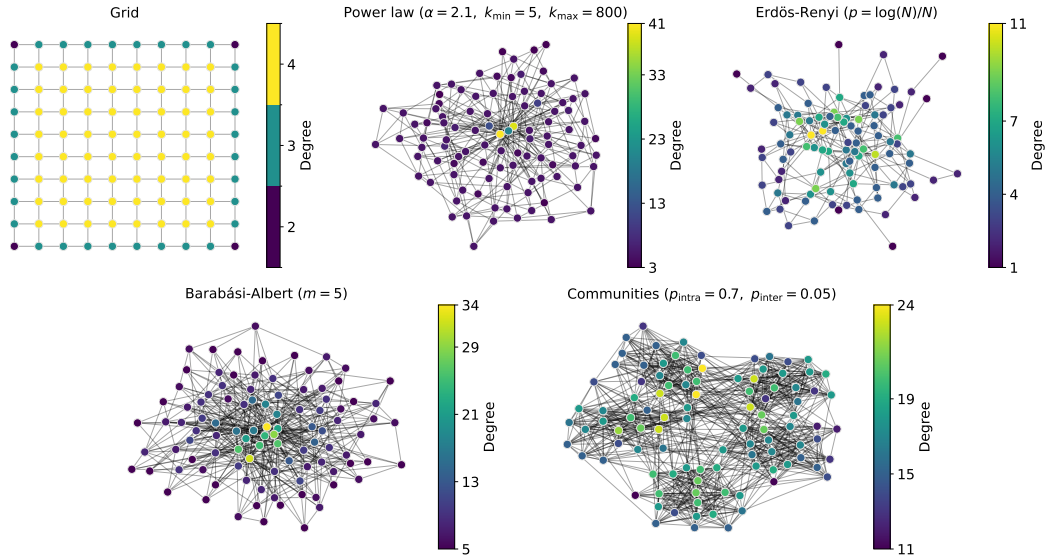

Fig. 2. Topologies considered for the benchmark dataset, along with their node-degree distributions. Graphs of size $|V| = N = 100$ nodes are shown.

mechanism that requires nodes to be systematically added so that each one has $m = 5$ connections to existing nodes. Therefore, the probability of connecting to an existing node v_i is proportional to its current degree $\deg(v_i) \equiv k_i$:

$$P(v_{\text{connected to } v_i}) = \frac{\deg(v_i)}{\sum_{v_j \in V_t} \deg(v_j)}$$

This growth process yields sparse graphs with an asymptotic power-law degree distribution $P(k) \sim k^{-3}$ and the emergence of hub nodes with high connectivity.

Finally, and in order to consider the existence of sparsely inter-connected node clusters, Community graphs (COMM) were generated as random graphs composed of five dense subgraphs of sizes $N/5$, which are Erdös-Renyi random networks where any given pair of nodes is connected with probability $p_{\text{intra}} = 0.7$. Between subgraphs, each possible edge is added with probability $p_{\text{inter}} = 0.05$. For each topology, we generated graphs of $|V| = N = 100$, 200, 500 and 1000 nodes. Both the datasets and the code used in this work are available at: https://github.com/QTCG-UGR/HQCWs-On-Graphs

III. Experiments and Results

The workflow of the evaluation pipeline is depicted in Figure 1. The evaluation follows a holdout-based protocol on the original positive graph, which is split into training and testing using a $70:30$ ratio. This split preserved the graph-topological structure to ensure that no nodes became isolated from the rest of the training dataset. Five different holdouts are generated for each graph for a specific size and topology.

Negative edges were generated by randomly sampling non-existing edges from the graph. During training, the dataset is balanced with a $1:1$ ratio between positive and negative edges, while for testing, we adopted an unbalanced setting with three times more negative edges than positive ones, since in actual applications usually negative edges outnumber positive ones.

For each holdout, only the edges belonging to the training dataset were used to generate random walk trajectories. These trajectories were produced by making use of the three different dynamics presented in this work: a classical random walk (CRW), a second-order classical random walk (2O-CRW), and a discrete hybrid quantum-classical walk (HQCRW). For the latter, $\gamma \in \{1,2,3,4\}$ were selected to study the effects of the number of coherent steps before each jump. As for the second-order classical random walk, parameters were set to $p = 1.5$ and $q = 0.5$.

For each methodology, we ran three trajectories of length $L_{walk} = 10$ on each of the five holdouts, starting from each node of the training dataset. The trajectories were then used to learn fixed-dimensional node embeddings, with embedding dimensions of $d = 16$, 32, 64 and 128. The learned node embeddings were subsequently used as input to the edge-prediction task. We used a decision tree classifier trained on the balanced training edge set for each holdout and evaluated on the corresponding unbalanced test set. The performance of each method was evaluated using the area under the receiver operating characteristic curve (AUC $\in [0,1]$).

We compared methods in terms of AUC using the paired-samples Wilcoxon test (Wilcoxon signed-rank test) with significance level $\alpha = 0.05$. Unless otherwise specified, tests were run on paired results obtained from the same holdout split, number of nodes, topology, and embedding dimension; additional pairing factors were matched depending on the specific comparative evaluation. To summarize statistical differences across multiple methods, we report win–tie–loss

(WTL) tables computed from all pairwise comparisons.[2]

Table I shows this comparative evaluation of performance with average results for all the tested methodologies and topologies. These results show that, among the classic RW embedding methods, 2O-CRW achieves statistically better performance; however, both classic methods are outperformed by all HQCRW variants. Among the HQCRWs, the $\gamma = 4$ configuration achieves significantly higher performance than all other HQCRWs except $\gamma = 2$. HQCRW-$\gamma = 1$ is the weakest HQCRW variant, while HQCRW-$\gamma = 3$ ties with HQCRW-$\gamma = 2$ but loses against HQCRW-$\gamma = 4$.

These results motivate a more detailed analysis to characterize the topologies in which HQCRWs are advantageous with respect to CRWs. Topology-specific WTL results are shown in Table II. `PL` graphs are better captured by CRWs: as the walk becomes more quantum, performance decreases. For `ER` graphs, 2O-CRW outperforms 1O-CRW and HQCRW-$\gamma \in \{1,4\}$, while it is not significantly different from HQCRW-$\gamma \in \{2,3\}$. Moreover, 1O-CRW ties with all HQCRWs, and HQCRWs largely tie among themselves, suggesting that classical and hybrid-quantum strategies are comparable on `ER` graphs. All models achieve similar performance on `BA` graphs (all ties in the corresponding section of Table II).

The trend is reversed for `COMM` and `GRID` topologies. On `COMM` graphs, all HQCRWs significantly outperform CRWs, and HQCRW-$\gamma = 2$ performs significantly better than HQCRW-$\gamma = 1$ and HQCRW-$\gamma = 4$. On `GRID` graphs, performance improves as γ increases, indicating that the more the hybrid-quantum walk weights its quantum component, the better the results.

IV. Conclusions

In this work we have introduced a discrete version of hybrid quantum-classical random walks that addresses the computational expensiveness in previous continuous-time models. By utilizing state vector representations instead of density matrices, our algorithm significantly reduces complexity from $O(N^2 \times N^2)$ to $O(N \times N)$. This advancement, inspired by the Quantum Jumps approach, allows for the efficient generation of node embeddings by alternating between coherent evolution and measurement-driven state collapse. Consequently, this method provides a viable alternative for analyzing large-scale networks, as it demands low computational resources while maintaining high performance.

Experimental results demonstrate that our hybrid quantum-classical walks generally outperform classical random walk strategies in link prediction tasks across various network topologies. While classical walks remained superior for Power-Law topologies, the hybrid approach showed significant advantages in Community and Grid structures, where increasing the quantum component (γ) often led to improved accuracy. In other topologies, such as Erdös-Rényi and Barabási-Albert, the performance of hybrid and classical strategies was found to be comparable. These findings suggest that quantum-inspired dynamics can effectively capture complex structural patterns in specific network architectures, being a powerful tool for GRL. Further work is needed to evaluate these models on real-world networks to demonstrate practical impact, better characterize when the hybrid quantum-classical dynamics help or hurt the performance and assess their scalability limits for very large graphs.

[2]For each ordered pair (A,B), we used a one-sided paired Wilcoxon signed-rank test with $H_1 : \mathrm{AUC}_A > \mathrm{AUC}_B$. We assign a win to A over B if $p < \alpha$; a loss if the opposite one-sided test supports $H_1 : \mathrm{AUC}_B > \mathrm{AUC}_A$ with $p < \alpha$; otherwise we record a tie. The WTL table reports, for each method, the total number of wins, ties, and losses across all pairwise comparisons.

Acknowledgment

This publication is part of the R+D projects PID2024-162141OB-I00 and PID2024-162155OB-I00, funded by MICIU/AEI/10.13039/501100011033/ FEDER, UE, and the Ministry of Economic Affairs and Digital Transformation of the Spanish Government through the QUANTUM ENIA project call - Quantum Spain project. EC was funded by the National Plan for NRRP Complementary Investments (PNC) in the call for the funding of research initiatives for technologies and innovative trajectories in the health—project n.PNC0000003—AdvaNced Technologies for Human-centrEd Medicine (project acronym: ANTHEM). Finally, we acknowledge the technical support provided by PROTEUS, the supercomputing center of the Institute Carlos I for Theoretical and Computational Physics in Granada, Spain.

References

[1] T. Zhou, "Progresses and challenges in link prediction," *iScience*, vol. 24, no. 11, art. no. 103217, 2021.

[2] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[3] T. Wang, X.-S. He, M.-Y. Zhou, and Z.-Q. Fu, "Link prediction in evolving networks based on popularity of nodes," *Scientific Reports*, vol. 7, art. no. 7147, 2017.

[4] I. A. Kovács, K. Luck, K. Spirohn, *et al.*, "Network-based prediction of protein interactions," *Nature Communications*, vol. 10, art. no. 1240, 2019.

[5] S. Daminelli, J. M. Thomas, C. Durán, and C. V. Cannistraci, "Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks," *New Journal of Physics*, vol. 17, art. no. 113037, 2015.

[6] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[7] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 833–852, 2019.

[8] P. Mei and Y.-H. Zhao, "Dynamic network link prediction with node representation learning from graph convolutional networks," *Scientific Reports*, vol. 14, art. no. 538, 2024.

[9] M. Xu, "Understanding Graph Embedding Methods and Their Applications", *SIAM Review*, vol. 63(4), pp. 825-853, 2021.

[10] L. Cappelletti, T. Fontana, E. Casiraghi, et al. "GRAPE for fast and scalable graph processing and random-walk-based embedding". Nat Comput Sci. vol. 3, 552–568, 2023.

[11] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 701–710.

[12] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16.New York, NY, USA: Association for Computing Machinery, 2016, p. 855–864.

[13] J. Kempe, "Quantum random walks: An introductory overview," *Contemporary Physics*, vol. 44, no. 4, p. 307, 2003.

[14] J. Kempe, "Discrete quantum walks hit exponentially faster," *Probab. Theory Relat. Fields*, vol. 133, p. 215, 2005.

method	parameters	wins	ties	losses	AUC_test	std(AUC_test)	method	2O-CRW	CRW	HQCRW$\gamma=1$	HQCRW$\gamma=2$	HQCRW$\gamma=3$	HQCRW$\gamma=4$
2O-CRW	$p=1.5, q=0.5$	1	0	4	0.5991	0.1593	2O-CRW		1	-1	-1	-1	-1
CRW		0	0	5	0.5935	0.1587	CRW	-1		-1	-1	-1	-1
HQCRW	$\gamma=1$	2	0	3	0.6050	0.1646	HQCRW$\gamma=1$	1	1		-1	-1	-1
HQCRW	$\gamma=2$	3	2	0	0.6099	0.1682	HQCRW$\gamma=2$	1	1	1		0	0
HQCRW	$\gamma=3$	3	1	1	0.6083	0.1704	HQCRW$\gamma=3$	1	1	1	0		-1
HQCRW	$\gamma=4$	4	1	0	0.6118	0.1707	HQCRW$\gamma=4$	1	1	1	0	1	

TABLE I

PAIRED WIN/TIE/LOSS SUMMARY AND PAIRWISE COMPARISON FOR AVERAGE RESULTS FOR ALL THE TOPOLOGIES TESTED.

method	parameters	wins	ties	losses	AUC_test	std(AUC_test)	method	2O-CRW	CRW	HQCRW$\gamma=1$	HQCRW$\gamma=2$	HQCRW$\gamma=3$	HQCRW$\gamma=4$
Topology: `PL`													
2O-CRW	$p=1.5, q=0.5$	4	1	0	0.5018	0.0261	2O-CRW		1	0	1	1	1
CRW		2	2	1	0.4979	0.0240	CRW	-1		0	0	1	1
HQCRW	$\gamma=1$	3	2	0	0.4989	0.0239	HQCRW$\gamma=1$	0	0		1	1	1
HQCRW	$\gamma=2$	2	1	2	0.4966	0.0182	HQCRW$\gamma=2$	-1	0	-1		1	1
HQCRW	$\gamma=3$	0	0	5	0.4870	0.0142	HQCRW$\gamma=3$	-1	-1	-1	-1		-1
HQCRW	$\gamma=4$	1	0	4	0.4937	0.0190	HQCRW$\gamma=4$	-1	-1	-1	-1	1	
Topology: `ER`													
2O-CRW	$p=1.5, q=0.5$	3	2	0	0.4852	0.0171	2O-CRW		1	1	0	0	1
CRW		0	4	1	0.4811	0.0206	CRW	-1		0	0	0	0
HQCRW	$\gamma=1$	0	4	1	0.4806	0.0209	HQCRW$\gamma=1$	-1	0		0	0	0
HQCRW	$\gamma=2$	0	5	0	0.4820	0.0197	HQCRW$\gamma=2$	0	0	0		0	0
HQCRW	$\gamma=3$	1	4	0	0.4828	0.0184	HQCRW$\gamma=3$	0	0	0	0		1
HQCRW	$\gamma=4$	0	3	2	0.4776	0.0218	HQCRW$\gamma=4$	-1	0	0	0	-1	
Topology: `BA`													
2O-CRW	$p=1.5, q=0.5$	0	5	0	0.4902	0.0128	2O-CRW		0	0	0	0	0
CRW		0	5	0	0.4908	0.0099	CRW	0		0	0	0	0
HQCRW	$\gamma=1$	0	5	0	0.4913	0.0090	HQCRW$\gamma=1$	0	0		0	0	0
HQCRW	$\gamma=2$	0	5	0	0.4903	0.0109	HQCRW$\gamma=2$	0	0	0		0	0
HQCRW	$\gamma=3$	0	5	0	0.4896	0.0127	HQCRW$\gamma=3$	0	0	0	0		0
HQCRW	$\gamma=4$	0	5	0	0.4912	0.0097	HQCRW$\gamma=4$	0	0	0	0	0	
Topology: `COMM`													
2O-CRW	$p=1.5, q=0.5$	0	1	4	0.8713	0.0857	2O-CRW		0	-1	-1	-1	-1
CRW		0	1	4	0.8754	0.0781	CRW	0		-1	-1	-1	-1
HQCRW	$\gamma=1$	2	2	1	0.8936	0.0472	HQCRW$\gamma=1$	1	1		-1	0	0
HQCRW	$\gamma=2$	4	1	0	0.8979	0.0396	HQCRW$\gamma=2$	1	1	1		0	1
HQCRW	$\gamma=3$	2	3	0	0.8957	0.0441	HQCRW$\gamma=3$	1	1	0	0		0
HQCRW	$\gamma=4$	2	2	1	0.8951	0.0473	HQCRW$\gamma=4$	1	1	0	-1	0	
Topology: `GRID`													
2O-CRW	$p=1.5, q=0.5$	1	0	4	0.6474	0.0871	2O-CRW		1	-1	-1	-1	-1
CRW		0	0	5	0.6225	0.0768	CRW	-1		-1	-1	-1	-1
HQCRW	$\gamma=1$	2	0	3	0.6609	0.0762	HQCRW$\gamma=1$	1	1		-1	-1	-1
HQCRW	$\gamma=2$	3	0	2	0.6828	0.0835	HQCRW$\gamma=2$	1	1	1		-1	-1
HQCRW	$\gamma=3$	4	0	1	0.6865	0.0948	HQCRW$\gamma=3$	1	1	1	1		-1
HQCRW	$\gamma=4$	5	0	0	0.7014	0.0866	HQCRW$\gamma=4$	1	1	1	1	1	

TABLE II

PAIRED WIN/TIE/LOSS SUMMARY AND PAIRWISE COMPARISON FOR EACH TOPOLOGY.

[15] A. Makmal, M. Zhu, D. Manzano, M. Tiersch, and H. J. Briegel, "Quantum walks on embedded hypercubes," *Phys. Rev. A*, vol. 90, p. 022314, 2014.

[16] N. Shenvi, J. Kempe, and K. B. Whaley, "Quantum random-walk search algorithm," *Phys. Rev. A*, vol. 67, p. 052307, 2003.

[17] G. Paparo, V. Dunjko, A. Makmal, M. Martin-Delgado, and H. Briegel, "Quantum speedup for active learning agents," *Phys. Rev. X*, vol. 4, p. 031002, 2014.

[18] E. Sanchez-Burillo, J. Duch, J. Gomez-Gardenes, and D. Zueco, "Quantum navigation and ranking in complex networks," *Scientific Reports*, vol. 2, p. 605, 2012.

[19] A. Marín, M. Soto-Gomez, G. Valentini, E. Casiraghi, C. Cano, and D. Manzano, "Hybrid quantum-classical walks for graph representation learning in community detection," *2025 IEEE International Conference on Quantum Artificial Intelligence (QAI)*, pages: 55-60, 2025.

[20] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality", *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111-3119, 2013.

[21] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[22] J. R. Norris, *Markov chains.* Cambridge University Press, 1997.

[23] M. Soto-Gomez, C. Cano, J. Reese, PN Robinson, G. Valentini, and E. Casiraghi, "Biasing second-order random walk sampling for heterogeneous graph embedding," *Int. Joint Conf. on Neural Networks - IJCNN2025*, 2025.

[24] G. Lindblad, "On the generators of quantum dynamical semigroups," *Commun. Math. Phys.*, vol. 119, p. 48, 1976.

[25] H. Breuer and F. Petruccione, *The theory of open quantum systems*. Oxford University Press, 2002.

[26] P. Schijven, J. Hohlberger, A. Blumen, and O. Mülken, "Modeling the quantum to classical crossover in topologically disordered networks," *J. Phys. A: Math. Theor.*, 2012.

[27] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. Cambridge University Press, 2007.

[28] M. B. Plenio and P. L. Knight, "The quantum-jump approach to dissipative dynamics in quantum optics," *Rev. Mod. Phys.*, vol. 70, no. 1, p. 101, 1998.

[29] M. E. J. Newman, "Power laws, Pareto distributions and Zipf's law," *Contemporary Physics*, vol. 46, no. 5, pp. 323–351, 2005.

[30] B. K. Fosdick, D. B. Larremore, J. Nishimura, and J. Ugander, "Configuring random graph models with fixed degree sequences," *SIAM Review*, vol. 60, no. 2, pp. 315–355, 2018.

[31] P. Erdős and A. Rényi, "On random graphs," *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.

[32] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.