

Quantum Link Prediction in Dynamic Networks: A Reproduction and Controlled Comparative Evaluation of Published Methods

Ashish Kumar Thakur
25pcsw02@iiitdmj.ac.in

Shivansh Mishra*
shivansh@iiitdmj.ac.in

Ashish Singh Parihar
ashish.parihar@iiitdmj.ac.in

Department of Computer Science and Engineering, Indian Institute of Information Technology, Design and Manufacturing, Jabalpur, India

*Corresponding author

Abstract. Several methods published between 2022 and 2026 use quantum computation for link prediction in dynamic networks and report gains over classical baselines. Their results are difficult to compare because the papers differ in temporal splitting, negative sampling, baselines, classifiers and metrics, and none of the methods has been re-executed under a common protocol. We reimplemented five of these methods and a static quantum-walk reference in one framework, using exact simulation in PennyLane or, for one method, a validated PyTorch state-vector simulator. The five temporal networks reproduce the published snapshot edge counts exactly. Each snapshot-based method was evaluated over ten seeds against three targets: observed edges, new links only, and the relabelled targets on which two of the methods report. Each quantum variant was compared with a classical counterpart on the same data and splits, using paired Wilcoxon tests with Benjamini–Hochberg correction. For the two projected-kernel methods under the observed-edge target, the quantum variant had the higher mean AUC in 7 of 60 matched comparisons. One of these differences was significant, whereas 48 of the 60 were significant in favour of the classical counterpart. Because these comparisons change both feature compression and the feature map, a further comparator gave the same classifiers the ten principal components that the circuit receives. Against those inputs the circuit’s features were significantly worse in 24 of 30 comparisons for one method and 28 of 30 for the other, and significantly better in none, so the circuit rather than the compression accounts for most of the gap. No snapshot-based method exceeded the strongest classical configuration on any network. Against the relabelled targets, the same projected kernel won 59 of 60 comparisons, although these targets agree with the observed edges on about half of all pairs. A parameter-free rule that predicts previously observed pairs reached an AUC of 0.93 to 0.99 on four of the five networks. Restricting QGCNLP’s message passing to training edges lowered every model’s AUC on all five graphs, by 0.010 to 0.147, and unevenly enough to change which classical encoder ranked first on two of them. Under the Temporal Graph Benchmark’s official candidate negatives, the adaptive quantum temporal network reached 0.032 ± 0.010 in mean reciprocal rank, against 0.030 with its circuit frozen, 0.028 with the circuit on every event and 0.031 for a classical ablation of the same architecture, all far below a memorisation rule (0.482). The claim that reproduced most closely was the parity claim of the quantum-walk reference. Under this common protocol, the evidence does not establish a quantum accuracy advantage for dynamic link prediction, and it does not establish that no such advantage exists.

Keywords: Link prediction, Dynamic networks, Quantum machine learning, Projected quantum kernels, Variational quantum circuits, Reproducibility, PennyLane

1 Introduction

1.1 Background

Networks represent entities as nodes and their relationships as edges. They describe systems as different as e-mail exchange, physical proximity, citation and protein interaction, and many of them share the heterogeneous degrees, clustered neighbourhoods and changing structure of

complex networks [1]. The recorded edges are rarely complete: some relationships exist but were never observed, and others have not yet formed. Link prediction estimates which pairs of nodes are, or will become, connected [1, 2]. It is used in recommendation, in completing interaction records and in planning biological experiments, where a predicted interaction can be tested directly [3].

In a dynamic network the question becomes temporal. Given the interactions observed so far, which pairs will interact next? A network with N nodes has $O(N^2)$ candidate pairs, and both the candidate set and the structure used to score it change as the network evolves.

1.2 Classical Link Prediction

Classical methods fall into three broad families [4]. Similarity indices score a pair from its neighbourhood: common neighbours, the Jaccard coefficient, Adamic–Adar and resource allocation reward shared neighbours, preferential attachment rewards high-degree endpoints [5–7], and path-based indices extend the idea to longer paths [3, 8]. Representation-learning methods such as DeepWalk, node2vec and Walklets learn node vectors from random walks [9–11], and continuous-time variants use time-respecting walks [12]. Graph neural networks learn node representations by message passing [13–15], and temporal graph networks keep a memory for each node across an event stream [16, 17].

In dynamic networks, indices have to be recomputed as the topology changes, and learned models have to be retrained or updated as interactions arrive. The evaluation protocol matters as well: on temporal benchmarks, a baseline that remembers past edges can be highly competitive [18].

1.3 Motivation for Quantum Link Prediction

Quantum machine learning asks whether quantum states and circuits can serve as learning models, or as components of them [19]. Two separate arguments have been made for applying it to link prediction. The first is representational. Encoding features into a quantum state maps them into a Hilbert space whose dimension grows exponentially with the number of qubits, where a quantum kernel or a variational classifier might separate classes that classical models separate poorly [20, 21]. The second is computational. An algorithm that samples high-scoring pairs directly, instead of computing every score, could scale better than its classical counterparts [22]. The first argument concerns accuracy and has to be tested experimentally; the second concerns complexity and is made analytically.

Neither argument guarantees an advantage on a particular task. Whether a quantum kernel outperforms classical models depends on the data [23], kernel values can concentrate exponentially as qubits are added [24], and variational circuits can be difficult to train [25, 26]. For link prediction the practical question is whether inserting a quantum component into a prediction pipeline makes that pipeline more accurate.

1.4 Existing Methods and the Comparability Problem

Five methods published between 2022 and 2026 apply quantum computation to link prediction in dynamic networks: PQKLP [27], PQCLP [28], PQKELP [29], QGCNLP [30] and A2QTGN [31]. They use projected quantum kernels, variational classifiers, circuits inside graph convolutions and a gated amplitude encoder (Section 3.5). A sixth method, QLPCN [22], predicts links in static networks with a controlled quantum walk. It underpins the walk-based line of work and serves here as a reference. Each dynamic method reports an improvement over the baselines it selects.

These reports cannot be placed on a single scale. The papers differ in temporal splitting, negative sampling, baselines, downstream classifiers, feature dimensionality, circuits and metrics, so a difference in AUC between two papers mixes a difference in method with a difference in protocol. The problem is measurable: two papers from the same research group report the same baseline on the same network with AUC values up to 0.31 apart (Section 3.7.1).

1.5 Research Gap

Two properties of the published evaluations have, to our knowledge, not been examined. PQKLP and PQKELP score their classifiers against labels constructed from kernel spectra rather than against observed edges. In the snapshot protocol shared by three of the methods, most positive pairs are edges that already occur in the history. The one survey that discusses several of these methods describes them without running them [32], and, to our knowledge, none of the methods has been re-executed under a common protocol.

1.6 Research Questions

- RQ1** When the methods are reimplemented under a common protocol and scored against observed edges, do their quantum variants achieve a higher AUC than matched classical counterparts run on the same data and splits?
- RQ2** How does the best configuration of each quantum method compare with the strongest classical configuration run on the same data?
- RQ3** How far do the published gains depend on the evaluation target (observed edges, new links only or relabelled targets) and on edges that recur across snapshots?
- RQ4** How sensitive are the results to the optimiser, the number of qubits, the circuit pattern, the embedding and the training ratio, and what quantum resources do the circuits use?

For PQKLP, PQCLP and PQKELP, the classical counterparts in RQ1 receive the uncompressed features, whereas the quantum pipelines compress them before the circuit. RQ1 therefore asks about the quantum pipeline as published, compression included, and not about the circuit in isolation (Section 4.1). To separate the two, we add a compression-matched comparator that trains the same classifiers on the same ten principal components the circuit receives (Section 4.11), so that the circuit is the only difference between the arms.

1.7 Approach

We reimplemented the six methods from their specifications in one evaluation framework (Figure 1), with a shared data pipeline, negative sampler, set of metrics and averaging procedure. All circuits were written and simulated in PennyLane [33], including the QGCNLP projection, which trains jointly with graph convolutions through PennyLane’s PyTorch [34] interface, and the QLPCN controlled walk. No simulation used shots or noise. The circuits were checked against independent constructions, and the five temporal networks reproduce the published snapshot edge counts exactly.

Each snapshot-based method was run over ten seeds against three targets, and each quantum variant was paired with a classical counterpart on the same data and splits. For the kernel methods the counterpart uses the same classifier; under the relabelled target it also uses the same compressed features, so that only the circuit differs. Matched comparisons were tested with a paired Wilcoxon signed-rank test [35] and Benjamini–Hochberg correction [36].

1.8 Contributions

1. A reimplementation of five quantum link-prediction methods for dynamic networks and a static quantum-walk reference, with validated circuits, thirteen resolved implementation discrepancies and a record of the deviations that remain (Sections 4 and 5).
2. An organisation of the quantum methods by the pipeline stage at which quantum computation enters, together with an audit of the published baselines (Section 3).

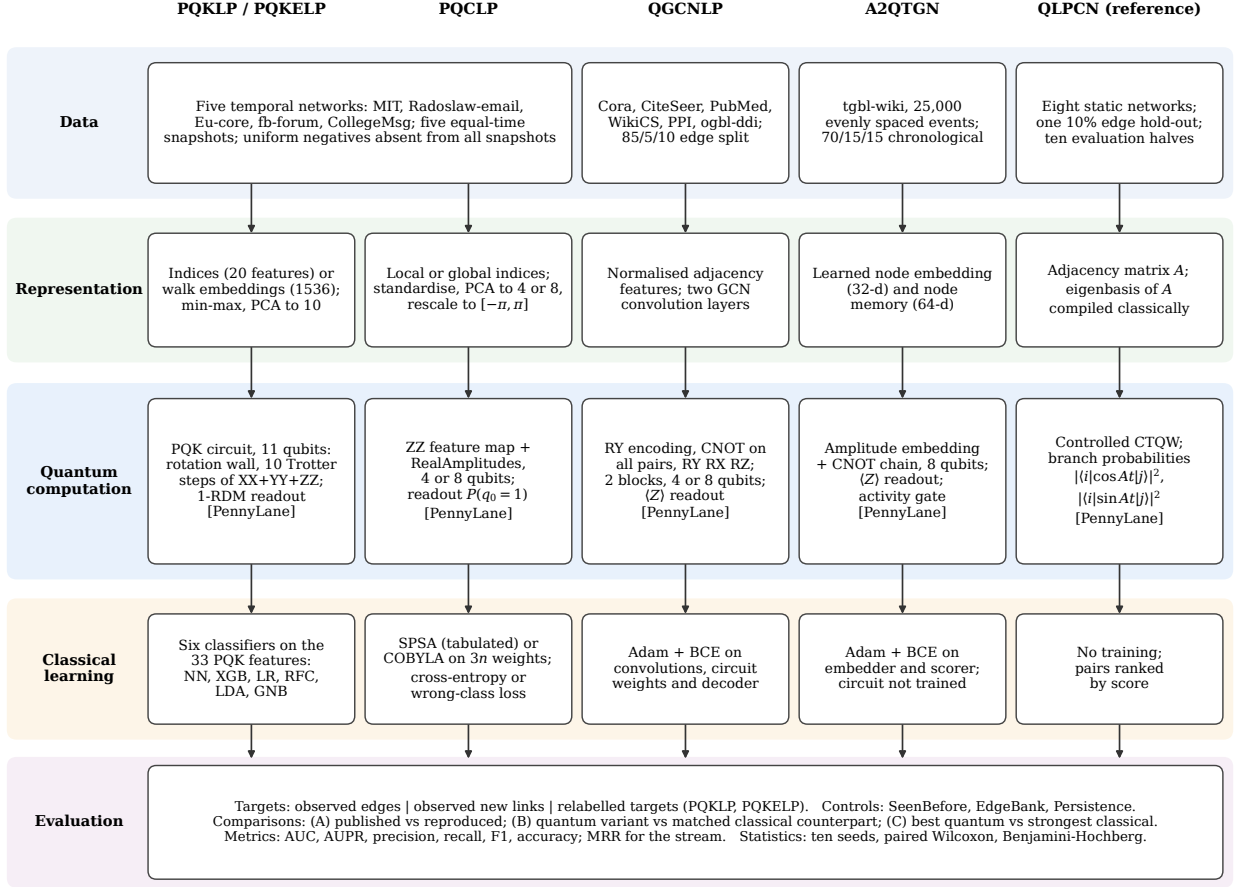


Figure 1: The reproduction framework. Rows are layers shared by all methods and columns are the reimplemented methods; each quantum block names its simulator, and the evaluation layer is common to all methods. Data loading and snapshotting are implemented in `implementation/preprocessing`, circuits in `implementation/pennylane`, classifiers in `implementation/baselines` and evaluation in `implementation/evaluation`.

3. A controlled evaluation against matched classical counterparts and against the strongest classical configuration on identical data, with paired tests over ten seeds (Sections 6.1 and 6.2).
4. Two protocol controls for snapshot evaluation, a parameter-free repeat-edge rule and a new-links target, and an analysis of the relabelled targets used by two of the methods (Section 6.6).
5. Analyses of training behaviour, sensitivity to configuration, and the resources and runtime of every executed circuit (Sections 6.5 to 6.8).

1.9 Organisation

Section 2 introduces notation and preliminaries. Section 3 reviews related work and the published evidence. Section 4 describes the framework and each method as implemented, and Section 5 the experimental setup. Section 6 reports the results, Section 7 discusses them, Section 8 sets out the threats to validity, and Section 9 concludes.

2 Background and Preliminaries

Table 1 lists the notation used in the rest of the article.

Table 1: Notation used throughout the article.

Symbol	Meaning
$G = (V, E), N$	Undirected, unweighted graph; $N = V $
A, D	Adjacency matrix; degree matrix of $A + I$
$\Gamma(u), k_u, C_u$	Neighbourhood, degree and local clustering coefficient of node u
$\mathcal{S} = \{(u_i, v_i, t_i)\}_{i=1}^M$	Timestamped interaction stream with M interactions
$G_1, \dots, G_5; E_j$	Equal-time snapshots and their edge sets; $G_1, \dots, G_4$ form the history and G_5 the target
$\mathcal{P}, \mathcal{Q}$	Positive and negative candidate pairs
$y^{\text{obs}}, y^{\text{new}}, y^{\text{rel}}$	Observed-edge, new-link and relabelled targets
$x_{uv} \in \mathbb{R}^d$	Pair feature vector
$z_u^{(j)}$	Embedding of node u in snapshot G_j
$\odot, \parallel$	Element-wise product; concatenation
$\tilde{x} \in \mathbb{R}^{d'}$	Scaled and PCA-compressed feature vector ($d' = 10$ for PQK)
n, q	Number of qubits; qubit index
$ \psi\rangle, \rho, \rho_q$	Pure state; density matrix; one-qubit reduced density matrix of qubit q
X_q, Y_q, Z_q	Pauli operators acting on qubit q
$R_P(\phi)$	Single-qubit rotation $e^{-i\phi P/2}$, $P \in \{X, Y, Z\}$
$U_{\text{enc}}(x), U(\theta)$	Encoding unitary; parameterised unitary
θ	Trainable circuit parameters (general; QGCNLP)
w	PQCLP circuit weights, $w \in \mathbb{R}^{3n}$
ϕ_i, T, r_q	PQK bond angles; number of Trotter steps ($T = 10$); rotation-wall parameters of qubit q
$\mathcal{H}$	Downstream classifier of a matched comparison
$\mathcal{L}, N_{\text{tr}}$	Training loss; number of training pairs
η, c, Δ, K	SPSA step size, perturbation size, Rademacher direction and number of iterations
$\alpha_{u,t}, \beta, \tau$	A2QTGN activity factor, sensitivity and threshold
t_w	QLPCN walk time
s, S	Seed (one seed fixes one train/test split); number of seeds ($S = 10$)
$\text{AUC}_s^Q, \text{AUC}_s^C, \Delta\text{AUC}$	Test AUC of the quantum and classical arms on seed s ; mean paired difference

2.1 Graphs and Network Representation

A graph $G = (V, E)$ has a node set V , with $N = |V|$, and an edge set $E \subseteq \{\{u, v\} : u, v \in V, u \neq v\}$. All graphs in this study are treated as undirected and unweighted: edge directions, weights and repeated interactions are discarded, and self-loops are removed (Section 4.3). The adjacency matrix $A \in \{0, 1\}^{N \times N}$ has $A_{uv} = 1$ exactly when $\{u, v\} \in E$. The neighbourhood of u is $\Gamma(u) = \{v : A_{uv} = 1\}$ and its degree is $k_u = |\Gamma(u)|$. The local clustering coefficient is $C_u = 2e_u/(k_u(k_u - 1))$, where e_u is the number of edges among the neighbours of u , and $d_G(u, v)$ denotes the shortest-path distance between u and v .

A *missing link* is a relationship that exists but is absent from the recorded edge set. A *future link* is a pair that is not an edge at the time of prediction but becomes one later. *Candidate pairs* are the pairs outside the observed edge set that a predictor scores.

A dynamic network is recorded as a time-ordered stream of interactions $\mathcal{S} = \{(u_i, v_i, t_i)\}_{i=1}^M$. In the snapshot setting the time range $[t_0, t_{\text{max}}]$ is divided into five windows of equal width $\delta = (t_{\text{max}} - t_0)/5$, and snapshot $G_j = (V, E_j)$ contains the pairs that interact at least once in window j ,

$$E_j = \{\{u, v\} : (u, v, t) \in \mathcal{S}, t_0 + (j-1)\delta \leq t < t_0 + j\delta, u \neq v\}, \quad j = 1, \dots, 5, \quad (1)$$

where the last window also includes t_{max} . All snapshots share the node set V .

2.2 The Link-Prediction Problem

In the static formulation, the observed graph $G^{\text{obs}} = (V, E^{\text{obs}})$ is a subgraph of an unknown graph $G = (V, E)$. A predictor assigns a score $\hat{y}(u, v) \in \mathbb{R}$ to each pair, and it performs well when pairs in $E \setminus E^{\text{obs}}$ receive higher scores than pairs outside E . The area under the ROC curve (AUC, Eq. 37) measures exactly this ordering.

In the snapshot formulation used by PQKLP, PQCLP, PQKELP and the temporal experiments of QGCNLP, the history $G_1, \dots, G_4$ is observed and the task is to estimate

$$\hat{p}_{uv} = P(\{u, v\} \in E_5 \mid G_1, \dots, G_4). \quad (2)$$

Only positive pairs are observed, so the task is posed as binary classification over a candidate set $\mathcal{P} \cup \mathcal{Q}$. The positives $\mathcal{P}$ are edges of G_5 , the negatives $\mathcal{Q}$ are sampled pairs that are not, and each pair is described by a feature vector $x_{uv} = \varphi(u, v; G_1, \dots, G_4) \in \mathbb{R}^d$. A classifier trained on labelled pairs returns $\hat{p}_{uv}$.

In the streaming formulation used by A2QTGN, the model processes events in time order. For each test event (u, v, t) it ranks the true destination v against a set of negative destinations using only events that precede t , and ranking quality is summarised by the mean reciprocal rank (Section 5.4).

2.3 Classical Link-Prediction Indices

The indices below are stated as the reimplementations compute them. For a pair (u, v) in a graph G ,

$$\begin{aligned} \text{CN}(u, v) &= |\Gamma(u) \cap \Gamma(v)|, & \text{JC}(u, v) &= \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}, \\ \text{AA}(u, v) &= \sum_{w \in \Gamma(u) \cap \Gamma(v), k_w > 1} \frac{1}{\ln k_w}, & \text{RA}(u, v) &= \sum_{w \in \Gamma(u) \cap \Gamma(v)} \frac{1}{k_w}, \\ \text{PA}(u, v) &= k_u k_v, & \text{CCLP}(u, v) &= \sum_{w \in \Gamma(u) \cap \Gamma(v)} C_w, \end{aligned} \quad (3)$$

with $\text{JC} = 0$ when both neighbourhoods are empty [5–7]. The shortest-path feature follows the definition used in PQKLP and PQCLP: the raw distance $\text{SP}(u, v) = d_G(u, v)$, set to -1 for disconnected pairs. The L3 index counts paths of length three with degree normalisation [3],

$$\text{L3}(u, v) = \sum_{a \in \Gamma(u) \setminus \{v\}} \sum_{b \in (\Gamma(a) \cap \Gamma(v)) \setminus \{u, a\}} \frac{1}{\sqrt{k_a k_b}}. \quad (4)$$

The local path feature is computed as $\text{LP}(u, v) = |\Gamma(u) \cap \Gamma(v)| + \varepsilon |\bigcup_{w \in \Gamma(u) \cap \Gamma(v)} \Gamma(w) \setminus \{u, v\}|$ with $\varepsilon = 0.01$. This is a set-based variant of the local path index $A^2 + \varepsilon A^3$ [8].

The published baselines keep the names used in the source papers: WEAK [37], CTDNE [12], and LGQ and a DeepWalk–GNN hybrid as used in PQKLP and PQCLP [27, 28]. The QLPCN comparators LO, CH-L2, CH-L3 and RA-L2 are defined in [22]. EdgeBank and Persistence predict an event when the pair has interacted before [18].

2.4 Quantum Computing Preliminaries

A qubit is a unit vector $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$ in $\mathbb{C}^2$, with $|a_0|^2 + |a_1|^2 = 1$. A register of n qubits has state $|\psi\rangle = \sum_{\mathbf{b} \in \{0,1\}^n} c_{\mathbf{b}}|\mathbf{b}\rangle \in \mathbb{C}^{2^n}$, a superposition of the 2^n computational basis states. The state is entangled when it cannot be written as a tensor product of single-qubit states. Its density matrix is $\rho = |\psi\rangle\langle\psi|$, and the reduced density matrix of qubit q is obtained by tracing out the other qubits, $\rho_q = \text{Tr}_{\bar{q}} \rho$. A one-qubit density matrix is determined by three expectation values,

$$\rho_q = \frac{1}{2}(I + \langle X_q \rangle X + \langle Y_q \rangle Y + \langle Z_q \rangle Z), \quad (5)$$

so the one-qubit reduced density matrices (1-RDMs) of an n -qubit state are described by $3n$ real numbers.

Gates are unitary operators. The circuits in this study use the rotations $R_P(\phi) = e^{-i\phi P/2}$ for $P \in \{X, Y, Z\}$, the Hadamard gate H , the controlled-NOT gate (CNOT) and the two-qubit Ising

rotations $\text{IsingPP}(\phi) = e^{-i\phi P \otimes P/2}$ for $P \in \{X, Y, Z\}$. A measurement in the computational basis returns $\mathbf{b}$ with probability $|\langle \mathbf{b} | \psi \rangle|^2$, and the expectation value of an observable O is $\langle O \rangle = \langle \psi | O | \psi \rangle$. On hardware these quantities are estimated from a finite number of shots. An exact state-vector simulation, as used throughout this study, returns them without sampling error.

Classical data enter a circuit through an encoding. Angle encoding sets rotation angles from feature values. Amplitude encoding writes a vector $x \in \mathbb{R}^{2^n}$ into the amplitudes of n qubits, $|\psi(x)\rangle = \sum_i (x_i / \|x\|) |i\rangle$; it is compact but discards the norm of x .

2.5 Variational Quantum Circuits

A variational circuit composes an encoding unitary with a parameterised unitary and reads out an expectation value,

$$|\psi(x, \theta)\rangle = U(\theta) U_{\text{enc}}(x) |0\rangle^{\otimes n}, \quad (6)$$

$$f_\theta(x) = \langle \psi(x, \theta) | O | \psi(x, \theta) \rangle, \quad (7)$$

where $U(\theta)$ consists of layers of parameterised single-qubit rotations and entangling gates [25, 26]. The parameters are fitted to N_{tr} labelled pairs (x_k, y_k) by minimising a loss

$$\mathcal{L}(\theta) = \frac{1}{N_{\text{tr}}} \sum_{k=1}^{N_{\text{tr}}} \ell(f_\theta(x_k), y_k) \quad (8)$$

in a hybrid loop: the circuit is evaluated on a simulator or a device, and a classical optimiser proposes new parameters [21].

Gradients can be computed analytically or by automatic differentiation in simulation, but the methods reproduced here also use gradient-free optimisers. Simultaneous perturbation stochastic approximation (SPSA) [38] estimates the gradient from two loss evaluations along a random direction $\Delta \in \{-1, +1\}^p$, where p is the number of parameters:

$$\hat{g}_k = \frac{\mathcal{L}(\theta_k + c\Delta_k) - \mathcal{L}(\theta_k - c\Delta_k)}{2c} \Delta_k, \quad \theta_{k+1} = \theta_k - \eta \hat{g}_k, \quad (9)$$

with perturbation size c and step size η . Constrained optimisation by linear approximation (COBYLA) [39] instead fits linear models of the objective to function values within a trust region. Trainability problems of variational circuits, including barren plateaus in which gradients vanish, are reviewed in [26].

2.6 Projected Quantum Kernels

A quantum kernel compares two inputs through their encoded states. The fidelity kernel $K(x, x') = \text{Tr}[\rho(x)\rho(x')]$ [20, 21] uses the full Hilbert space, and its values can concentrate exponentially in the number of qubits [24]. The projected quantum kernel of Huang et al. [23] compares reduced density matrices instead:

$$K^{\text{PQ}}(x, x') = \exp\left(-\gamma \sum_{q=1}^n \|\rho_q(x) - \rho_q(x')\|_F^2\right). \quad (10)$$

By Eq. (5), $\|\rho_q(x) - \rho_q(x')\|_F^2 = \frac{1}{2} \sum_{P \in \{X, Y, Z\}} (\langle P_q \rangle_x - \langle P_q \rangle_{x'})^2$, so Eq. (10) is a Gaussian kernel on the $3n$ -dimensional vector of one-qubit expectation values. PQKLP and PQKELP use this vector as a feature map for conventional classifiers. Huang et al. also construct datasets on which a quantum kernel outperforms a classical one, by relabelling the points according to the geometric difference between the two kernels. Section 4.10 describes the construction as PQKLP and PQKELP use it.

2.7 Continuous-Time Quantum Walks

A continuous-time quantum walk takes the adjacency matrix as a Hamiltonian and evolves a node state as $e^{-iAt_w}|j\rangle$, where t_w is the walk time. QLPCN controls the walk with an ancilla in superposition, applying e^{-iAt_w} on one branch and e^{+iAt_w} on the other, and interferes the branches with a second Hadamard gate [22]:

$$|\psi_j(t_w)\rangle = |0\rangle \otimes \cos(At_w)|j\rangle - i|1\rangle \otimes \sin(At_w)|j\rangle. \quad (11)$$

The series $\cos(At_w) = \sum_k (-1)^k t_w^{2k} A^{2k} / (2k)!$ and $\sin(At_w) = \sum_k (-1)^k t_w^{2k+1} A^{2k+1} / (2k+1)!$ weight walks of even and odd length, because $(A^\ell)_{ij}$ counts walks of length ℓ . Measuring the ancilla selects a parity, and measuring the register then samples a node i with probability proportional to $|\langle i | \cos(At_w) | j \rangle|^2$ or $|\langle i | \sin(At_w) | j \rangle|^2$. Paths of length two dominate the even branch and correspond to triadic-closure indices; paths of length three dominate the odd branch and correspond to L3. The paper estimates the cost of sampling predictions for all nodes as $\tilde{O}(N s_{av} k_{max})$, where s_{av} is the average number of samples per node and k_{max} the maximum degree, compared with approximately $O(N^{2.4})$ for the closest comparable classical method [22].

3 Related Work

3.1 Classical Link Prediction

Liben-Nowell and Kleinberg posed link prediction on evolving co-authorship networks as the ranking of node pairs by proximity and compared neighbourhood- and path-based measures [2]. Lü and Zhou later organised the resulting indices into local, quasi-local and global families, alongside likelihood-based and probabilistic models [1]. Local indices such as common neighbours, the Jaccard coefficient, Adamic-Adar, resource allocation and preferential attachment need only the neighbourhoods of the two endpoints [5–7]. Quasi-local indices such as the local path index add longer paths at modest cost [8]. Kovács et al. showed that protein-protein interactions are better predicted by paths of length three than by triadic closure [3].

Each index encodes an assumption about how links form, and its accuracy depends on whether a network follows that assumption. Triadic closure suits social networks, and length-three paths suit protein interactomes. QLPCN targets this dependence with its even and odd walk branches (Section 3.5). In dynamic networks the indices are recomputed on each snapshot and either used directly as scores or passed to a classifier. They have no trainable parameters, which makes them cheap and reproducible baselines.

3.2 Machine-Learning and Deep-Learning Link Prediction

Supervised link prediction describes each candidate pair by a feature vector, typically indices computed on one or more snapshots, and trains a classifier such as logistic regression, a random forest, gradient boosting [40] or a neural network [1]. For dynamic networks, WEAK combines weak estimators of past interactions with deep learning [37], and CTDNE learns embeddings from time-respecting random walks [12]. These methods, together with LGQ and a DeepWalk-GNN hybrid, are the baselines of the snapshot-based quantum papers [27–29].

The evaluation of dynamic link prediction has itself been studied. Poursafaei et al. showed that random negative sampling overstates performance on temporal graphs and that a memorisation baseline, EdgeBank, is competitive with learned models; they proposed historical and inductive negative sampling instead [18]. The Temporal Graph Benchmark fixed these protocols with official negative candidate lists [41]. The repeat-edge control and the new-links target used in this study (Section 4.11) apply the same concern to the snapshot protocol of the quantum papers.

3.3 Graph Representation Learning

DeepWalk learns node embeddings by applying skip-gram training to truncated random walks [9]. Node2vec biases the walks between breadth-first and depth-first exploration and forms edge representations with operators such as the Hadamard product [10], and Walklets learns embeddings at several scales by skipping nodes along each walk [11]. Graph neural networks learn representations by message passing: graph convolutional networks (GCNs) use a normalised neighbourhood average [13], GraphSAGE samples and aggregates neighbourhoods inductively [14], and graph attention networks (GATs) weight neighbours by learned attention [15]. For event streams, JODIE learns trajectories of user and item embeddings [17], and temporal graph networks (TGNs) combine a per-node memory with message and embedding modules [16]. The Open Graph Benchmark [42] and the Temporal Graph Benchmark [41] provide standard datasets and evaluation protocols for static and temporal link prediction.

These methods need training and tuning, and on benchmark suites they generally outperform fixed indices. The quantum papers engage with this literature to different extents. The snapshot-based papers compare against Node2Vec, CTDNE, WEAK, LGQ and a DeepWalk–GNN hybrid, QGCNLP compares against GCN, GraphSAGE and GAT, and A2QTGN is built on and compared with a TGN backbone.

3.4 Quantum Machine Learning for Graphs

Quantum machine learning covers quantum algorithms for learning problems as well as learning models built from quantum circuits [19]. In the kernel view, encoding data into a quantum state is a feature map into Hilbert space [20], and Havlíček et al. trained variational classifiers and estimated quantum kernels on such feature spaces [21]. Parameterised circuits are now studied as machine-learning models in their own right [25], together with the trainability problems of variational algorithms [26].

Two results bear directly on the methods reproduced here. Huang et al. showed that classical models with access to data can match quantum models unless the geometric difference between the corresponding kernels is large. They introduced projected quantum kernels to limit the concentration of kernel values and constructed relabelled datasets on which a quantum kernel outperforms classical ones [23]. Thanasilp et al. showed that quantum kernel values, projected ones included, can concentrate exponentially in the number of qubits under expressive encodings, noise or global measurements [24]. The projected-kernel construction in PQKLP and PQKELP follows the TensorFlow Quantum tutorial based on Huang et al. [43]. PQCLP was implemented in Qiskit [44] and A2QTGN in PennyLane [33].

For graph problems, a quantum walk can encode the graph itself in its dynamics, as QLPCN does [22]. Hybrid graph models instead insert circuits into classical architectures, as QGCNLP and its quantum baselines QSGCN and GQWNN do [30]. A survey of quantum social network analysis covers link prediction together with influence maximisation and community detection; its empirical section runs classical similarity indices and describes the quantum methods without running them [32].

3.5 Quantum Link Prediction

Quantum link-prediction methods are usually grouped by the quantum primitive they use: kernels, variational circuits or walks. This grouping separates methods that behave almost identically and joins methods that do not. PQKLP and PQKELP use the same primitive but differ in every classical component, while PQCLP and QGCNLP both train variational circuits but place them at different points of the pipeline. We therefore organise the methods by their *insertion point*, the stage of the prediction pipeline at which quantum computation enters (Figure 2).

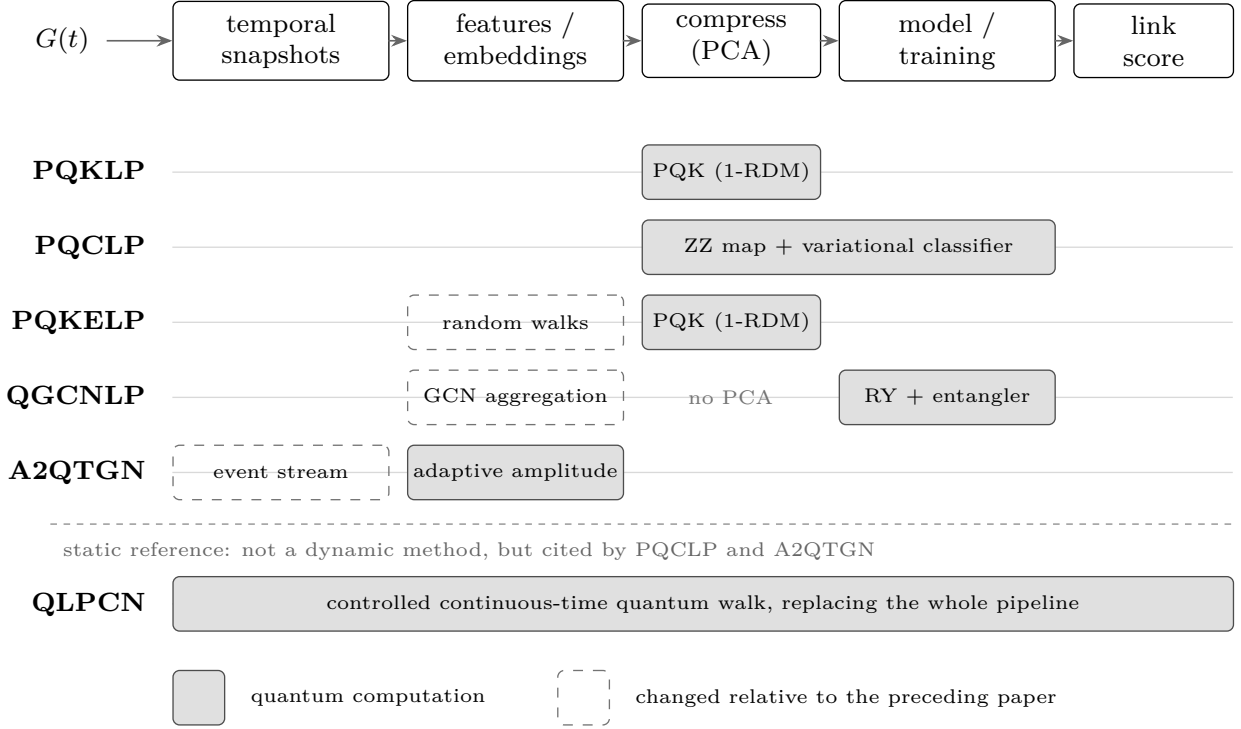


Figure 2: The dynamic link prediction pipeline and the point at which each method inserts quantum computation. PQLP and PQKELP occupy the *same* quantum position; their difference is entirely upstream, in whether the features are hand-crafted topological indices or learned random-walk embeddings. PQCLP is the only method whose circuit spans both projection and training. QGCNLP moves the circuit inside the layer and removes the compression stage that every earlier method depends on. A2QTGN is the only method whose circuit is executed conditionally, gated on temporal activity. QLPCN, shown below the rule, is a static-network algorithm that replaces the pipeline rather than augmenting it; it is included as the reference point on which the dynamic literature draws.

3.5.1 Insertion at compression: PQLP and PQKELP

PQLP [27] computes common neighbours, Jaccard, preferential attachment, Adamic–Adar and shortest-path features on four history snapshots. It compresses the twenty values to ten principal components, encodes them on eleven qubits and reads out the one-qubit reduced density matrices, which gives 33 projected-kernel features for six conventional classifiers. PQKELP [29] uses the same position but replaces the indices with Node2Vec, DeepWalk and Walklets embeddings, combined by the element-wise product of endpoint embeddings. In both methods the circuit is fixed, and the high-dimensional Hilbert space invoked in their motivation is reached only after compression to ten dimensions.

3.5.2 Insertion at the model: PQCLP and QGCNLP

PQCLP [28] groups the indices into a local family (Adamic–Adar, common neighbours, Jaccard, preferential attachment) and a global family (shortest path, local path, L3 and CCLP). It encodes them with a ZZ feature map and classifies them with a trainable RealAmplitudes ansatz on four qubits per family, or eight for the combined set, in a quantum neural network (QNN) and a variational quantum classifier (VQC) variant. It is the first method in this corpus whose circuit parameters are trained, and the paper also reports a small number of runs on quantum hardware.

QGCNLP [30] places the circuit inside a graph convolutional layer. With $\hat{A} = D^{-1/2}(A +$

$I)D^{-1/2}$, where D is the degree matrix of $A + I$, its propagation rule is

$$H_Q^{(l+1)} = f_Q\left(\sigma\left(\hat{A} H_Q^{(l)} W^{(l)}\right)\right), \quad (12)$$

where f_Q applies RY encoding, CNOT gates between every pair of qubits, parameterised RX, RY and RZ rotations and measurement. The paper argues for this placement on resource grounds. Applying the circuit only to aggregated activations costs $O(2^q)$ for a small fixed number of qubits q , compared with $O(N^2)$ qubits for per-node encodings and $O(T_w N)$ steps for walk-based models with T_w walk steps [30].

3.5.3 Insertion at encoding: A2QTGN

A2QTGN [31] keeps a TGN-style memory for each node over a continuous event stream. It encodes node features into the amplitudes of eight qubits and concatenates the resulting Pauli-Z expectation values with that memory. The circuit is executed conditionally. For node u at time t with feature change $\Delta x_{u,t}$, the activity factor

$$\alpha_{u,t} = \sigma(\beta \|\Delta x_{u,t}\|_2), \quad \beta > 0, \quad (13)$$

decides whether the node is re-encoded: the circuit runs only when $\alpha_{u,t}$ exceeds a threshold $\tau \in (0, 1)$, and the previous embedding is reused otherwise. A2QTGN is the only method in the corpus that treats circuit executions as a budgeted resource.

3.5.4 Whole-pipeline quantum walk: QLPCN

QLPCN [22] does not insert a component into a pipeline. It replaces the pipeline with the controlled quantum walk of Eq. (11), without training, feature engineering or a classifier. Its claim concerns complexity rather than accuracy: parity with classical path-based predictors at the sampling cost given in Section 2.7.

3.5.5 The design space

Read in order of publication, and leaving aside the return of PQKELP to the compression stage, the insertion point moves from compression to the model, into the layer and then to the encoding. This resembles the general movement of the field from feature engineering towards end-to-end learning. The quantum component also becomes smaller along the way, from eleven qubits applied to a compressed feature vector to four or eight qubits applied to aggregated activations. Three of the five dynamic methods, PQKLP, PQKELP and A2QTGN, never optimise a quantum parameter, so any gain they achieve comes from a fixed map rather than from quantum learning.

3.6 Relationships among the Methods

Figure 3 shows the citations among the six methods, taken from their reference lists. PQCLP is the earliest paper to cite both the kernel line (PQKLP) and the walk line (QLPCN), and it is the only method that a later member of the corpus uses as a baseline. QGCNLP’s comparison with PQCLP is the only published comparison between two members of this corpus that we found. A2QTGN cites PQKELP and QLPCN and motivates its contribution by stating that existing quantum approaches “predominantly operate on static graphs”. This description fits QLPCN and the main experiments of QGCNLP, but not the snapshot-based temporal handling of PQKLP, PQCLP and PQKELP. PQKELP does not cite PQKLP, although it uses the same projected kernel on the same five networks and differs mainly in replacing hand-crafted indices with learned embeddings.

Four of the five dynamic methods come from overlapping author teams, and only A2QTGN is external. Shared authorship does not make the results less reliable. It does mean that the successive methods, evaluated largely on the same networks against the same named baselines,

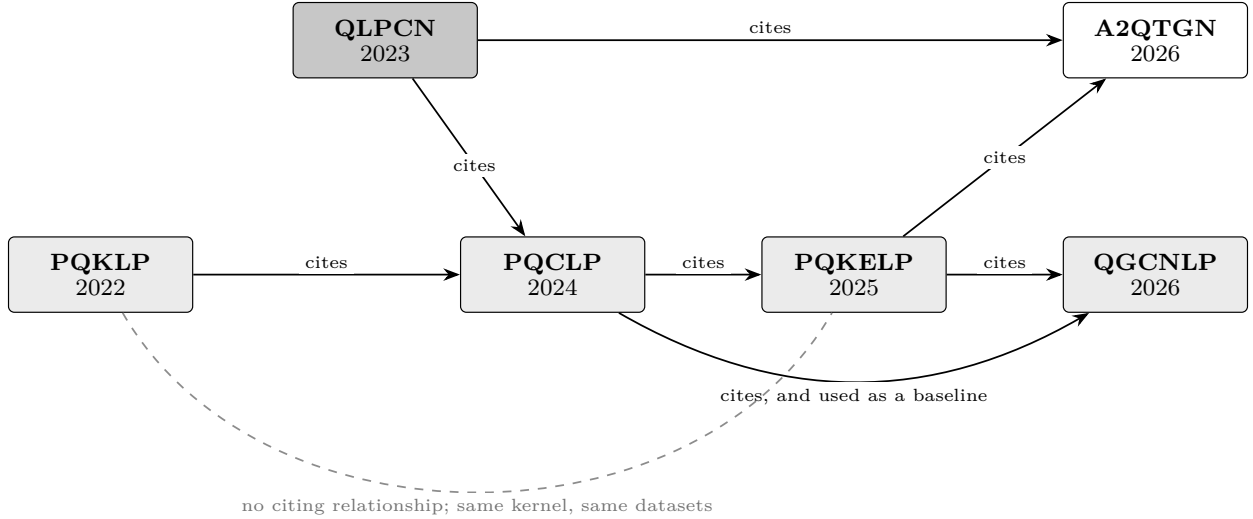


Figure 3: Citation edges among the six methods, arranged on a publication-year axis. Solid arrows point from a paper to later work that cites it. The dashed edge indicates a pair of methods with substantial methodological similarity between which no citation is present: PQKELP applies the projected quantum kernel to the same task on the same five datasets as PQKLP without citing it. Section 6.2 reports the contrast this observation motivates.

have not yet been corroborated by independent groups. The authorship of the present study is also relevant here. S. Mishra is a co-author of PQKLP. The reimplementations and evaluations reported here were carried out by the first author, who is not associated with any of the original publications. The study is therefore neither a fully external replication nor a self-assessment, and readers may wish to take this into account for the reproduction of PQKLP and for the baseline analysis of Section 3.7.1, which draws on PQKLP.

3.7 Published Evidence

Table 2 collects the best AUC that each snapshot-based method reports on the five shared temporal networks. None of the papers presents this comparison.

Table 2: Best AUC reported by each method on the five shared temporal networks, taken from each paper’s own tables and maximised over the configurations each paper reports, which favours papers reporting more configurations. The PQKELP column is the combined-embedding configuration the paper names PQKELP (its Tables 9 and 11), maximised over classifiers and train/test ratios; its single-embedding variants (its Tables 3–8) report higher values on MIT, Radoslaw-email, Eu-core and fb-forum (0.957, 0.931, 0.9431 and 0.954). QGCNLP does not report Radoslaw-email.

Network	PQKLP	PQCLP	PQKELP	QGCNLP
MIT	0.8662	0.7802	0.8732	0.9118
Radoslaw-email	0.9293	0.8993	0.9196	n/a
Eu-core	0.9454	0.8369	0.8966	0.9721
fb-forum	0.9437	0.7660	0.8799	0.9531
CollegeMsg	0.8827	0.6627	0.9548	0.8115

The published values do not improve with publication date. PQCLP, the method that moved training into the circuit, reports the lowest AUC among the methods reporting each network on all five networks, below the 2022 method that preceded it. PQKLP (2022) also reports a higher AUC than the combined-embedding configuration of PQKELP (2025) on three of the five networks. Against the single-embedding variants of PQKELP, however, PQKLP is higher only on Eu-core,

Table 3: QGCNLP evaluated against PQCLP under a single protocol, as reported in the QGCNLP paper, with Node2Vec as a classical reference. Bold marks the best value for each metric on each network. The identical precision and recall for QGCNLP on Eu-core are as printed in the source table.

Network	N2V	PQCLP		QGCNLP		
	AUC	AUC	Precision	AUC	Precision	Recall
CollegeMsg	0.6457	0.6627	0.9509	0.8115	0.9244	0.6741
Eu-core	0.9320	0.8226	0.9749	0.9721	0.9003	0.9003
fb-forum	0.9220	0.7660	0.9744	0.9531	0.8835	0.8970
MIT	0.9302	0.7802	0.8573	0.9118	0.9391	0.9794

Table 4: The same baseline methods on the same networks as reported by PQKLP and PQCLP. Δ is the absolute difference in reported AUC. The DeepWalk–GNN values are identical in both papers to four decimal places.

Baseline	Network	PQKLP	PQCLP	Δ
WEAK	fb-forum	0.9182	0.6092	0.309
WEAK	Eu-core	0.9718	0.7070	0.265
WEAK	Radoslaw-email	0.9063	0.6210	0.285
LGQ	fb-forum	0.8845	0.6168	0.268
LGQ	Eu-core	0.9165	0.7041	0.212
LGQ	MIT	0.8327	0.7167	0.116
CTDNE	MIT	0.6101	0.7375	0.127
CTDNE	Eu-core	0.7277	0.6666	0.061
DeepWalk–GNN	All five	Identical values		0.000

by 0.0023, a gap too small to rank the two. The columns were also produced under protocols that differ in negative sampling, train/test ratio and the number of averaged runs, so they are not directly comparable.

One published comparison avoids this problem because a single group produced it under a single protocol (Table 3). QGCNLP exceeds PQCLP in AUC, accuracy and recall on all four shared networks. PQCLP, however, has the highest precision on three of them, together with the lowest recall of any method compared (0.3439 on CollegeMsg). The two methods operate at different points of the precision–recall trade-off rather than in a strict order. Node2Vec, a 2016 classical method, reports a higher AUC than both quantum methods on MIT and trails QGCNLP by 0.040 on Eu-core and by 0.031 on fb-forum.

3.7.1 Baseline consistency

WEAK, CTDNE and LGQ, together with a DeepWalk–GNN hybrid, recur across the corpus. Two papers that report the same baseline on the same network would be expected to agree to within run-to-run variation. Table 4 shows that PQKLP and PQCLP differ by up to 0.31 AUC, which is enough to reverse the sign of a reported margin. PQCLP reports its best fb-forum configuration 0.157 AUC above WEAK; measured against the value that PQKLP reports for WEAK on the same network, the same configuration would lie 0.152 below it. The DeepWalk–GNN values, by contrast, are identical in both papers to four decimal places, which is consistent with those values having been carried over rather than re-measured. None of this implies an error in either paper, because each set of values may be correct under its own protocol. It does mean that comparisons across these papers rest on an assumption that the available evidence does not support.

3.8 Research Gap and Position of This Study

Table 5 summarises the related work along the dimensions that matter for this study.

Table 5: Related work relevant to this study. “Reported result” paraphrases each source, with numerical values taken from the source papers’ tables (Section 3.7). “Relevance here” states the property that matters for this study; it is not an assessment of the work as a whole.

Reference	Approach	Data	Quantum component	Reported result	Relevance here
[1, 2, 5, 7]	Similarity indices	Static networks	None	Local indices are strong and cheap predictors	Recomputed per snapshot; no learned representation
[9–11]	Random-walk embeddings	Static graphs	None	Embeddings support link prediction through edge operators	Trained per snapshot; stochastic training
[12, 37]	CTDNE; WEAK	Temporal networks	None	Temporal information improves on static embeddings	Values differ between the quantum papers (Section 3.7.1)
[13–15]	GCN, GraphSAGE, GAT	Static graphs	None	Message passing learns node representations	Classical comparators of QGCNLP
[16, 18]	TGN; EdgeBank	Event streams	None	Memorisation baselines are competitive on temporal benchmarks	Motivates the repeat-edge control
[23]	Projected quantum kernel	Engineered datasets	Trotterised evolution, 1-RDM kernel	Advantage on data relabelled from kernel spectra	Relabelling is a construction, not a link-prediction target
PQKLP [27] (2022)	PQK on topological indices	Five temporal networks	PQK, 11 qubits, 10 PCA inputs; six classifiers	Best AUC 0.8662–0.9454	Relabelled targets; baselines differ from PQCLP
PQCLP [28] (2024)	Variational classifier	Same five networks	ZZ feature map and RealAmplitudes, 4 and 8 qubits	Best AUC 0.6627–0.8993	SPSA perturbation not stated; optimiser choice unclear
QLPCN [22] (2023)	Controlled quantum walk	Static networks	CTQW, $\log_2 N + 1$ qubits	Parity with path-based predictors; $\tilde{O}(N s_{\text{av}} k_{\text{max}})$ sampling cost	Static networks only; the asymptotic claim assumes fault-tolerant hardware
PQKELP [29] (2025)	PQK on walk embeddings	Same five networks	PQK, 11 qubits, 10 PCA inputs; six classifiers	Combined embedding; best AUC 0.8732–0.9548	Relabelled targets; no comparison with PQKLP
QGCNLP [30] (2026)	Circuit inside a GCN layer	Citation, protein and OGB graphs; four temporal networks	RY encoding, all-pairs CNOT, RX/RZ; 4 and 8 qubits	Best AUC 0.8115–0.9721 on the four temporal networks	Temporal protocol differs from the other snapshot papers
A2QTGN [31] (2026)	Gated amplitude encoder in a TGN	Five TGB streams	Amplitude embedding, CNOT chain, 8 qubits	Adaptive gating improves 25k-event test accuracy by 13.36 and 22.44 points over always and never updating	Needs the official negative lists; values of τ and β not located
Survey [32] (2025)	Quantum social network analysis	Five small graphs	Described, not executed	Classical indices executed; quantum methods described	No quantum method re-executed
This study	Reproduction under a common protocol	18 networks (Table 6)	As published; PennyLane simulation	Section 6	Simulation only (Section 8)

The gaps can be stated briefly. No quantum link-prediction method for dynamic networks has been re-executed by others under a shared protocol. PQKLP and PQCLP report classical counterparts for their quantum variants, but under their own protocols, and PQKELP reports none. PQKLP and PQKELP report accuracy against relabelled targets, and in the shared snapshot protocol most positive pairs recur from the history; neither property has been examined. None of the papers reports significance tests for the differences attributed to the quantum component, or the gate counts, circuit depths and circuit executions on which a resource argument would rest.

This study addresses these gaps without proposing a new predictor. It re-executes all six methods in one framework with shared data pipelines and baselines, pairs each quantum variant with a classical counterpart, evaluates the snapshot-based methods against three targets with a parameter-free repeat-edge control, tests every matched comparison over ten seeds and counts the resources of every executed circuit.

4 Reproduction Framework and Methods as Implemented

This section describes the evaluation framework and how each method runs inside it. Where an implementation choice is ours rather than the original paper’s, the text says so, and Section 5.7 lists all such choices.

4.1 Problem Formulation

Three evaluation regimes from Section 2.2 are used. For PQKLP, PQCLP, PQKELP and the temporal experiments of QGCNLP, the snapshots of Eq. (1) define a candidate set $\mathcal{P} \cup \mathcal{Q}$ and pair features x_{uv} , and a model outputs $\hat{p}_{uv}$ as in Eq. (2). A pair receives its target in one of three ways:

$$y_{uv}^{\text{obs}} = \mathbf{1}[\{u, v\} \in E_5], \quad (14)$$

$$y_{uv}^{\text{new}} = \mathbf{1}[\{u, v\} \in E_5 \setminus \bigcup_{j=1}^4 E_j], \quad (15)$$

$$y_{uv}^{\text{rel}} = \text{the relabelled target of Section 4.10}, \quad (16)$$

where $\mathbf{1}[\cdot]$ is the indicator function. For the new-links target the positives are restricted to $E_5 \setminus \bigcup_{j \leq 4} E_j$ and the negatives are unchanged. For QGCNLP on static graphs and for QLPCN, held-out edges are scored against sampled non-edges. For A2QTGN, events are ranked in time order (Section 4.6.4).

Matched comparisons. A matched comparison runs a quantum arm and a classical arm on the same data and the same split. If AUC_s^Q and AUC_s^C are the test AUCs of the two arms on the split of seed s ,

$$\Delta\text{AUC} = \frac{1}{S} \sum_{s=1}^S (\text{AUC}_s^Q - \text{AUC}_s^C), \quad S = 10, \quad (17)$$

and the quantum variant is said to *win* when $\Delta\text{AUC} > 0$. The term is descriptive; significance is assessed separately (Section 4.12).

What else the two arms share depends on the method and the target. For PQKLP and PQKELP under the observed-edge and new-links targets, both arms use the same classifier with the same settings. The quantum arm scales the features, compresses them to $d' = 10$ principal components and applies the circuit before the classifier, whereas the classical arm applies the classifier to the scaled, uncompressed features (20 for PQKLP and 1536 for the combined PQKELP embedding). These comparisons therefore preserve the downstream classifier and the split but also differ in feature compression, so they do not isolate the quantum transformation alone. Under the relabelled target, the classical arm receives the same ten principal components that enter the circuit, and the comparison isolates the feature map more directly. For PQCLP, the quantum classifier is paired with a classical classifier of a different kind, the QNN with a neural network

and the VQC with a support vector machine; the classical classifier receives the uncompressed feature family, whereas the quantum classifier receives n principal components. For all three methods, Comparison B (Section 4.12) therefore evaluates the published quantum pipeline against a classical pipeline and not the circuit in isolation, except for the relabelled-target comparisons.

4.2 Overall Architecture

Figure 1 shows the framework as five layers shared by all methods: data, representation, quantum computation, classical learning and evaluation. Each column of the figure is one method, so the insertion points of Section 3.5 appear as the layer in which each quantum component sits. Figure 4 follows one snapshot-based experiment from the interaction stream to evaluation. The stages are the same for all snapshot methods; what changes is the feature map, the circuit, and whether the circuit parameters are updated in a hybrid loop.

4.3 Data Preparation

Loading and cleaning. Each temporal network was loaded from its original edge list (Section 5.2). Node identifiers were re-indexed to $0, \dots, N - 1$ and interactions were sorted by timestamp. Every graph built from them is undirected and has no self-loops, and repeated interactions between the same pair within a snapshot count as one edge. Static graphs were loaded through PyTorch Geometric [45] and the OGB loaders [42] and made symmetric. The QLPCN networks were loaded from local copies of the networks listed in the paper’s Table I [22], and tgbl-wiki was read from a local copy of the TGB arrays and reduced to 25,000 events taken at even intervals over the full stream of 157,474 events, so that they span its 31 days.

Snapshots. The interaction range was split into five windows of equal duration (Eq. 1). This rule reproduces the node counts and all twenty-five per-snapshot edge counts published for the five temporal networks (Section 5.2).

Positive and negative pairs. All edges of G_5 are positives. An equal number of negatives is drawn uniformly at random from node pairs that are not edges of any of $G_1, \dots, G_5$. The candidate set is drawn once per network with a fixed seed (42) and reused by every model and every seed.

Splits and seeds. For each seed $s \in \{1, \dots, 10\}$, the candidate pairs are divided into training and test sets by a stratified random split with a test fraction of 0.2; PQKELP is additionally run at training ratios from 0.5 to 0.9. The seed also fixes the model initialisation, the PQQ rotation wall and the SPSA directions, so the quantum and classical arms of a matched comparison share the split for each seed. QGCNLP on static graphs uses a random 85/5/10 split of edges into training, validation and test sets, each with sampled negatives. QLPCN holds out 10% of the edges once, with an equal number of random negatives, and each seed divides the resulting pairs into a threshold-calibration half and an evaluation half. A2QTGN uses a 70/15/15 chronological split of the event stream.

Leakage. Pair features are computed only from the history $G_1, \dots, G_4$ and never from G_5 . Every scaler, principal component analysis (PCA) and angle rescaling is fitted on the training pairs and applied unchanged to the test pairs. QLPCN scores are computed on the graph from which the held-out edges have been removed. There are two exceptions. First, the relabelled-target protocol computes kernel spectra over training and test pairs together, as the tutorial construction does (Section 4.10). Second, our QGCNLP implementation passes messages over the full edge set, including held-out edges, for QGCNLP and for its classical graph comparators alike. The models can therefore use test edges when computing node representations, which can inflate absolute link-prediction performance. Because the quantum and classical graph models share this construction,

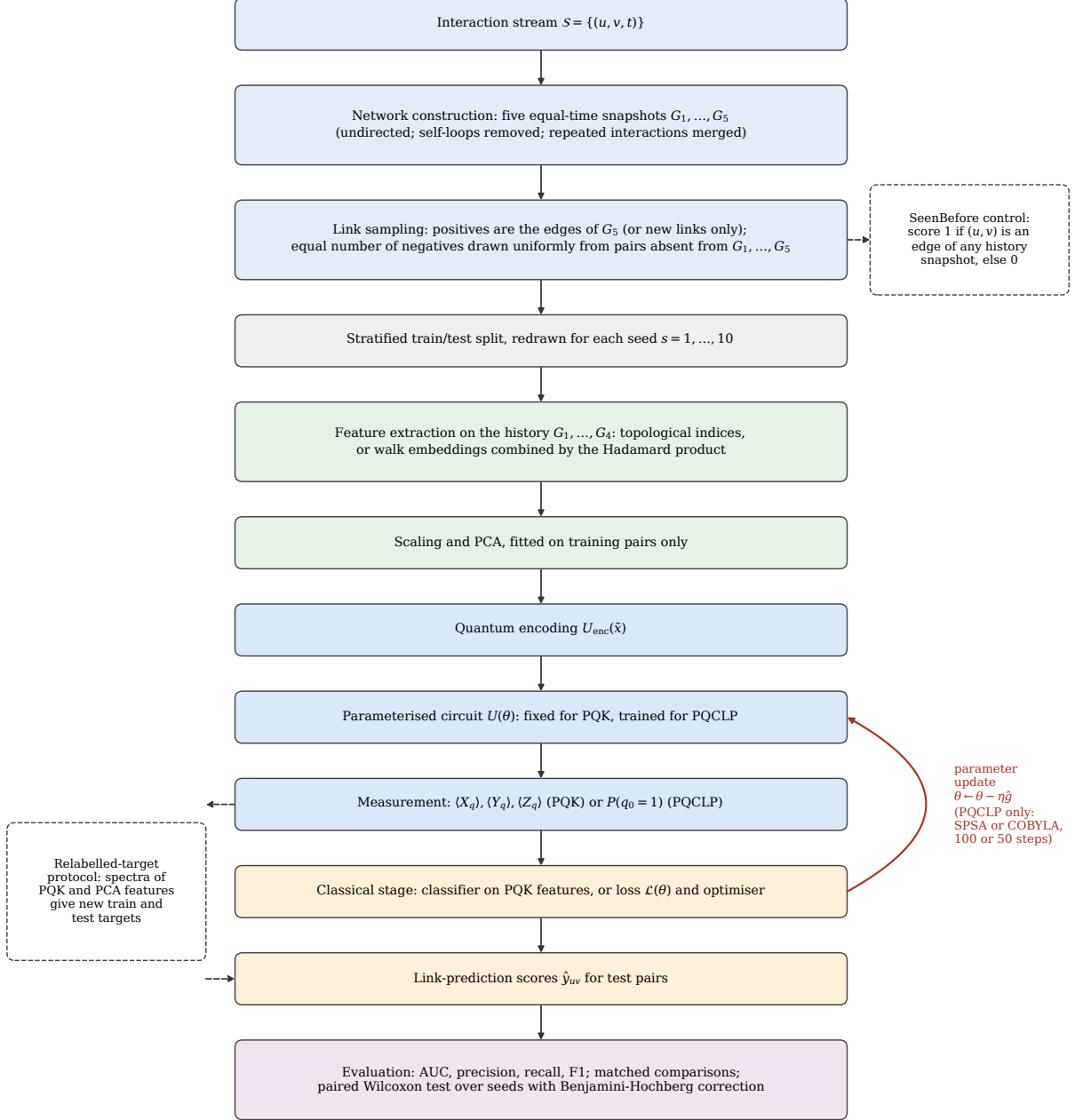


Figure 4: Methodology for the snapshot-based methods. The red arrow marks the hybrid quantum–classical loop, which is executed only for PQCLP. For PQKLP and PQKELP the circuit is fixed and its measurements feed conventional classifiers. Dashed boxes show the repeat-edge control and the relabelled-target protocol.

comparisons among them are internally controlled, but their absolute AUC values should not be compared with results from protocols that exclude test edges.

4.4 Additional controls

Three controls were added after the first round of experiments, each removing one confound from a comparison that the published protocols leave entangled.

Compression-matched comparator. The matched comparisons of PQKLP and PQKELP give the classical arm the uncompressed features and the quantum arm ten principal components, so they change compression and the feature map together. The comparator trains each classifier on the same ten components the circuit receives, after the quantum arm’s own scaling, leaving the circuit as the only difference. Writing Δ_{circuit} for the quantum arm minus this comparator, $\Delta_{\text{compression}}$ for the comparator minus the uncompressed arm, and $\Delta_{\text{published}}$ for the published matched difference, the three decompose the published gap into the part contributed by the circuit and the part contributed by compression.

Message passing restricted to training edges. Our QGCNLP runs, like the paper’s, propagate over the full edge set, so held-out edges take part in message passing for the quantum model and for its classical comparators alike. The controlled variant rebuilds the propagation graph from training edges only, leaving the split, the seeds and the models unchanged.

Official candidate negatives for A2QTGN. The reproduction ranks each test event against sampled historical negatives. The controlled variant uses the Temporal Graph Benchmark’s own candidate lists, 999 per event, with the stream’s edge features, five seeds and ten epochs. Scoring a candidate must not advance the model’s state: Algorithm 1 of the paper encodes the nodes of observed events only, so a candidate is scored from each node’s cached quantum state, and that state is written only by real events.

4.5 Feature Representation

For PQKLP, the indices CN, AA, JC, PA and SP (Section 2.3) are computed on each history snapshot and concatenated into $x_{uv} = \parallel_{j=1}^4 (\text{CN}, \text{AA}, \text{JC}, \text{PA}, \text{SP})_{G_j}(u, v) \in \mathbb{R}^{20}$. PQCLP uses the local family (AA, CN, JC, PA; $\mathbb{R}^{16}$), the global family (SP, LP, L3, CCLP; $\mathbb{R}^{16}$) or both ($\mathbb{R}^{32}$). For PQKELP, node embeddings $z_u^{(j)}$ with 128 dimensions are trained on each history snapshot with Node2Vec, DeepWalk or Walklets, or concatenated into a 384-dimensional combined embedding, and the pair feature is

$$x_{uv} = \parallel_{j=1}^4 (z_u^{(j)} \odot z_v^{(j)}), \quad (18)$$

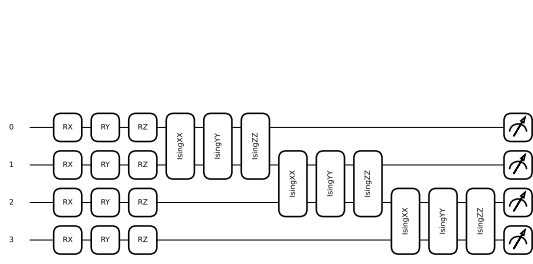
which has 512 dimensions for a single embedding and 1536 for the combined one. QGCNLP uses the rows of the normalised adjacency matrix $D^{-1/2}(A + I)D^{-1/2}$ as node features. A2QTGN uses a learned 32-dimensional node embedding e_u , because the local tgbl-wiki arrays contain no edge features. QLPCN uses the adjacency matrix directly.

4.6 Quantum Data Encoding and Circuits

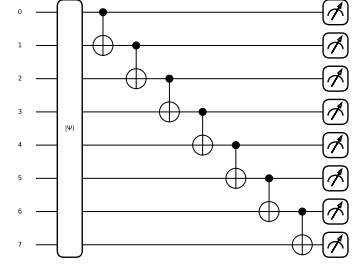
Figure 5 shows each circuit as drawn by PennyLane from the executed code.

4.6.1 PQKLP and PQKELP: the projected-kernel circuit

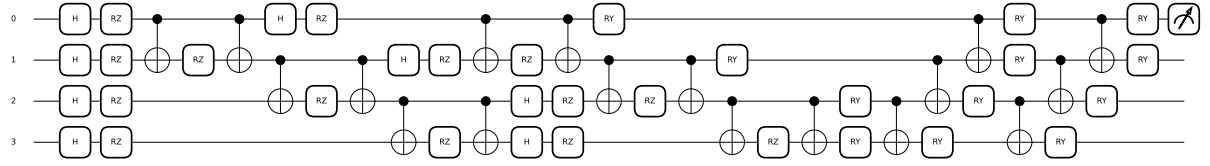
The features are min-max scaled and reduced by PCA to $\tilde{x} \in \mathbb{R}^{d'}$ with $d' = 10$, both fitted on the training pairs (Figure 6). On $n = 11$ qubits, a rotation wall with parameters $r_q \in \mathbb{R}^3$, drawn from



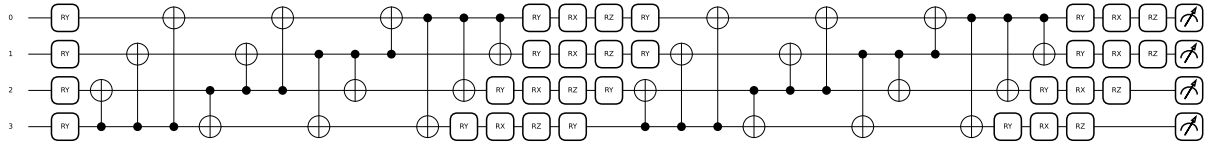
(a) PQQ circuit, 4 qubits, one Trotter step



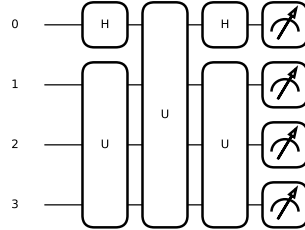
(d) A2QTGN encoder, 8 qubits



(b) PQCLP classifier, 4 qubits: ZZ feature map (2 repetitions) and RealAmplitudes (2 repetitions)



(c) QGCNLP projection, 4 qubits: two blocks of RY encoding, CNOT on all ordered pairs and trainable RY, RX and RZ



(e) QLPCN controlled walk on an 8-node path: Hadamard on the ancilla, $\Phi^\top$, ancilla-controlled phase, Φ , Hadamard

Figure 5: Circuits as implemented, drawn by `qml.draw_mpl` with random angles. (a) The PQQLP/PQKELP circuit at reduced size. The experiments use 11 qubits and 10 Trotter steps with the same pattern, a rotation wall followed by IsingXX, IsingYY and IsingZZ gates on neighbouring qubits, and measure $\langle X \rangle$, $\langle Y \rangle$ and $\langle Z \rangle$ on every qubit. (b) The PQCLP circuit on four qubits, measured on qubit 0. (c) The QGCNLP projection as executed; its outputs match the PyTorch state-vector simulator of the earlier runs to within 10^{-6} . (d) The A2QTGN encoder, an amplitude embedding $|\Psi\rangle$ of a 256-dimensional vector followed by a CNOT chain. (e) The QLPCN circuit on an 8-node path graph, with three register qubits and the ancilla; $\Phi^\top$, Φ and the controlled phase are applied as unitaries computed from the eigendecomposition. Measurement symbols denote expectation values or probabilities computed exactly.

$\mathcal{U}(-2, 2)$ once per seed, is

$$U_{\text{wall}}(r) = \bigotimes_{q=0}^{10} R_Z(\pi r_{q,3}) R_Y(\pi r_{q,2}) R_X(\pi r_{q,1}). \quad (19)$$

The data enter as bond angles $\phi_i = \tilde{x}_i n / (3T)$ for $i = 1, \dots, 10$, with $T = 10$ Trotter steps. One step applies, to each neighbouring pair $(i-1, i)$ in turn,

$$U_i(\phi_i) = e^{-i\phi_i Z_{i-1} Z_i} e^{-i\phi_i Y_{i-1} Y_i} e^{-i\phi_i X_{i-1} X_i}, \quad (20)$$

implemented as IsingXX, IsingYY and IsingZZ gates with angle $2\phi_i$. The encoded state is

$$|\psi(\tilde{x})\rangle = [U_{10}(\phi_{10}) \cdots U_1(\phi_1)]^T U_{\text{wall}}(r) |0\rangle^{\otimes 11}, \quad (21)$$

a first-order Trotterisation of $\exp(-i \sum_i \frac{n}{3} \tilde{x}_i (X_{i-1} X_i + Y_{i-1} Y_i + Z_{i-1} Z_i))$ applied to the wall state. The three terms on one bond commute, so Trotter error arises only between neighbouring bonds. The readout is the vector of one-qubit expectation values,

$$f(\tilde{x}) = (\langle X_q \rangle, \langle Y_q \rangle, \langle Z_q \rangle)_{q=0}^{10} \in [-1, 1]^{33}, \quad (22)$$

which by Eq. (5) determines every 1-RDM. The features are standardised on the training pairs, additionally min-max scaled for PQKLP, and passed to the classifier. The circuit has no trainable parameters. It matches a dense matrix exponential of the same Trotter product to 4.0×10^{-15} and an independent Cirq construction to 1.3×10^{-14} , and batched simulation of all pairs gives the same result as simulating the pairs one at a time.

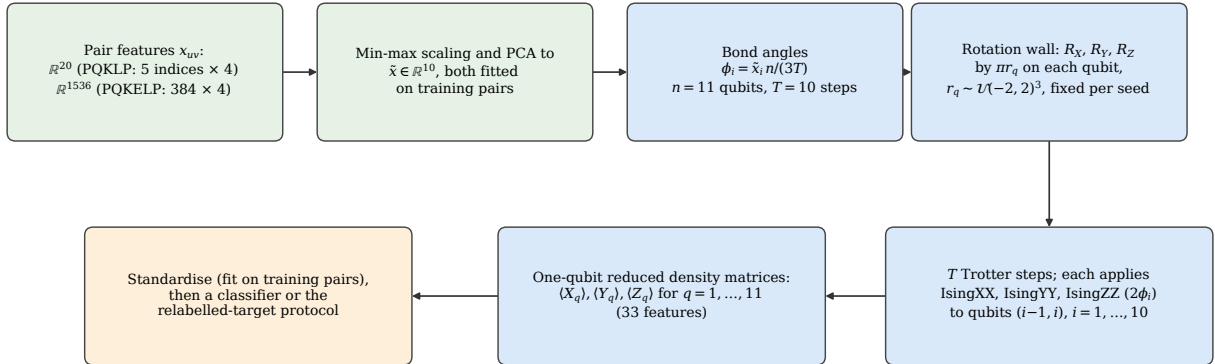


Figure 6: Encoding in PQKLP and PQKELP, as implemented in `pqk_tfq.py`. The upper row prepares the bond angles and the rotation wall. The lower row applies the Trotterised evolution, reads out the one-qubit reduced density matrices and passes the 33 features to a classifier.

4.6.2 PQCLP: the variational classifier

The features of one family are standardised, reduced by PCA to n components and rescaled to $[-\pi, \pi]$, all fitted on the training pairs, and test values are clipped to the same interval. Writing products in circuit order, the ZZ feature map with two repetitions is

$$U_{ZZ}(x) = [U_{\Phi}(x) H^{\otimes n}]^2, \quad (23)$$

$$U_{\Phi}(x) = \prod_{i=0}^{n-2} \left[\text{CNOT}_{i,i+1} R_Z^{(i+1)}(2(\pi - x_i)(\pi - x_{i+1})) \text{CNOT}_{i,i+1} \right] \bigotimes_q R_Z(2x_q),$$

and it is followed by a RealAmplitudes ansatz with two repetitions,

$$U_{\text{RA}}(w) = R_Y(w^{(2)}) U_{\text{ent}} R_Y(w^{(1)}) U_{\text{ent}} R_Y(w^{(0)}), \quad U_{\text{ent}} = \prod_{i=n-2}^0 \text{CNOT}_{i,i+1}, \quad (24)$$

where $R_Y(w^{(r)}) = \bigotimes_q R_Y(w_q^{(r)})$ and U_{ent} applies CNOT gates in reverse-linear order, giving $3n$ weights. The model output is the probability of measuring qubit 0 in state $|1\rangle$,

$$p_1(x; w) = \langle \psi(x, w) | \frac{1}{2}(I - Z_0) | \psi(x, w) \rangle, \quad (25)$$

$$|\psi(x, w)\rangle = U_{\text{RA}}(w) U_{\text{ZZ}}(x) |0\rangle^{\otimes n},$$

which is the class probability that Qiskit’s parity interpretation assigns for this circuit. The PennyLane circuit matches Qiskit’s exact state vector to 5.6×10^{-16} . The per-index columns of the PQCLP tables apply the same model to the four snapshot values of a single index.

4.6.3 QGCNLP: the quantum projection inside a graph convolution

Each of the two layers applies a GCN convolution, maps its output to rotation angles, evaluates the circuit and combines the result with the convolution output:

$$h^{(l)} = \text{ReLU}(\text{GCNConv}(H^{(l)})), \quad a^{(l)} = \pi \tanh(W_r^{(l)} h^{(l)}) \in \mathbb{R}^n, \quad (26)$$

$$H^{(l+1)} = \text{LayerNorm}(W_e^{(l)} f_Q(a^{(l)}; \theta^{(l)}) + h^{(l)}),$$

with a ReLU after the first layer. The circuit re-uploads the angles in two blocks,

$$f_Q(a; \theta) = (\langle \psi_Q(a, \theta) | Z_q | \psi_Q(a, \theta) \rangle)_{q=1}^n, \quad (27)$$

$$|\psi_Q(a, \theta)\rangle = \prod_{b=1}^2 \left[\bigotimes_q R_Z(\theta_{b,q,3}) R_X(\theta_{b,q,2}) R_Y(\theta_{b,q,1}) C_{\text{all}} \bigotimes_q R_Y(a_q) \right] |0\rangle^{\otimes n},$$

where C_{all} applies a CNOT for every ordered pair of distinct qubits, giving $6n$ trainable circuit parameters. The residual connection and the layer normalisation in Eq. (26) are ours; the published rule, Eq. (12), has neither. A pair is scored by $\hat{y}_{uv} = \sigma(\kappa \cos(H_u, H_v) + b)$ with a learnable temperature κ , initialised at 10, and a bias b . The circuit is a PennyLane QNode with the PyTorch interface and backpropagation, simulated from a single-precision start state, so its gradients are exact and flow jointly with those of the convolutions. All nodes are evaluated together by parameter broadcasting, in chunks of 4,096 nodes under gradient checkpointing. The four-qubit and eight-qubit variants are called QGCNLP-4Q and QGCNLP-8Q.

4.6.4 A2QTGN: the gated amplitude encoder

For node u , the encoder input $v_u = [e_u \| 0]$ is the node embedding padded with zeros to $2^8 = 256$ entries. A node seen for the first time is encoded directly. Afterwards, with $\Delta_u = v_u - v_u^{\text{prev}}$ and $\alpha_u = \sigma(\beta \|\Delta_u\|_2)$, the adaptive arm re-encodes

$$v'_u = v_u^{\text{prev}} + \alpha_u \Delta_u \quad \text{only if } \alpha_u > \tau, \quad (28)$$

and otherwise reuses the cached output. The circuit prepares the amplitude-encoded state of $v'_u / \|v'_u\|$, applies CNOT gates between qubits $(q, q+1)$ for $q = 0, \dots, 6$ and returns $z_u = (\langle Z_q \rangle)_{q=0}^7$. We use $\beta = 1$ and $\tau = 0.5$ because we did not find values for these parameters in the paper. Since $\sigma(\cdot) \geq \frac{1}{2}$ for non-negative arguments, this choice opens the gate whenever $\|\Delta_u\|_2 > 0$. The encode-once and always-re-encode arms are the paper’s ablations. The encoded vector enters

$$h_u = \text{MLP}_{\text{emb}}([m_u \| z_u \| e_u]), \quad \hat{y}_{uv} = \sigma(\text{MLP}_{\text{score}}([h_u \| h_v])), \quad (29)$$

where the 64-dimensional memory m_u is overwritten with h_u after each batch of true events. The classical TGN arm replaces the circuit with $\tanh(Wv_u) \in \mathbb{R}^8$. The encoder is not differentiated, and the node embedding is trained through the direct path in Eq. (29).

4.6.5 QLPCN: the controlled walk as a circuit

With the eigendecomposition $A = \Phi \Lambda \Phi^\top$ of the observed graph, the branch operators of Eq. (11) are $\cos(At_w) = \Phi \cos(\Lambda t_w) \Phi^\top$ and $\sin(At_w) = \Phi \sin(\Lambda t_w) \Phi^\top$, and a pair is scored by

$$\hat{y}_{\text{even}}(u, v) = [\cos(At_w)]_{uv}^2, \quad \hat{y}_{\text{odd}}(u, v) = [\sin(At_w)]_{uv}^2. \quad (30)$$

These probabilities are read from the executed circuit, exactly rather than from samples. PennyLane has no native gate for e^{-iAt_w} on an arbitrary graph, so, as in the paper, the time evolution is compiled classically. Both branches share the eigenbasis Φ , so the circuit applies $\Phi^\top$ and Φ to a register of $\lceil \log_2 N \rceil$ qubits and, between them, a phase $e^{\mp i \Lambda t_w}$ whose sign is controlled by the ancilla, all between two Hadamard gates on the ancilla. This is the controlled walk of Eq. (11); written directly as a controlled unitary it gives the same probabilities to 3.3×10^{-15} on small graphs. The walk times are those of the paper’s Table II (Section 5.3). On the example network distributed with the paper, the eigendecomposition reproduces the authors’ top-100 predictions for both parities.

At the shortest walk times of Table II (10^{-6} to 5×10^{-5} for the even branch), the amplitude between two distinct nodes is of order t_w^2 per common neighbour while the diagonal stays close to one. Double precision then ranks pairs without common neighbours through rounding, in the circuit and in the eigendecomposition alike. We therefore also compute the scores without this cancellation: from the power series of $\cos(At_w)$ and $\sin(At_w)$ (Section 2.7) when $t_w \lambda_{\max} < 0.1$, and otherwise by setting to zero the pairs whose amplitude is exactly zero, that is, pairs in different components or on the unreachable side of a bipartite component. On a 59-node graph with several components, this reference ranks pairs with Kendall’s τ between 0.992 and 1.000 against 60-digit arithmetic, compared with 0.740 to 0.956 for double precision.

4.7 From Measurement to Link Score

PQKLP and PQKELP pass the 33 features of Eq. (22) to a classifier, whose predicted probability is the link score. PQCLP uses $p_1(x; w)$ from Eq. (25) directly. QGCNLP and A2QTGN pass n Pauli- Z expectation values to classical layers that produce $\hat{y}_{uv}$, and QLPCN uses the branch probabilities of Eq. (30). Thresholded metrics use a threshold of 0.5 for the snapshot methods, a threshold chosen on validation data for QGCNLP and a threshold chosen on the calibration half for QLPCN.

4.8 Loss Functions

The PQCLP VQC minimises the cross-entropy and the QNN minimises the mean probability assigned to the wrong class, which is the objective of the Qiskit classifier interface for two outputs. Over N_{tr} training pairs,

$$\mathcal{L}_{\text{VQC}}(w) = -\frac{1}{N_{\text{tr}}} \sum_{k=1}^{N_{\text{tr}}} \left[y_k \ln p_1(x_k; w) + (1 - y_k) \ln(1 - p_1(x_k; w)) \right], \quad (31)$$

$$\mathcal{L}_{\text{QNN}}(w) = \frac{1}{N_{\text{tr}}} \sum_{k=1}^{N_{\text{tr}}} \left[y_k (1 - p_1(x_k; w)) + (1 - y_k) p_1(x_k; w) \right], \quad (32)$$

with probabilities clipped to $[10^{-12}, 1 - 10^{-12}]$ inside the logarithm. QGCNLP minimises the binary cross-entropy on training edges and sampled non-edges. A2QTGN minimises the sum of the binary cross-entropies on each batch of true events and on one historical negative destination per event. The classifiers of PQKLP and PQKELP minimise their own objectives with the settings listed in Section 5.3.

4.9 Training Procedures

PQCLP. The weights are initialised from $\mathcal{U}[0,1)^{3n}$ with the split seed, which is the default of the Qiskit classifiers. SPSA then runs for 100 iterations on four qubits or 50 on eight, with constant step size $\eta = 0.001$ and perturbation size $c = 0.05$ (Eq. 9), which is how Qiskit’s SPSA behaves when both are given as constants. Each iteration evaluates the loss twice over the whole training set, batched over pairs by PennyLane parameter broadcasting (Algorithm 2, Figure 7). The COBYLA variant runs SciPy’s COBYLA for the same number of iterations.

QGCNLP. Training is full-batch, with Adam (learning rate 0.01, weight decay 5×10^{-4}), a warm-up and cosine learning-rate schedule and gradient clipping at 1. Validation AUC is computed every 20 epochs. Training stops after 40 epochs without improvement or after 200 epochs, and test metrics are taken from the epoch with the best validation AUC.

A2QTGN. Training runs for 10 epochs with Adam (learning rate 10^{-3}), batches of 32 events, cosine annealing with warm restarts and gradient clipping at 1. The memory and the encoder cache are reset at the start of each epoch. Within each batch, true events and their negatives are scored against the memory as it stood before the batch, and the memory is then updated with the true events.

PQKLP, PQKELP and the classical models. The circuit is fixed, so only the classifiers are trained. They are fitted once per seed on the training pairs (Algorithm 1).

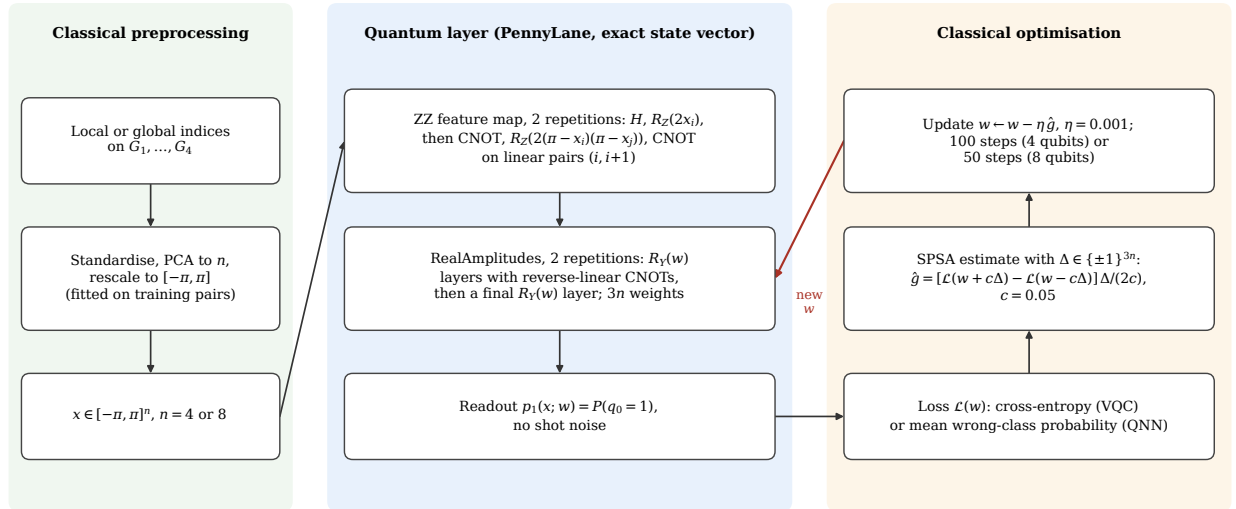


Figure 7: Hybrid quantum–classical training of PQCLP. Classical preprocessing produces rotation angles, the PennyLane circuit returns $p_1(x; w)$ for all training pairs in one broadcast evaluation, and the classical optimiser updates the $3n$ weights from two loss evaluations per SPSA step (red arrow).

4.10 The Relabelled-Target Protocol

PQKLP (Section 4 of that paper) and PQKELP (its Algorithm 1) do not score their classifiers against observed edges. Following the tutorial construction that both papers cite [23, 43], they derive new binary targets for the training and test pairs from the spectra of the projected-kernel features and of the classical features, and they report accuracy against these targets. The construction is designed to emphasise geometric differences between the quantum and classical feature

Algorithm 1 PennyLane-based projected-kernel link prediction with a matched classical counterpart (PQKLP, PQKELP)

Require: interaction stream $\mathcal{S}$; feature map φ (indices or embeddings); classifier $\mathcal{H}$ with its stated settings; target $\in \{\text{obs}, \text{new}, \text{rel}\}$

Ensure: test AUCs of the quantum and classical arms for seeds $s = 1, \dots, 10$

```

1: build the snapshots  $G_1, \dots, G_5$  (Eq. 1)
2:  $\mathcal{P} \leftarrow E_5$ , or  $E_5 \setminus \bigcup_{j \leq 4} E_j$  if target = new
3:  $\mathcal{Q} \leftarrow |\mathcal{P}|$  pairs drawn uniformly from pairs absent from  $G_1, \dots, G_5$  (seed 42)
4:  $x_{uv} \leftarrow \varphi(u, v; G_1, \dots, G_4)$  for all  $(u, v) \in \mathcal{P} \cup \mathcal{Q}$ 
5: for  $s = 1, \dots, 10$  do
6:   stratified split of  $\mathcal{P} \cup \mathcal{Q}$  with seed  $s$ 
7:   fit min-max scaling and PCA ( $d' = 10$ ) on the training pairs;  $\tilde{x} \leftarrow \text{PCA}(\text{scale}(x))$ 
8:   draw  $r \sim \mathcal{U}(-2, 2)^{11 \times 3}$  with seed  $s$ ;  $\phi \leftarrow \tilde{x} n / (3T)$ 
9:   simulate Eq. (21) on default.qubit for all pairs;  $f \leftarrow \text{Eq. (22)}$ 
10:  standardise  $f$  with training statistics (PQKLP: then min-max scale)
11:  if target = rel then
12:     $y \leftarrow$  relabelled targets from  $f$  and  $\tilde{x}$  (Eqs. 33–35); classical input  $\leftarrow$  standardised  $\tilde{x}$ 
13:  else
14:     $y \leftarrow y^{\text{obs}}$  or  $y^{\text{new}}$ ; classical input  $\leftarrow$  scaled, uncompressed  $x$  (min-max for PQKLP,
    standardised for PQKELP)
15:  end if
16:  train  $\mathcal{H}$  on  $(f, y)$  and a new instance of  $\mathcal{H}$  on (classical input,  $y$ ), using the training pairs
17:  record  $\text{AUC}_s^Q$  and  $\text{AUC}_s^C$  on the test pairs
18: end for
19: return  $\{\text{AUC}_s^Q, \text{AUC}_s^C\}_{s=1}^{10}$ 

```

Algorithm 2 Hybrid training of the PQCLP classifier with SPSA

Require: training features X of one family and labels y ; number of qubits n ; iterations K (100 for $n = 4, 50$ for $n = 8$); $\eta = 0.001$; $c = 0.05$; seed s

Ensure: trained weights w_K and test scores

```

1: fit standardisation, PCA to  $n$  components and rescaling to  $[-\pi, \pi]$  on  $X$ ; transform  $X$ 
2:  $w_0 \sim \mathcal{U}[0, 1)^{3n}$  with seed  $s$ 
3:  $\mathcal{L} \leftarrow \mathcal{L}_{\text{VQC}}$  (Eq. 31) or  $\mathcal{L}_{\text{QNN}}$  (Eq. 32), each evaluating Eq. (25) for all training pairs in one
   broadcast call
4: for  $k = 0, \dots, K - 1$  do
5:   draw  $\Delta_k \in \{-1, +1\}^{3n}$  from a generator seeded by  $s$ 
6:    $\hat{g}_k \leftarrow [\mathcal{L}(w_k + c\Delta_k) - \mathcal{L}(w_k - c\Delta_k)] \Delta_k / (2c)$ 
7:    $w_{k+1} \leftarrow w_k - \eta \hat{g}_k$ 
8: end for
9: transform the test features with the fitted preprocessing and clip them to  $[-\pi, \pi]$ 
10: return  $w_K$  and  $p_1(x; w_K)$  for every test pair

```

representations. We reproduce it as the tutorial code performs it. For the 33-dimensional PQK features F of all pairs and the 10-dimensional PCA features $\tilde{X}$,

$$\mathcal{D}_{ij}^Q = \frac{\gamma_Q}{33\sigma_F} \|f_i - f_j\|^2, \quad \mathcal{D}_{ij}^C = \frac{\gamma_C}{10\sigma_{\tilde{X}}} \|\tilde{x}_i - \tilde{x}_j\|^2, \quad (33)$$

with $\gamma_Q = 1$, $\gamma_C = 0.005$ and σ the standard deviation over all entries of the respective matrix. With the eigendecompositions $\mathcal{D}^Q = V_Q \text{diag}(\lambda^Q) V_Q^\top$ and $\mathcal{D}^C = V_C \text{diag}(\lambda^C) V_C^\top$, and with $\Sigma_Q = \text{diag}(|\lambda^Q|)$ and $\Sigma_C = \text{diag}(|\lambda^C|)$,

$$\Omega = \Sigma_Q^{1/2} V_Q^\top V_C \Sigma_C (\Sigma_C + \mu I)^{-2} V_C^\top V_Q \Sigma_Q^{1/2}, \quad \mu = 1.1. \quad (34)$$

Let U_Ω be the matrix of eigenvectors of Ω returned by an unsorted general eigensolver, and let u_Ω be its last row, as in the tutorial code. Then

$$\tilde{y} = \text{Re}(V_Q \Sigma_Q^{1/2} u_\Omega), \quad y_i^{\text{rel}} = \mathbf{1}[\tilde{y}_i > \text{median}(\tilde{y})] \oplus \xi_i, \quad \xi_i \sim \text{Bernoulli}(0.05), \quad (35)$$

where $\oplus$ denotes a label flip. The code differs from the description in two respects, which we keep: $\mathcal{D}$ is a scaled squared distance rather than a kernel, and u_Ω is a row, not a column, of an unsorted eigenvector matrix. Under this target the classical comparator is the same classifier on the standardised PCA features. For every split we record the fraction of pairs on which y^{rel} agrees with y^{obs} .

4.11 Protocol Controls

The *SeenBefore* control scores a pair by whether it was an edge of any history snapshot,

$$\hat{y}_{\text{seen}}(u, v) = \mathbf{1}\left[\{u, v\} \in \bigcup_{j=1}^4 E_j\right], \quad (36)$$

and has no parameters. Under the observed-edge target a negative is never a history edge by construction, so *SeenBefore* measures how much accuracy the recurrence of edges supplies on its own. Under the new-links target of Eq. (15) no positive is a history edge, and *SeenBefore* scores exactly 0.5. *EdgeBank* and *Persistence* play the same role for A2QTGN [18].

4.12 Comparisons and Statistical Analysis

We use three comparisons. Comparison A sets reproduced orderings against published orderings. It never compares absolute values, because the protocols differ. Comparison B is the matched comparison of Eq. (17), with the qualifications given in Section 4.1. For PQKLP and PQKELP, six classifiers on five networks give thirty comparisons per target. For PQCLP, the QNN is paired with a neural network and the VQC with a support vector machine on each of three feature families, which again gives thirty comparisons per target. Comparison C sets the best quantum configuration against the strongest classical configuration run on the same data. The classical configurations include the classifiers, the published baselines, the per-index models and *SeenBefore*.

For each matched comparison, the paired AUCs over the ten seeds are tested with a two-sided Wilcoxon signed-rank test [35]. One comparison uses nine pairs: for PQKELP under the new-links target on Radoslaw-email, the classical linear discriminant analysis did not converge on seed 10 (the singular value decomposition failed), and the test uses the nine seeds for which both arms completed. With ten pairs the smallest attainable two-sided p -value is $2/2^{10} \approx 0.002$, so the test can reject only when the difference has the same sign on nearly every seed. The p -values are adjusted with the Benjamini–Hochberg procedure [36] within each family of thirty comparisons, defined by one method and one target, and a difference is called significant at $q < 0.05$. We also report the t -based 95% confidence interval of the mean paired difference. The QGCNLP and A2QTGN comparisons, which have fewer seeds, are reported descriptively. Figure 8 summarises the evaluation workflow.

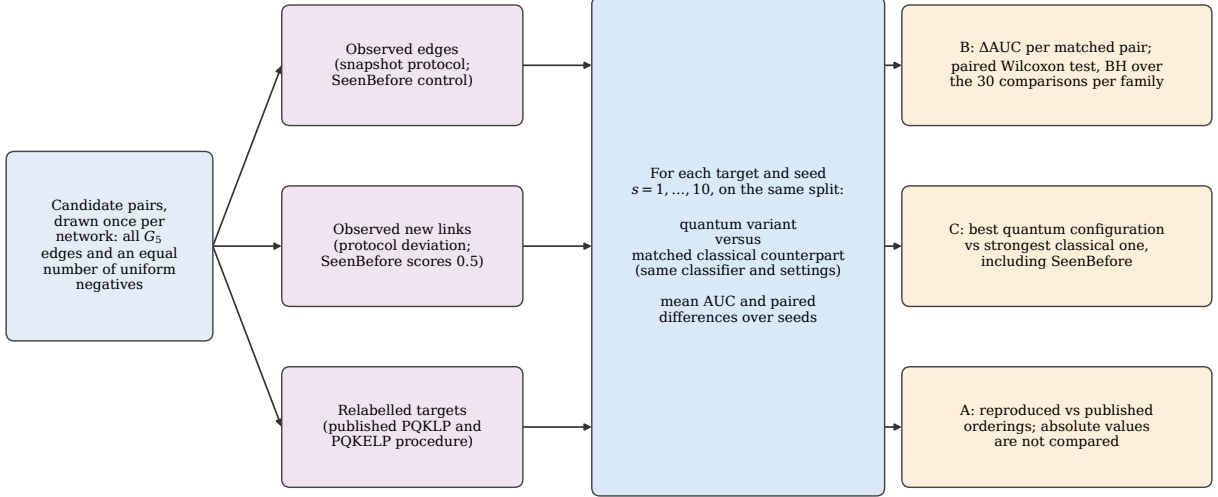


Figure 8: Evaluation workflow for the snapshot-based methods. One candidate set per network is scored under three targets. For each target and seed, the quantum variant and its matched classical counterpart share the split, and the per-seed AUCs feed Comparisons B and C. Comparison A sets the resulting orderings against those reported in the papers.

5 Experimental Setup

5.1 Research Questions and Experimental Design

Each research question in Section 1.6 maps to a set of experiments. RQ1 is addressed by the matched comparisons (Comparison B) against observed edges for PQKLP, PQCLP and PQKELP (Section 6.2). As explained in Section 4.1, these comparisons differ in feature compression as well as in the quantum stage. RQ2 is addressed by Comparison C for all six methods (Section 6.1) and by the published orderings of Comparison A (Section 6.3). RQ3 is addressed by repeating the matched comparisons against the new-links and relabelled targets, by the SeenBefore control and by the agreement between relabelled targets and observed edges (Section 6.6). RQ4 is addressed by the training-behaviour diagnostic, the sensitivity analyses and the resource counts (Sections 6.5, 6.7 and 6.8).

5.2 Datasets

Table 6 lists every network, with statistics computed from the loaded files. The five temporal networks, MIT [46], Radoslaw-email [47], Eu-core [48], fb-forum [49] and CollegeMsg [50], were obtained from KONECT [51], the Network Repository [52] and SNAP. When partitioned by Eq. (1), they reproduce the published node counts and per-snapshot edge counts exactly: MIT 1792, 1778, 817, 622 and 362; Radoslaw-email 2018, 1851, 1294, 1238 and 1428; Eu-core 8012, 8259, 9247, 5526 and 1093; fb-forum 5950, 1663, 1255, 858 and 728; and CollegeMsg 8289, 4802, 1116, 790 and 497. The papers describe the partitioning rule only informally, so agreement on all twenty-five counts indicates that the same rule was recovered.

The candidate sets of the five networks contain 724, 2856, 2186, 1456 and 994 pairs, respectively. At a test fraction of 0.2, the test sets contain 145, 572, 438, 292 and 199 pairs. The new-links test sets are much smaller and uneven, with 3, 76, 16, 42 and 128 pairs.

QGCNLP was evaluated on Cora, CiteSeer and PubMed [53, 54], WikiCS [55], PPI [14, 56] and ogbl-ddi [42], and on four of the temporal networks. The PPI graph is the subset assembled by the loader, with 1767 nodes and 16,159 edges; the paper reports 2245 nodes and 61,318 edges but does not state which subgraphs it used. A2QTGN was evaluated on 25,000 events of tgbl-wiki [17, 41],

Table 6: Datasets, with statistics computed from the files the harness loads. Interactions count timestamped events; edges are unique undirected pairs without self-loops. Average degree is $2|E|/N$ and density $2|E|/(N(N-1))$. PPI is the subset the loader assembles from the first training graphs (Section 4.3).

Dataset	Type	Nodes	Interactions	Edges	Avg. degree	Density	Used by	Source
MIT	Proximity (temporal)	96	1,086,404	2,539	52.90	5.57×10^{-1}	PQKLP, PQCLP, PQKELP, QGCNLP	[46, 51]
Radoslaw-email	E-mail (temporal)	167	82,876	3,250	38.92	2.34×10^{-1}	PQKLP, PQCLP, PQKELP	[47, 51]
Eu-core	E-mail (temporal)	986	332,334	16,064	32.58	3.31×10^{-2}	PQKLP, PQCLP, PQKELP, QGCNLP	[48]
fb-forum	Forum (temporal)	899	33,686	7,036	15.65	1.74×10^{-2}	PQKLP, PQCLP, PQKELP, QGCNLP	[49, 52]
CollegeMsg	Messaging (temporal)	1,899	59,835	13,838	14.57	7.68×10^{-3}	PQKLP, PQCLP, PQKELP, QGCNLP	[50]
tgbl-wiki (25k events)	Edits (event stream)	9,227	25,000	6,617	1.43	1.55×10^{-4}	A2QTGN	[17, 41]
Cora	Citation	2,708	–	5,278	3.90	1.44×10^{-3}	QGCNLP, QLPCN	[53, 54]
CiteSeer	Citation	3,327	–	4,552	2.74	8.23×10^{-4}	QGCNLP, QLPCN	[53, 54]
PubMed	Citation	19,717	–	44,324	4.50	2.28×10^{-4}	QGCNLP	[54]
WikiCS	Hyperlink	11,701	–	215,603	36.85	3.15×10^{-3}	QGCNLP	[55]
PPI	Protein interaction	1,767	–	16,159	18.29	1.04×10^{-2}	QGCNLP	[14, 56]
ogbl-ddi	Drug interaction	4,267	–	1,067,911	500.54	1.17×10^{-1}	QGCNLP	[42]
Facebook	Social	4,039	–	88,234	43.69	1.08×10^{-2}	QLPCN	as in [22]
Hamsterster	Social	2,426	–	16,631	13.71	5.65×10^{-3}	QLPCN	[51]
HuRI (HI-III-20)	Protein interaction	8,245	–	52,068	12.63	1.53×10^{-3}	QLPCN	[57]
P2P-Gnutella	Peer-to-peer	10,876	–	39,994	7.35	6.76×10^{-4}	QLPCN	[58]
AS routes	Autonomous systems	6,474	–	12,572	3.88	6.00×10^{-4}	QLPCN	[58]
Wiki-Vote	Voting	7,115	–	100,762	28.32	3.98×10^{-3}	QLPCN	[59]

the subset size used in the paper’s ablation, taken at even intervals over the full stream. QLPCN was evaluated on the eight of its networks available locally, three of which load with fewer edges than the paper reports (Section 5.7).

5.3 Baselines

Table 7 lists the configuration, classical counterparts and baselines of each method. For PQKLP and PQKELP, the classical configurations are the six classifiers of the paper applied without compression or circuit, the published baselines WEAK, LGQ and CTDNE (and, for PQKLP, Node2Vec and the DeepWalk–GNN hybrid), the per-index models and SeenBefore. For PQCLP, they are a neural network and a support vector machine with a degree-two polynomial kernel on each feature family, the per-index models, the published baselines and SeenBefore. QGCNLP is compared with GCN, GraphSAGE and GAT, with the quantum graph models QSGCN and GQWNN, and, on temporal networks, with Node2Vec, LGQ and PQCLP. A2QTGN is compared with its encode-once and always-re-encode ablations, the TGN backbone, EdgeBank and Persistence. QLPCN is compared with LO, L3, CH-L3, RA-L2 and CH-L2, and Table 8 lists its walk times. All baselines were run inside the same framework on the same splits, and no baseline value was copied from a paper.

5.4 Evaluation Metrics

For positive pairs $\mathcal{P}$ and negative pairs $\mathcal{Q}$ with scores $\hat{y}$, the area under the ROC curve is

$$\text{AUC} = \frac{1}{|\mathcal{P}||\mathcal{Q}|} \sum_{e^+ \in \mathcal{P}} \sum_{e^- \in \mathcal{Q}} \left(\mathbf{1}[\hat{y}_{e^+} > \hat{y}_{e^-}] + \frac{1}{2} \mathbf{1}[\hat{y}_{e^+} = \hat{y}_{e^-}] \right), \quad (37)$$

which is the probability that a randomly chosen positive is scored above a randomly chosen negative. A constant score gives exactly 0.5. The area under the precision–recall curve (AUPR) is computed as average precision, $\sum_k (R_k - R_{k-1}) P_k$ over decreasing score thresholds. Given a threshold on $\hat{y}$ and the resulting counts of true and false positives and negatives,

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP}, & \text{Recall} &= \frac{TP}{TP + FN}, \\ F_1 &= \frac{2 \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, & \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN}, \end{aligned} \quad (38)$$

Table 7: Experimental configuration of each reimplemented method as run. “Paper” marks settings taken from the source paper; other settings are ours and are listed as deviations in Table 10 where the paper states a different value.

Method	Data and split	Features and compression	Model and training	Classical counterparts	Seeds and targets
PQKLP	Five temporal networks; positives all G_5 edges; equal uniform negatives; test fraction 0.2 (paper)	CN, AA, JC, PA, SP on G_1 – G_4 (20; paper); min-max, PCA to 10 (paper)	PQK features (33; paper) into NN (1024, 1024; lr 0.001; 5 epochs; batch 32), XGB (50 trees, lr 0.01), RFC (100 trees), LR, LDA, GNB (paper)	Same classifiers on the 20 min-max scaled indices; baselines WEAK, LGQ, CTDNE, N2V, DeepWalk–GNN; SeenBefore	10 seeds; observed, new links, relabelled
PQCLP	As PQKLP	Local (AA, CN, JC, PA) or global (SP, LP, L3, CCLP) on G_1 – G_4 ; standardise, PCA to n , rescale to $[-\pi, \pi]$	QNN and VQC, $n = 4$ (100 iterations) or $n = 8$ combined (50 iterations), SPSA lr 0.001 (paper), perturbation 0.05 (ours); COBYLA variant	NN (64, 32) and polynomial SVM (degree 2) on the uncompressed family; baselines; SeenBefore	10 seeds; observed, new links
PQKELP	As PQKLP; training ratios 0.5–0.9 (paper); matched comparison at 0.8	Node2Vec, DeepWalk, Walklets (128-d per snapshot; paper) and combined (384-d); Hadamard product of endpoints (paper); four snapshots; min-max, PCA to 10	PQK features into NN (512, 256; lr 0.03; up to 1000 epochs; batch 32), XGB (100 trees, lr 0.3, depth 6), RFC, LR, LDA, GNB (paper)	Same classifiers on the standardised uncompressed features (1536); WEAK, LGQ, CTDNE; SeenBefore	10 seeds; observed, new links, relabelled
QGCNLP	Five static graphs, 85/5/10 edge split with sampled negatives (test ratio paper); OGB ddi; four temporal networks at test fraction 0.2	Normalised adjacency features $D^{-1/2}(A + I)D^{-1/2}$	Two GCN layers (64) with quantum projection on PennyLane, 4 or 8 qubits, RY/CNOT all pairs/RX RY RZ, 2 blocks (paper Algorithm 2); full-batch Adam lr 0.01 (paper), weight decay 5×10^{-4} , warm-up and cosine schedule, up to 200 epochs, validation every 20, patience 40	GCN, GraphSAGE, GAT (64); QSGCN, GQWNN; Node2Vec, LGQ and the PennyLane PQCLP VQC (COBYLA) on temporal data	5 seeds (static), 3 (circuit grid, OGB, temporal)
A2QTGN	tgbl-wiki, 25,000 events at even intervals over the stream (paper ablation size); 70/15/15 chronological (paper)	Learned node embedding (32-d); the numpy cache carries no edge features	8-qubit amplitude encoder, memory 64, embedding 64 (paper); Adam lr 10^{-3} , batch 32 (paper), 10 epochs; $\beta = 1$, $\tau = 0.5$	TGN with a linear encoder; frozen and always-update arms; EdgeBank, Persistence	1 seed (42); 100 historical negatives
QLPCN	Eight static networks; one hold-out of 10% of edges (seed 42) with an equal number of random negatives	Adjacency matrix of the remaining graph	Even and odd branch probabilities from the circuit on PennyLane, and a cancellation-free reference; walk time per network from the paper’s Table II	LO, L3, CH-L3, RA-L2, CH-L2 (paper)	10 seeds, each halving the pairs into threshold calibration and evaluation

Table 8: QLPCN walk times t_w for the even and odd branches, taken from the paper’s Table II. The paper gives no even-branch walk time for P2P-Gnutella.

Network	t_w (even)	t_w (odd)
HuRI (HI-III-20)	3×10^{-6}	1.00
Hamsterster	0.4	1.60
Facebook	0.2	1.00
Wiki-Vote	2×10^{-6}	1.00
AS routes	5×10^{-4}	0.60
P2P-Gnutella	–	1.30
CiteSeer	5×10^{-5}	1.40
Cora	1×10^{-6}	0.03

with thresholds chosen as described in Section 4.7. For A2QTGN, each test event e is scored against 100 negative destinations, and its rank is

$$\text{rank}_e = 1 + |\{\text{negatives scored higher}\}| + \frac{1}{2}|\{\text{negatives with an equal score}\}|. \quad (39)$$

The mean reciprocal rank over the test events $\mathcal{T}$ is

$$\text{MRR} = \frac{1}{|\mathcal{T}|} \sum_{e \in \mathcal{T}} \frac{1}{\text{rank}_e}. \quad (40)$$

For the QGCNLP circuit ablation, Hits@50 is the number of positive pairs among the 50 highest-scored test pairs divided by $\min(50, |\mathcal{P}|)$. Runtime is the wall-clock time for fitting a model, simulating a batch of circuits or completing a training epoch.

5.5 Quantum Implementation Using PennyLane

Table 9 gives the configuration of every circuit. The circuits of all six methods were implemented in PennyLane 0.44.1 on the `default.qubit` device, with analytic expectation values and no shots. The PQK, PQCLP and QGCNLP circuits use parameter broadcasting to evaluate all pairs or nodes of a batch in one call. The QGCNLP projection uses PennyLane’s PyTorch interface, so its gradients flow jointly with those of the graph convolutions, and it is simulated in single precision, as was the PyTorch state-vector simulator of our earlier runs. The QLPCN circuit is described in Section 4.6.5. No experiment used a noise model or quantum hardware.

The original PQKLP and PQKELP circuits were TensorFlow Quantum constructions [43], and PQCLP was a Qiskit model [44]. We checked each port against an independent reference. The PQK circuit matches a dense matrix exponential of the same Trotter product to 4.0×10^{-15} and a Cirq construction to 1.3×10^{-14} . The PQCLP circuit matches Qiskit’s exact state vector to 5.6×10^{-16} , and on Radoslaw-email over three seeds the PennyLane and Qiskit PQCLP models reach mean AUCs of 0.6708 and 0.6716 (VQC) and 0.5771 and 0.5741 (QNN). The PennyLane QGCNLP projection matches the earlier PyTorch simulator to 9×10^{-7} in its outputs and to 1.6×10^{-5} (relative) in its gradients for all six circuit patterns, and on PubMed the two gave training losses within 3.2×10^{-3} over 30 epochs. The QLPCN eigendecomposition reproduces the authors’ top-100 predictions for both parities on the example network distributed with the paper, and the QLPCN circuit matches it to 3.3×10^{-15} on small graphs.

Seeds. The candidate pairs were drawn with seed 42. The snapshot methods use seeds 1 to 10, and each seed fixes the split, the model initialisation, the PQK rotation wall and the SPSA directions. QGCNLP uses seeds 1 to 5 in the qubit-count runs and seeds 1 to 3 in the comparison with GraphSAGE, GAT, QSGCN and GQWNN, the circuit ablation, the OGB experiment and the temporal experiments. A2QTGN uses seed 42, and QLPCN uses seeds 1 to 10.

Table 9: Quantum-circuit configuration as implemented. All circuits are simulated exactly (analytic expectation values, no shots). PennyLane device `default.qubit`, PennyLane 0.44.1. The QGCNLP projection starts from a single-precision state and is trained inside the PyTorch model by backpropagation. Gate and depth counts are in Table 23.

Method	Simulator	Qubits	Encoding	Ansatz / layers	Trainable	Measurement	Optimiser	Loss
PQKLP, PQKELP	PennyLane	11	Rotation wall $R_X, R_Y, R_Z(\pi r_q)$, $r_q \sim \mathcal{U}(-2, 2)^3$ fixed per seed; bond angles $\phi_i = \tilde{x}_i n / (3T)$	$T = 10$ Trotter steps of IsingXX, IsingYY, IsingZZ($2\phi_i$) on qubits $(i-1, i)$, $i = 1, \dots, 10$	0	$\langle X_q \rangle, \langle Y_q \rangle, \langle Z_q \rangle$; 33 values	None in the circuit; classifiers as in Table 7	Classifier losses
PQCLP	PennyLane	4 or 8	ZZ feature map, 2 repetitions, linear pairs, $x \in [-\pi, \pi]^n$	RealAmplitudes, 2 repetitions, reverse-linear CNOTs, final R_Y layer	$3n$ (12 or 24), initialised $\mathcal{U}[0, 1)$	$P(q_0=1)$	SPSA, $\eta = 0.001$, $c = 0.05$, 100 or 50 steps; COBYLA variant	Cross-entropy (VQC); mean wrong-class probability (QNN) BCE with logits
QGCNLP	PennyLane	4 or 8	$R_Y(\pi \tanh(Wh))$ of the aggregated representation h	2 blocks of RY encoding, CNOT on all ordered pairs, trainable RY, RX, RZ	$6n$ (24 or 48)	$\langle Z_q \rangle, n$ values	Adam, lr 0.01, weight decay 5×10^{-4}	BCE with logits
A2QTGN	PennyLane	8	Amplitude embedding of a 256-d vector (normalised, zero-padded)	One CNOT chain	0	$\langle Z_q \rangle, 8$ values	Adam on classical parts, lr 10^{-3}	BCE, one negative per positive
QLPCN	PennyLane	$\lceil \log_2 N \rceil + 1$ (13–15)	Basis state $ j\rangle$ of a node, batched	Hadamard on the ancilla; $\Phi^\top$; phase $e^{\mp i \Lambda t w}$ controlled by the ancilla; Φ ; Hadamard	0	Probabilities of ancilla and register, Eq. (30)	None	None

5.6 Computing Environment

All experiments ran on a single Apple M4 machine with 10 CPU cores and 16 GB of unified memory, under macOS 26.6.2 with Python 3.13.9. The software comprised PennyLane 0.44.1 [33], NumPy 2.4.2, SciPy 1.16.3, NetworkX 3.5, scikit-learn 1.7.2 [60], XGBoost 3.2.0 [40], PyTorch 2.10.0 [34], PyTorch Geometric 2.7.0 [45], OGB 1.3.6 [42], node2vec 0.5.0, Karate Club 1.2.2 [61] and gensim 4.4.0. Qiskit 2.4.0 with qiskit-machine-learning 0.9.0, and Cirq 1.6.1, were used for validation only. These versions were recorded for the PQKLP, PQCLP and PQKELP runs of 13 September 2026 and apply to the QGCNLP and QLPCN runs of 14 and 15 September 2026, which used the same environment. The A2QTGN results come from a run on 6 September 2026 on the same machine, for which versions were not recorded separately. No discrete GPU was used. The classical layers of A2QTGN select the Apple MPS device when it is available, and all other computation ran on the CPU. The PQKELP walk embeddings were trained with a single worker thread for reproducibility.

5.7 Reproduction Fidelity

Building each method from its specification revealed thirteen implementation discrepancies, points at which a direct reading of a paper would diverge from the behaviour the paper describes. Table 10 lists them together with the deviations that remain. We use the word *discrepancy* neutrally. In most cases the paper is unambiguous and a naive reimplementing would be wrong; in others the specification allows more than one reading.

The study keeps four categories apart. A *published result* is a value reported in an original paper, and a *reproduced result* is a value measured in our framework. An *implementation deviation* is a difference between a specification and our code. A *protocol deviation* is a difference in the evaluation

procedure; the A2QTGN negative sampler and the new-links target are the two instances. The relabelled-target protocol is not a deviation, because it reproduces the published procedure.

Table 10: Reproduction fidelity, part 1 of 2: implementation discrepancies identified and resolved, where a direct reading of a specification would diverge from the behaviour the paper describes. Part 2 lists the deviations that remain.

Issue	Method	Specification	Our implementation
<i>Discrepancies identified and resolved</i>			
Per-index baselines	PQKLP, PQCLP	Columns headed CN, AA, JC, PA, SP are models trained on that index across the history snapshots	PQKLP: a single-feature classifier. PQCLP: the table’s own quantum model on the four snapshot values
Negative sampling	PQKLP	Uniformly random non-edges	Uniform sampling among pairs absent from every snapshot
Shortest path	PQKLP, PQCLP	$SP(x, y) = d(x, y) $, the raw distance	Raw distance with sentinel -1 for disconnected pairs
Qubits and width	PQKLP, PQKELP	Ten principal components on eleven qubits, $11 \times 3 = 33$ features	As specified
PQK circuit	PQKLP, PQKELP	Random single-qubit X , Y , Z rotation wall, then $\exp(-i \sum_i \phi_i (X_{i-1} X_i + Y_{i-1} Y_i + Z_{i-1} Z_i))$ in ten Trotter steps, 1-RDM readout	As specified; agrees with a dense matrix exponential to 4.0×10^{-15} and with an independent Cirq construction to 1.3×10^{-14}
Classifier input	PQKLP	A conventional classifier on the projected-kernel features	The 33 PQK features alone, not concatenated with the indices
Classifier settings	PQKLP, PQKELP	Stated layer sizes, learning rates, epochs, batch sizes and ensemble sizes	As stated (Table 7)
Edge operator	PQKELP	Element-wise product of endpoint embeddings	Element-wise product, not concatenation
Embedding methods	PQKELP	Node2Vec ($P = Q = 1$, 10 walks of length 80, dimension 128, window 10), DeepWalk (128, window 5), Walklets (32 per scale, four scales), combined	node2vec and karateclub libraries with these parameters; combined variant concatenates the three
Qubits, epochs, optimiser	PQCLP	Four qubits, SPSSA, 100 epochs, lr 0.001 (local, global); eight qubits, 50 epochs (combined)	As stated; a separate statement preferring COBYLA in simulation is run as a variant
QNN objective	PQCLP	Qiskit classifier interface	The objective that interface minimises for two outputs: mean probability of the wrong class
QGCNLP circuit	QGCNLP	RY encoding, CNOT on every qubit pair, parameterised RX, RY, RZ	As specified
Batch size	A2QTGN	Batch size 32	Batch size 32

6 Results

Unless stated otherwise, AUC is measured against observed edges and reported as the mean over ten seeds. A *win* means a higher mean AUC and carries no claim of significance; significance was assessed as described in Section 4.12.

6.1 Main Performance Results

Table 11 compares the best quantum configuration of each snapshot-based method with the strongest classical configuration run on the same data (Comparison C). No snapshot-based method exceeded the strongest classical configuration on any of the five temporal networks. The mean gap was -0.044 AUC for PQKLP, -0.395 for PQCLP and -0.160 for PQKELP, and the largest gaps were -0.100 , -0.484 and -0.240 , respectively. The best PQKLP configuration was the random forest on projected-kernel features on every network, and its gap was smallest on Eu-core. In 13 of the 15 method-network cells, the strongest classical configuration was WEAK or LGQ, both of which use the snapshot sequence directly; Section 6.6.1 examines how much accuracy the recurrence of edges supplies.

On the static benchmarks (Table 12), QGCNLP-8Q scored below the strongest of GCN, GraphSAGE and GAT on all five graphs, by 0.076 to 0.270 AUC (mean -0.136). Against GCN alone, in the separate runs of the qubit-count experiment, the gaps were 0.057 to 0.247 (mean -0.120).

Table 10: Reproduction fidelity, part 2 of 2 (continued): deviations that remain and bound the claims of Section 6.

Issue	Method	Specification	Our implementation
<i>Remaining deviations</i>			
Neural heads	PQKLP, PQKELP	Keras networks	scikit-learn MLPs with the stated sizes, learning rate and batch size; PQKELP training stops after ten epochs without improvement
Rotation wall	PQKLP, PQKELP	Tutorial code draws separate walls for training and test sets	One wall per seed, shared by training and test pairs
Embedding threads	PQKELP	Four to eight workers	One worker, for reproducibility under a fixed seed
SPSA	PQCLP	Not stated	$c = 0.05$
perturbation	PQKELP	The paper reports no classical counterpart	Standardised inputs; with min-max inputs the NN counterpart collapses to a constant output (Section 6.2)
Classical inputs of the matched comparison	PQKELP		
Simulator and layer	QGCNLP	Eq. (12), without a residual branch	PennyLane circuit inside the PyTorch model (single precision, backpropagation); the circuit output is expanded to the hidden width, added to the convolution output and layer-normalised; angles $\pi \tanh(Wh)$
Message passing	QGCNLP	Not stated	Message passing uses the full edge set, including held-out edges, for QGCNLP and its classical comparators; this can inflate absolute AUC, while comparisons among these models remain internally controlled
PPI subset	QGCNLP	2245 nodes, 61,318 edges	1767 nodes, 16,159 edges assembled by the loader
Negative candidates	A2QTGN	Official TGB candidate lists	Historical sampler; absolute MRR not comparable
Stream features	A2QTGN	172-d edge features	None in the local cache; a learned 32-d node embedding carries identity; β, τ not located in the paper ($\beta = 1, \tau = 0.5$ used)
Runs and subset	A2QTGN	Full streams; FakeTorino and hardware runs	One seed on the 25,000-event subset; noise-free simulation only
Sampling	QLPCN	Samples drawn from the circuit	Probabilities read exactly from the executed circuit, without sampling; at the shortest walk times of Table II they are limited by double precision, so a cancellation-free reference is reported beside them (Section 6.1)
Evaluation splits	QLPCN	Ten-fold cross-validation, 10% of links removed per fold	One 10% hold-out; seeds redraw the calibration/evaluation halves only
Temporal comparator	QGCNLP	PQCLP as published	PennyLane VQC with COBYLA, 4 qubits, 100 iterations
Network sizes	QLPCN	HuRI 52,569, AS routes 13,895, Wiki-Vote 103,689 edges	52,068, 12,572 and 100,762 edges after loading

Message passing included the held-out edges for all of these models, so the absolute values may be inflated, but the comparison among the models is internally controlled. On ogbl-ddi, QGCNLP-8Q reached 0.887, compared with 0.843 for GraphSAGE and 0.840 for GCN. This is the only case outside QLPCN in which a quantum configuration exceeded the strongest classical one. The gap of 0.044 exceeds the standard deviation of QGCNLP-8Q over its three seeds (0.005), but three seeds do not support a significance test, and the earlier runs with the PyTorch simulator gave 0.857 against 0.842 with a standard deviation of 0.056. On the four temporal networks, QGCNLP scored below LGQ on all four.

QLPCN exceeded the strongest classical path-based predictor on three of the eight networks (CiteSeer, Cora and P2P-Gnutella), with a mean gap of -0.023 and a largest gap of -0.100 on HuRI; the lead on CiteSeer was 0.001. The even branch scored higher than the odd branch on the social networks Facebook (0.980 against 0.951), Hamsterster (0.957 against 0.895) and Wiki-Vote (0.925 against 0.796). The odd branch scored higher on HuRI (0.826 against 0.774), AS routes (0.689 against 0.655) and Cora (0.823 against 0.725). On P2P-Gnutella, where the paper gives no even-branch walk time, the odd branch reached 0.628, the highest value of any method on that network.

Table 11: Best quantum configuration against the strongest classical configuration on the five temporal networks (Comparison C), scored against the observed edges; mean $\pm$ standard deviation over ten seeds. Precision, recall and F1 use a threshold of 0.5 on the predicted probability. Classical configurations include the published baselines (WEAK, LGQ, CTDNE) and SeenBefore. PQCLP counts its QNN and VQC variants only; PQKELP uses the combined embedding at training ratio 0.8.

Method	Network	Best quantum configuration					Strongest classical		
		Model	AUC	Prec.	Rec.	F1	Model	AUC	Gap
PQKLP	MIT	PQKLP-RFC	0.899 $\pm$ 0.016	0.782	0.854	0.815	WEAK	0.999 $\pm$ 0.001	-0.100
	Radoslaw-email	PQKLP-RFC	0.958 $\pm$ 0.009	0.862	0.915	0.888	WEAK	0.983 $\pm$ 0.004	-0.026
	Eu-core	PQKLP-RFC	0.984 $\pm$ 0.005	0.926	0.960	0.943	WEAK	0.997 $\pm$ 0.002	-0.013
	fb-forum	PQKLP-RFC	0.923 $\pm$ 0.019	0.807	0.892	0.847	LGQ	0.965 $\pm$ 0.012	-0.042
	CollegeMsg	PQKLP-RFC	0.847 $\pm$ 0.028	0.689	0.926	0.790	LGQ	0.884 $\pm$ 0.024	-0.038
PQCLP	MIT	VQC (global)	0.515 $\pm$ 0.051	0.507	0.600	0.548	NN (combined)	0.999 $\pm$ 0.001	-0.484
	Radoslaw-email	VQC (local)	0.510 $\pm$ 0.115	0.502	0.614	0.550	WEAK	0.983 $\pm$ 0.004	-0.473
	Eu-core	VQC (local)	0.679 $\pm$ 0.164	0.621	0.594	0.588	WEAK	0.997 $\pm$ 0.002	-0.317
	fb-forum	VQC (global)	0.599 $\pm$ 0.120	0.594	0.549	0.566	LGQ	0.964 $\pm$ 0.012	-0.365
	CollegeMsg	VQC (combined)	0.555 $\pm$ 0.144	0.664	0.477	0.496	LGQ	0.891 $\pm$ 0.022	-0.335
PQKELP	MIT	PQKE-LDA	0.759 $\pm$ 0.047	0.695	0.705	0.698	WEAK	0.999 $\pm$ 0.001	-0.240
	Radoslaw-email	PQKE-RFC	0.783 $\pm$ 0.029	0.687	0.821	0.748	WEAK	0.985 $\pm$ 0.005	-0.202
	Eu-core	PQKE-RFC	0.809 $\pm$ 0.015	0.732	0.732	0.731	WEAK	0.995 $\pm$ 0.003	-0.186
	fb-forum	PQKE-RFC	0.868 $\pm$ 0.020	0.775	0.808	0.791	WEAK	0.960 $\pm$ 0.009	-0.092
	CollegeMsg	PQKE-RFC	0.800 $\pm$ 0.036	0.661	0.880	0.754	XGB	0.881 $\pm$ 0.021	-0.081

Table 12: Best quantum against strongest classical configuration for QGCNLP and QLPCN. QGCNLP static: QGCNLP-8Q against GCN, GraphSAGE and GAT, three seeds (the runs of QGCNLP’s Table 2); message passing includes held-out edges for all four models, so absolute values may be inflated. OGB and temporal rows: three seeds; the temporal comparators are Node2Vec, LGQ and PQCLP (the PennyLane VQC trained with COBYLA), as in the QGCNLP paper. QLPCN: QLP-Even/QLP-Odd from the circuit executed on PennyLane against LO, L3, CH-L3, RA-L2 and CH-L2 on one 10% edge hold-out, with ten seeds that redraw the half of the pairs used for evaluation; precision and recall are not recorded for this protocol. The last two columns give the best QLPCN branch computed without double-precision cancellation (Section 4.6.5) and its gap to the same classical predictor.

Method	Network	Best quantum configuration					Strongest classical			QLPCN reference	
		Model	AUC	Prec.	Rec.	F1	Model	AUC	Gap	AUC	Gap
QGCNLP (static)	Cora	QGCNLP-8Q	0.855 $\pm$ 0.006	0.754	0.846	0.797	GAT	0.990 $\pm$ 0.001	-0.135	-	-
	CiteSeer	QGCNLP-8Q	0.883 $\pm$ 0.018	0.762	0.891	0.821	GAT	0.993 $\pm$ 0.001	-0.111	-	-
	PubMed	QGCNLP-8Q	0.718 $\pm$ 0.032	0.658	0.957	0.779	GAT	0.989 $\pm$ 0.001	-0.270	-	-
	WikiCS	QGCNLP-8Q	0.884 $\pm$ 0.008	0.765	0.873	0.815	GAT	0.960 $\pm$ 0.001	-0.076	-	-
	PPI	QGCNLP-8Q	0.830 $\pm$ 0.007	0.703	0.916	0.795	GCN	0.916 $\pm$ 0.013	-0.086	-	-
QGCNLP (OGB)	ogbl-ddi	QGCNLP-8Q	0.887 $\pm$ 0.005	0.785	0.898	0.838	GraphSAGE	0.843 $\pm$ 0.014	+0.044	-	-
QGCNLP (temporal)	MIT	QGCNLP	0.861 $\pm$ 0.086	0.747	0.968	0.842	LGQ	0.997 $\pm$ 0.002	-0.136	-	-
	Eu-core	QGCNLP	0.813 $\pm$ 0.033	0.698	0.882	0.778	LGQ	0.997 $\pm$ 0.002	-0.185	-	-
	fb-forum	QGCNLP	0.695 $\pm$ 0.067	0.577	0.893	0.698	LGQ	0.961 $\pm$ 0.023	-0.266	-	-
	CollegeMsg	QGCNLP	0.661 $\pm$ 0.084	0.612	0.856	0.712	LGQ	0.879 $\pm$ 0.043	-0.218	-	-
QLPCN	Facebook	QLP-Even	0.980 $\pm$ 0.001	-	-	0.933	RA-L2	0.994 $\pm$ 0.000	-0.014	0.980 (Even)	-0.014
	Hamsterster	QLP-Even	0.957 $\pm$ 0.003	-	-	0.900	RA-L2	0.973 $\pm$ 0.003	-0.016	0.956 (Even)	-0.016
	Cora	QLP-Odd	0.823 $\pm$ 0.016	-	-	0.788	L3	0.785 $\pm$ 0.010	+0.039	0.833 (Even)	+0.049
	CiteSeer	QLP-Even	0.723 $\pm$ 0.014	-	-	0.611	L3	0.722 $\pm$ 0.010	+0.001	0.755 (Even)	+0.033
	Wiki-Vote	QLP-Even	0.925 $\pm$ 0.002	-	-	0.884	L3	0.959 $\pm$ 0.002	-0.033	0.942 (Even)	-0.016
	HuRI (HI-III-20)	QLP-Odd	0.826 $\pm$ 0.003	-	-	0.752	L3	0.927 $\pm$ 0.003	-0.100	0.888 (Even)	-0.039
	P2P-Gnutella	QLP-Odd	0.628 $\pm$ 0.007	-	-	0.561	L3	0.609 $\pm$ 0.005	+0.019	0.628 (Odd)	+0.019
	AS routes	QLP-Odd	0.689 $\pm$ 0.013	-	-	0.687	L3	0.770 $\pm$ 0.010	-0.081	0.692 (Odd)	-0.077

These values come from the circuit in double precision. With the cancellation-free reference of Section 4.6.5, the even branch rose to 0.888 on HuRI, 0.833 on Cora, 0.942 on Wiki-Vote and 0.755 on CiteSeer, and the odd branch to 0.736 on CiteSeer; every other branch changed by at most 0.003. With the reference scores, QLPCN exceeded the strongest classical predictor on the same three networks, by 0.033 (CiteSeer), 0.049 (Cora) and 0.019 (P2P-Gnutella), the mean gap was -0.008 and the largest shortfall was -0.077 on AS routes, and the even branch overtook the odd branch on HuRI and Cora. We describe both results as approximate parity.

Table 13: A2QTGN, its update-strategy ablations and memorisation baselines on 25,000 evenly spaced events of tgbl-wiki, from a single run (seed 42) with our historical negative sampler and without edge features; absolute values are not comparable with the published ones. Circuit calls and encode rate are counted during evaluation; training time in seconds.

Model	AUC	MRR	Circuit calls	Encode rate	Train (s)
A2QTGN (adaptive gate)	0.8182	0.3135	9,227	0.012	497.2
A2QTGN (encode once)	0.8125	0.3224	9,227	0.012	118.9
A2QTGN (always re-encode)	0.8119	0.3019	765,000	1.000	810.2
TGN (classical encoder)	0.8084	0.3266	–	–	36.4
EdgeBank	0.8497	0.7018	–	–	0.0
Persistence	0.8498	0.7050	–	–	0.0

Table 13 reports the single A2QTGN run. Under our implementation and sampling protocol, the adaptive gate ranked first among the learned models on AUC, ahead of the encode-once arm, the always-re-encode arm and the classical TGN. On MRR the order was different: TGN ranked first, followed by the encode-once arm, with the adaptive arm third. Every difference on which these orderings rest is at most 0.013, so a single run cannot establish them. During evaluation, the adaptive and encode-once arms each executed the circuit 9,227 times, once per node, compared with 765,000 executions for the always-re-encode arm. EdgeBank and Persistence, which predict previously observed interactions, reached more than twice the MRR of every learned model.

6.2 Quantum Variants against Matched Classical Counterparts

Table 14 and Figure 9 summarise Comparison B. Under the observed-edge and new-links targets, the quantum arm of PQKLP, PQKELP and PQCLP includes compression to a few principal components, whereas the classical arm uses the uncompressed features (Section 4.1). These rows therefore compare pipelines, not the circuit in isolation. The relabelled-target rows compare the same principal components with and without the circuit.

Observed edges. PQKLP won two of its thirty comparisons, with logistic regression and linear discriminant analysis on fb-forum, by 0.004 and 0.010 AUC. Neither difference was significant ($q = 0.72$ and $q = 0.35$), and 26 of the 30 differences were significant in favour of the classical counterpart. PQKELP won five of its thirty comparisons. One was significant: linear discriminant analysis on fb-forum, by 0.110 ($q = 0.003$), where the classical arm applies LDA to 1536 standardised features. Twenty-two of the 30 PQKELP differences were significant in favour of the classical counterpart. PQCLP won none of its thirty comparisons, and all 30 differences were significant in favour of the classical counterpart. For the two kernel methods together, the quantum variant won 7 of 60 comparisons; 1 of these 60 differences was significant in favour of the quantum variant and 48 in favour of the classical counterpart. When PQCLP is included, 1 of the 90 observed-edge comparisons was significant in favour of a quantum variant and 78 in favour of the classical counterpart.

Classifier capacity. Against observed edges, the kernel-method comparisons did not separate by the capacity of the downstream classifier (Table 15). Linear and naive Bayes classifiers did not gain more from the quantum pipeline than neural networks and tree ensembles did.

Table 14: Matched comparisons (Comparison B), thirty per row. Each row pools six classifiers on five networks (PQKLP, PQKELP) or two quantum variants on three feature families and five networks (PQCLP). A win is a higher mean AUC over the ten seeds (nine for one comparison; Appendix A); significance is a two-sided paired Wilcoxon signed-rank test with Benjamini–Hochberg correction within each row, $q < 0.05$. Under the observed-edge and new-links targets the classical arm receives uncompressed features, so these rows compare the quantum pipeline, compression included, with a classical pipeline; the relabelled-target rows compare the same ten principal components with and without the circuit. PQKELP uses the combined embedding at training ratio 0.8, with standardised classical inputs (with min-max inputs the observed-edge row reads 7 wins, mean -0.071 ; Section 6.2).

Method	Target	n	Quantum		Mean		BH $q < 0.05$	
			wins		ΔAUC	Range	$Q > C$	$C > Q$
PQKLP	Observed edges	30	2		-0.058	-0.169 to $+0.010$	0	26
	New links only	30	0		-0.112	-0.400 to -0.013	0	16
	Relabelled targets	30	29		$+0.075$	-0.0002 to $+0.294$	27	0
PQKELP	Observed edges	30	5		-0.100	-0.242 to $+0.110$	1	22
	New links only	30	2		-0.187	-0.450 to $+0.103$	0	17
	Relabelled targets	30	30		$+0.192$	$+0.033$ to $+0.370$	30	0
PQCLP	Observed edges	30	0		-0.399	-0.539 to -0.177	0	30
	New links only	30	0		-0.419	-0.850 to -0.087	0	30

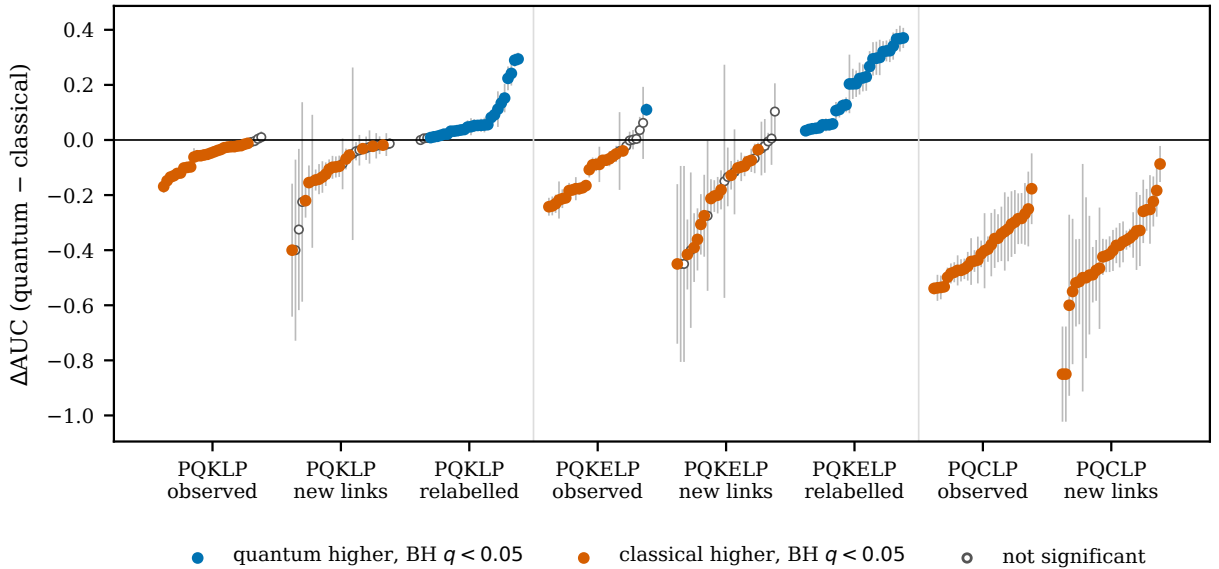


Figure 9: Every matched comparison: ΔAUC (quantum minus classical) with its 95% confidence interval over ten seeds, sorted within each method and target. Filled points are significant after Benjamini–Hochberg correction within the group ($q < 0.05$). Observed-edge and new-link groups compare the compressed quantum pipeline with a classical pipeline on uncompressed features; relabelled-target groups use the same principal components in both arms.

Table 15: Matched comparisons of PQKLP and PQKELP partitioned by the capacity of the downstream classifier: quantum wins out of fifteen, mean ΔAUC in parentheses. Observed-edge and new-link columns compare compressed quantum features with uncompressed classical features.

Method	Classifiers	Observed edges	New links	Relabelled
PQKLP	LR, LDA, GNB	2 (−0.063)	0 (−0.107)	15 (+0.122)
	NN, RFC, XGB	0 (−0.053)	0 (−0.117)	14 (+0.029)
PQKELP	LR, LDA, GNB	3 (−0.086)	1 (−0.172)	15 (+0.261)
	NN, RFC, XGB	2 (−0.114)	1 (−0.201)	15 (+0.122)

PQKELP classical inputs. The PQKELP classical counterpart receives standardised inputs. With the harness’s default min-max scaling, the 512–256 neural network with the stated learning rate of 0.03 returned a constant score on 38 of 50 seeds. The observed-edge row of Table 14 would then read seven wins, four of them significant, with the neural-network comparisons contributing gains of up to 0.276 AUC against a classifier that had not trained. A diagnostic on fb-forum confirmed the cause. On seeds 1 to 3, the output of the min-max network varied by less than 10^{-7} (AUC 0.5000), whereas the same network on standardised inputs reached 0.812, 0.893 and 0.728, and on min-max inputs with a learning rate of 0.001 it reached 0.903, 0.911 and 0.880. With standardised inputs, the neural network still returned a constant score on 4 of 50 seeds.

New links only. PQKLP won none of its thirty comparisons (mean −0.112), PQKELP two (mean −0.187) and PQCLP none (mean −0.419). No difference in favour of a quantum variant was significant. The new-links test sets contain only 3 pairs on MIT and 16 on Eu-core, so these two networks provide weak evidence; they are included in the counts but not interpreted individually.

6.3 Comparison among Quantum Methods and with Published Orderings

On identical data, the best configurations of the three snapshot-based methods followed the same order on all five networks. The best PQKLP configuration exceeded the best PQKELP configuration, by 0.047 to 0.175 AUC, which in turn exceeded the best PQCLP configuration. Table 16 sets these and other orderings against those reported in the papers (Comparison A). Orderings that correspond to large published effects, such as PQCLP ranking lowest, also appeared in the reproduction. Under our conditions, the orderings that support the accuracy claims of QGCNLP did not reproduce: QGCNLP-8Q exceeded GCN on none of the five static graphs, and QGCNLP exceeded PQCLP on two of the four temporal networks. For A2QTGN, within a single run using our sampler, the accuracy gain of the adaptive gate was a fraction of a percentage point rather than tens of points, and only the AUC ordering held. The QGCNLP qubit-count trend and the QLPCN parity claim did reproduce.

6.4 Dataset-Wise Analysis

MIT. The densest network (density 0.557, average degree 52.9) has the highest share of positives that recur from the history, 98.1%. SeenBefore reached 0.992 AUC and WEAK 0.999, whereas the best quantum configuration (PQKLP) reached 0.899 and PQCLP stayed close to chance. The new-links test set contains three pairs.

Radoslaw-email. Of the four networks on which most positives recur (86.8% here), this is the only one on which the best PQKLP configuration (0.958) exceeded SeenBefore (0.931) by more than 0.01. It remained 0.026 below WEAK.

Eu-core. Here 96.4% of positives recur. SeenBefore reached 0.982, and PQKLP reached 0.984, which is 0.013 below WEAK and the smallest Comparison C gap of any snapshot-based method.

Table 16: Comparison A: orderings reported in the source papers against the same orderings in the reproduction. Published values are from the source tables or text; reproduced values are from Tables 11–13 and the matched comparisons. Absolute values are not compared, because the protocols differ.

Ordering	Published	Reproduced	Transfers
PQCLP has the lowest AUC of the snapshot-based methods	Lowest on 5 of 5 networks (Table 2)	Best configuration lowest on 5 of 5	Yes
PQKLP (2022) against combined-embedding PQKELP (2025)	PQKLP higher on 3 of 5	PQKLP higher on 5 of 5, by 0.047–0.175	Direction only
PQKLP quantum classifiers against the same classifiers without the circuit (PQKLP Table 8)	Quantum higher AUC in 14 of 30	Observed edges: 2 of 30; relabelled targets: 29 of 30	No
QGCNLP against PQCLP, AUC, four temporal networks (QGCNLP Table 5)	QGCNLP higher on 4 of 4	QGCNLP higher on 2 of 4	No
QGCNLP-8Q against GCN, AUC, five static graphs (QGCNLP Table 1)	QGCNLP-8Q higher on 4 of 5	Higher on 0 of 5	No
QGCNLP-8Q against QGCNLP-4Q, AUC, five static graphs (QGCNLP Table 1)	QGCNLP-8Q higher on 3 of 5	Higher on 5 of 5	Yes
A2QTGN adaptive gate against always-update and no-update, 25k-event tgbl-wiki	Test accuracy +13.36 and +22.44 percentage points	One run with our sampler: accuracy +0.15 and –0.69 points; AUC highest, 0.006 above the next arm; MRR third of four	AUC ordering only
QLPCN parity with classical path-based predictors	Parity claimed	Higher on 3 of 8 networks, mean gap –0.023 (reference scores: 3 of 8, –0.008)	Yes
QLPCN even branch on social networks, odd branch on biological and peer-to-peer networks	Predicted	Even leads on Facebook, Hamsterster, Wiki-Vote; odd leads on HuRI with double-precision scores but even leads with the reference scores	Partly

The new-links test set contains 16 pairs.

fb-forum. Both PQKLP wins and the one significant PQKELP win occurred on this network. In the significant case, the classical arm applies linear discriminant analysis to 1536 standardised features and reaches 0.731, the lowest AUC of the six classical classifiers on this network, whereas the quantum arm applies it to 33 features derived from ten principal components. This comparison does not separate the effect of compression from that of the circuit (Section 7).

CollegeMsg. The sparsest temporal network (density 7.7×10^{-3}) is the only one on which most positives are new (35.6% recur). SeenBefore fell to 0.677 and was exceeded by the best PQKLP (0.847) and PQKELP (0.800) configurations, but not by PQCLP (0.555). Its new-links test set, with 128 pairs, is the largest.

Other networks. On the static benchmarks, the QGCNLP gap was largest on PubMed, the largest of the five graphs, and smallest on WikiCS and PPI. The QLPCN wins occurred on the sparse networks CiteSeer, Cora and P2P-Gnutella, and its largest shortfalls on HuRI and AS routes, where L3 remained stronger than either branch.

6.5 Training Behaviour

Figure 10 shows the training loss of the PQCLP classifiers under the tabulated setting for one split of each network, and the diagnostic reproduces the seed-1 test AUC of the table runs exactly. Under SPSA, the loss fell by at most 0.007 for the VQC and at most 0.0007 for the QNN over 100 steps, and no weight moved by more than 0.021 radians from its initial value. The classifiers therefore stayed close to their random initialisation. Six of the thirty mean AUCs over variants

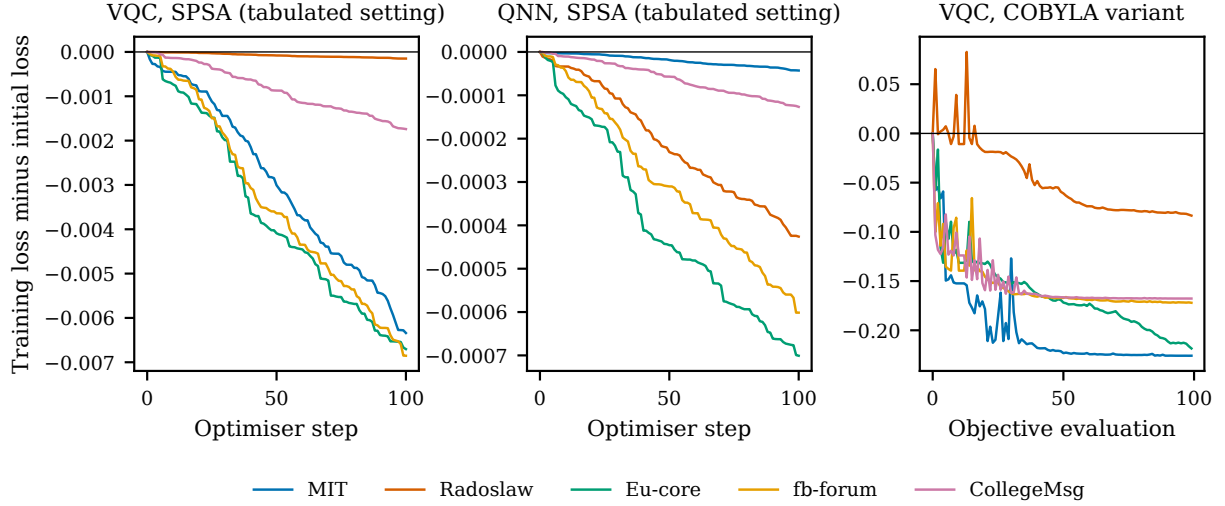


Figure 10: Training loss of the PQCLP classifiers relative to the initial loss, for the local feature family, four qubits and seed 1. SPSA uses the tabulated setting ($\eta = 0.001$, $c = 0.05$, 100 steps) and reproduces the seed-1 weights of the table runs exactly; COBYLA uses the same budget of objective evaluations. The vertical scales differ between panels.

and feature families lie within 0.01 of 0.5, and the QNN and VQC means agree to within 0.001 in eight of fifteen cells, even though the two variants minimise different losses. With the same budget of objective evaluations, COBYLA reduced the VQC loss by 0.08 to 0.23, moved the weights by 1.4 to 2.6 radians and raised the seed-1 test AUC to between 0.62 and 0.80. For QGCNLP, early stopping on validation AUC ended training after 181 and 190 epochs on average for the four- and eight-qubit models, compared with 88 for GCN. Per-epoch loss curves were not stored for QGCNLP or A2QTGN.

6.6 Protocol Controls

6.6.1 The repeat-edge control

Table 17: The repeat-edge control under the snapshot protocol. “Seen” is the fraction of positive pairs that are edges of some history snapshot (zero for negatives by construction). Method columns give the best quantum configuration, mean AUC against the observed edges. The last column is the size of the new-links test set.

Network	Seen	SeenBefore	PQKLP	PQCLP	PQKELP	New-link test pairs
MIT	98.1%	0.992	0.899	0.515	0.759	3
Radoslaw-email	86.8%	0.931	0.958	0.510	0.783	76
Eu-core	96.4%	0.982	0.984	0.679	0.809	16
fb-forum	85.6%	0.932	0.923	0.599	0.868	42
CollegeMsg	35.6%	0.677	0.847	0.555	0.800	128

On four of the five networks, between 85.6 and 98.1 per cent of positive pairs were already edges in the history, whereas no negative pair was (Table 17, Figure 11). SeenBefore, which uses only this fact, reached an AUC of 0.93 to 0.99 on those four networks. It exceeded the best PQCLP configuration on all five networks and the best PQKELP configuration on all networks except CollegeMsg. The best PQKLP configuration exceeded it on Radoslaw-email and CollegeMsg, and on Eu-core by 0.002, a margin too small to interpret. On such a protocol, high accuracy mainly shows that a pipeline retains the history of past edges, and differences between methods near the top of the scale say little about how new links form. Restricting the positives to new links removes

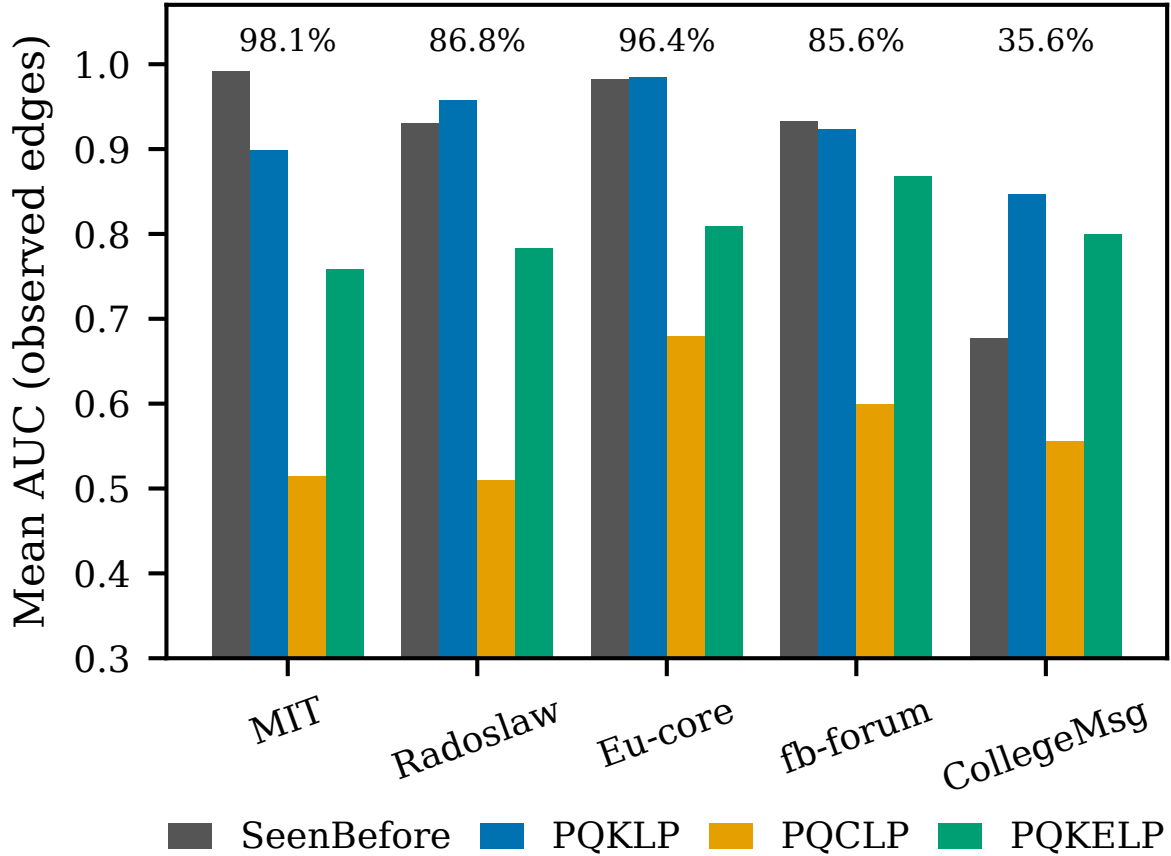


Figure 11: SeenBefore compared with the best quantum configuration of each snapshot-based method, as mean AUC against observed edges. Percentages give the share of positive pairs that are edges of some history snapshot.

this effect by construction, and under that target no quantum-favouring matched difference was significant (Section 6.2).

6.6.2 Relabelled targets

When the relabelling procedure was reproduced as the tutorial code performs it, the relabelled targets agreed with observed edges on a mean fraction of 0.41 to 0.53 of pairs per network for PQKLP and 0.50 overall for PQKELP (Figure 12). Individual splits ranged from 0.16 to 0.82 for PQKLP and from 0.24 to 0.76 for PQKELP. The relabelled targets are therefore close to unrelated to the observed links.

Against these targets, the projected quantum kernel won 29 of 30 matched comparisons for PQKLP, 27 of them significant, and all 30 for PQKELP, all significant (Table 14). The benefit was largest for the linear and naive Bayes classifiers (Table 15). The relabelling construction is designed to emphasise geometric differences between the quantum and classical feature representations. Strong performance on these constructed labels therefore does not, by itself, demonstrate better prediction of observed network links, and against observed edges the same circuits did not show a corresponding advantage in our experiments.

6.6.3 Compression-matched comparator

Table 18 decomposes the published matched difference of the two projected-kernel methods into the part contributed by the circuit and the part contributed by compression. Against the same ten

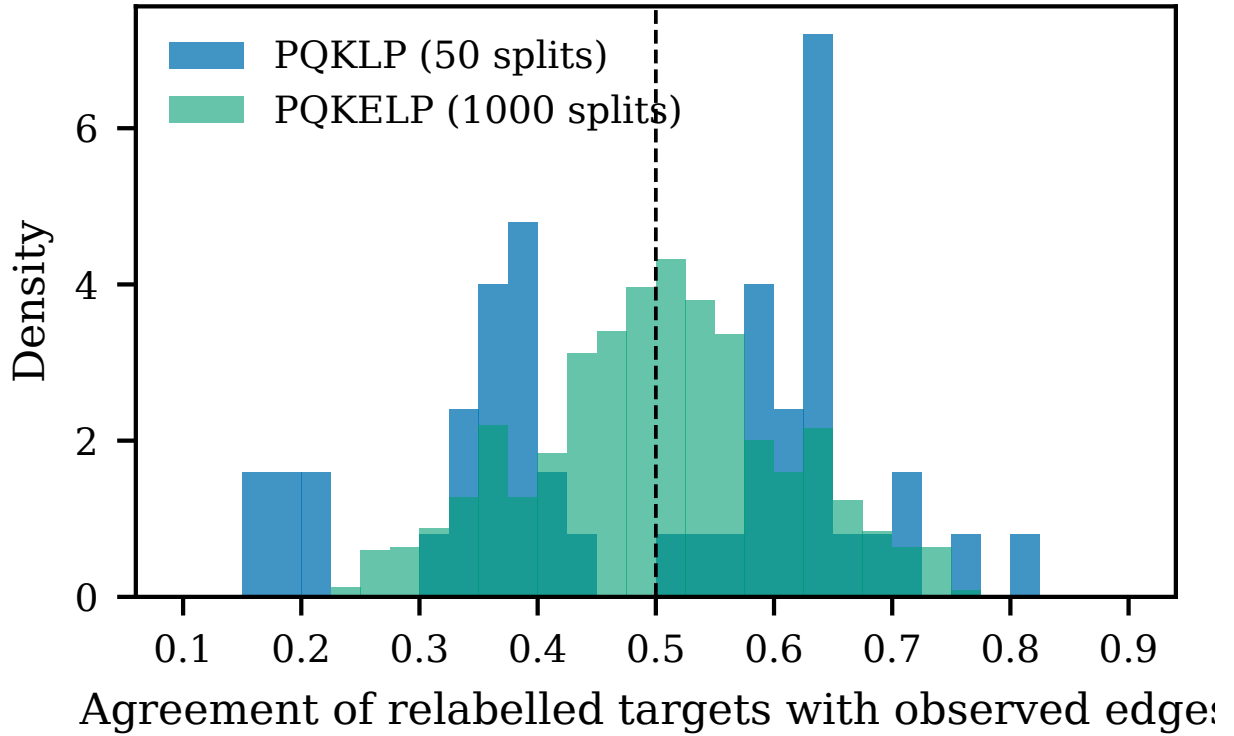


Figure 12: Agreement of the relabelled targets with observed edges for each split: 50 PQLP splits and 1000 PQLP splits (embeddings, training ratios and seeds). The dashed line marks the agreement expected from unrelated labels.

Table 18: Decomposition of the published matched difference for the two projected-kernel methods. Each cell of a family is one network and one classifier; the difference is tested by a paired Wilcoxon test over ten seeds, with Benjamini–Hochberg correction within the family of 30 cells and significance at $q < 0.05$. The circuit contrast compares the quantum arm with the same classifiers trained on the ten principal components the circuit receives; the compression contrast compares those components with the uncompressed features; the published contrast is the matched comparison as the papers frame it.

Method	Target	Contrast	Mean Δ AUC	Sig. higher	Sig. lower
PQLP	observed	circuit (Q – PCA-10)	-0.048	0	24
		compression (PCA-10 – full)	-0.010	0	21
		published (Q – full)	-0.058	0	26
PQLP	new links	circuit (Q – PCA-10)	-0.064	0	14
		compression (PCA-10 – full)	-0.048	1	7
		published (Q – full)	-0.112	0	16
PQLP	observed	circuit (Q – PCA-10)	-0.131	0	28
		compression (PCA-10 – full)	+0.031	11	13
		published (Q – full)	-0.100	1	22
PQLP	new links	circuit (Q – PCA-10)	-0.207	0	25
		compression (PCA-10 – full)	+0.022	5	4
		published (Q – full)	-0.187	0	16

principal components that the circuit receives, the circuit’s features were significantly worse in 24 of 30 comparisons for PQKLP (mean -0.048) and in 28 of 30 for PQKELP (mean -0.131) under the observed-edge target, and significantly better in none; their means were higher in 2 and 1 cells. Compression alone cost PQKLP 0.010 (21 of 30 significant in favour of the uncompressed features) and helped PQKELP by 0.031 (11 significant in favour of the ten components, 13 against). Under the new-links target the circuit’s deficit was 0.064 (14 of 30) and 0.207 (25 of 30), while compression cost PQKLP 0.048 and helped PQKELP by 0.022.

The published matched differences recomputed here, -0.058 with 26 significant for PQKLP and -0.100 with 22 for PQKELP, reproduce the counts of Table 14. The decomposition therefore attributes most of the published gap to the circuit rather than to the compression, and it removes the qualification that the observed-edge comparisons could not separate the two.

6.6.4 Message passing restricted to training edges

Table 19: QGCNLP and its classical graph comparators with message passing over all edges, as the paper specifies, and restricted to training edges. Mean AUC over three seeds with standard deviation; the change is the training-edge mean minus the all-edge mean.

Graph	Model	All edges	Training edges only	Change
CiteSeer	QGCNLP-8Q	0.883 ± 0.018	0.758 ± 0.011	-0.125
	GCN	0.988 ± 0.002	0.908 ± 0.006	-0.080
	GraphSAGE	0.986 ± 0.005	0.872 ± 0.012	-0.115
	GAT	0.993 ± 0.001	0.904 ± 0.009	-0.089
Cora	QGCNLP-8Q	0.855 ± 0.006	0.725 ± 0.015	-0.130
	GCN	0.982 ± 0.003	0.881 ± 0.008	-0.101
	GraphSAGE	0.968 ± 0.010	0.857 ± 0.011	-0.111
	GAT	0.990 ± 0.001	0.888 ± 0.001	-0.103
PPI	QGCNLP-8Q	0.830 ± 0.007	0.817 ± 0.013	-0.013
	GCN	0.916 ± 0.013	0.857 ± 0.013	-0.060
	GraphSAGE	0.870 ± 0.014	0.841 ± 0.012	-0.029
	GAT	0.915 ± 0.001	0.878 ± 0.002	-0.036
PubMed	QGCNLP-8Q	0.718 ± 0.032	$0.676 \pm <0.001$	-0.043
	GCN	0.965 ± 0.009	0.887 ± 0.010	-0.078
	GraphSAGE	0.951 ± 0.002	0.804 ± 0.005	-0.147
	GAT	0.989 ± 0.001	0.915 ± 0.009	-0.074
WikiCS	QGCNLP-8Q	0.884 ± 0.008	0.874 ± 0.006	-0.011
	GCN	0.939 ± 0.010	0.929 ± 0.016	-0.010
	GraphSAGE	0.959 ± 0.004	0.939 ± 0.003	-0.020
	GAT	0.960 ± 0.001	0.941 ± 0.006	-0.019

Rebuilding the propagation graph from training edges only lowered every model’s AUC on all five graphs (Table 19), by amounts that differ between graphs and between models on the same graph: 0.080–0.125 on CiteSeer, 0.101–0.130 on Cora, 0.043–0.147 on PubMed, 0.013–0.060 on PPI and 0.010–0.020 on WikiCS. QGCNLP-8Q lost the most on CiteSeer and Cora (0.125 and 0.130), so its gap to the best classical encoder widened there, from 0.111 to 0.150 and from 0.135 to 0.162; on PPI, PubMed and WikiCS it lost the least or nearly so and the gap narrowed slightly. QGCNLP-8Q remained last on all five graphs under both settings, but the leading classical encoder changed on CiteSeer (GAT to GCN) and on PPI (GCN to GAT): with all edges visible the classical encoders sat between 0.87 and 0.99, close enough to the ceiling that leakage decided their order. The inflation is therefore model- and graph-dependent, and affects the comparisons between models as well as the absolute values.

Table 20: A2QTGN under the Temporal Graph Benchmark’s official evaluation: each of the 3750 test events ranked against its 999 candidate negatives, on 25,000 evenly spaced events of tgbl-wiki with the stream’s edge features, five seeds and ten epochs. "Circuit runs on" is the measured fraction of observed events that triggered a re-encoding. EdgeBank and persistence are deterministic, so their spread over seeds is zero.

Model	MRR	sd	Range over seeds	AUC	Hits@10	Circuit runs on
Persistence (memorisation rule)	0.626		deterministic	0.753	0.695	–
EdgeBank (memorisation rule)	0.482		deterministic	0.673	0.697	–
A2QTGN (adaptive circuit)	0.032	0.010	0.019–0.042	0.457	0.060	88% of events
TGN (classical ablation)	0.031	0.005	0.023–0.037	0.443	0.063	–
A2QTGN, circuit frozen	0.030	0.007	0.023–0.038	0.448	0.060	24% of events
A2QTGN, circuit every event	0.028	0.007	0.021–0.038	0.453	0.058	100% of events

6.6.5 Official candidate negatives for A2QTGN

Under the Temporal Graph Benchmark’s own candidate lists, with the stream’s edge features and five seeds (Table 20), the adaptive model reached 0.032 ± 0.010 MRR, the same model with its circuit frozen 0.030 ± 0.007 , with the circuit on every event 0.028 ± 0.007 , and a classical ablation of the same architecture 0.031 ± 0.005 . These differ by less than the spread across seeds within any one of them, although the measured share of events that triggered a circuit ranged from 24% to 100%: the amount of quantum computation varied fourfold without changing the ranking. All four fell far below the two memorisation rules, EdgeBank at 0.482 and persistence at 0.626, and their AUC lies slightly below 0.5, so against a single random candidate they rank the true destination no better than chance. The two rules score sensibly on the same candidate lists, so the candidate sets and the ranking code behave as intended; the learned models do not learn to rank from the 17,500 training events of this subset. The paper’s ablation ordering, adaptive best and frozen worst, did not reproduce.

6.7 Sensitivity Analyses

Table 21: Sensitivity analyses on the five temporal networks, mean AUC against the observed edges. (a) PQCLP VQC, best feature family under the tabulated SPSA setting and under COBYLA with the same iteration budget. (b) PQKELP, best PQKE classifier per embedding at training ratio 0.8, and for the combined embedding at training ratios 0.5 and 0.9.

(a) PQCLP optimiser

Network	SPSA best	AUC	COBYLA best	AUC
MIT	VQC (global)	0.515	VQC (global)	0.648
Radoslaw-email	VQC (local)	0.510	VQC (local)	0.727
Eu-core	VQC (local)	0.679	VQC (local)	0.876
fb-forum	VQC (global)	0.599	VQC (global)	0.794
CollegeMsg	VQC (combined)	0.555	VQC (global)	0.794

(b) PQKELP embedding and training ratio

Network	Embedding (ratio 0.8)				Combined	
	N2V	DW	WL	Comb.	0.5	0.9
MIT	0.760	0.734	0.841	0.759	0.750	0.789
Radoslaw-email	0.736	0.696	0.804	0.783	0.767	0.791
Eu-core	0.828	0.820	0.831	0.809	0.798	0.822
fb-forum	0.847	0.858	0.856	0.868	0.845	0.874
CollegeMsg	0.801	0.819	0.813	0.800	0.781	0.804

Optimiser. Replacing SPSSA with COBYLA raised the best VQC configuration on every network, by 0.134 to 0.239 AUC (Table 21a). The VQC nevertheless remained below the classical twins on all five networks, by 0.194 AUC on average.

Test ratio. Across the test ratios of 0.1 to 0.5 that PQCLP varies, the combined-family QNN and VQC stayed between 0.509 and 0.530 in mean accuracy, the neural network between 0.905 and 0.912, the support vector machine between 0.865 and 0.870, and SeenBefore between 0.903 and 0.905.

Embedding and training ratio. A single embedding scored higher than the combined embedding on four of the five networks, all except fb-forum (Table 21b), although the combined embedding is the configuration that the PQKELP paper names. The best PQKE configuration gained 0.023 to 0.039 AUC between training ratios of 0.5 and 0.9, although not at every intermediate step (Figure 13).

Qubits and circuit pattern. QGCNLP-8Q exceeded QGCNLP-4Q on all five static graphs, by 0.007 to 0.045 AUC, at 23 times the time per epoch (Figure 13). Across the six circuit patterns of the QGCNLP ablation, mean AUC ranged from 0.803 to 0.838 (Table 22). The configuration used elsewhere, with RX, RY and RZ rotations, all-pairs entanglement and two blocks, was the best pattern and took 4.3 times as long to fit as the simplest circuit.

Table 22: QGCNLP circuit patterns (the grid of the paper’s Table 3), eight qubits, mean over the five static benchmarks and three seeds. The last row is the configuration used elsewhere in this study.

Trainable gates	Entanglement	Blocks	AUC	Accuracy	Hits@50	Fit (s)
RY	linear	1	0.803	0.756	0.887	902
RX, RY, RZ	linear	1	0.826	0.766	0.909	1405
RY	ring	2	0.834	0.770	0.867	1795
RX, RY	ring	2	0.825	0.769	0.871	2656
RX, RY, RZ	all pairs	1	0.814	0.763	0.892	2044
RX, RY, RZ	all pairs	2	0.838	0.777	0.875	3909

Update strategy. The A2QTGN update strategies differed by at most 0.013 on either metric in the single run (Table 13).

6.8 Quantum Resources and Runtime

Table 23 and Figure 14 give the resources of each circuit as executed. Without decomposing state preparation, the projected-kernel circuit is the largest, with 300 two-qubit gates and depth 87 on eleven qubits, and it has no trainable parameters; simulating it took 16.6 seconds per 2,000 pairs. The PQCLP circuits contain 18 and 42 CNOT gates. A PQCLP fit took 3.5 seconds on four qubits and 37.5 seconds on eight, compared with 0.45 to 0.59 seconds for its neural twin and at most 0.02 seconds for its support vector machine. The QGCNLP-8Q projection contains 112 CNOT gates, and one training epoch took 19.4 seconds, compared with 0.24 seconds for GCN, with four experiments sharing the machine. The QLPCN circuit has 13 to 15 qubits on the eight networks; its register and phase operators are dense or diagonal unitaries that were not decomposed into gates, and scoring all candidate pairs of a network took from 7 seconds (Cora) to 1,152 seconds (HuRI). In simulation, the A2QTGN encoder is a single 8-qubit state preparation, but decomposing it into elementary gates gives 515 CNOT gates and a depth of 1011. All runtimes were recorded in simulation, partly while other evaluation jobs were running, and indicate orders of magnitude only. None of the circuits was run on quantum hardware, so shot counts and device-level costs were not measured.

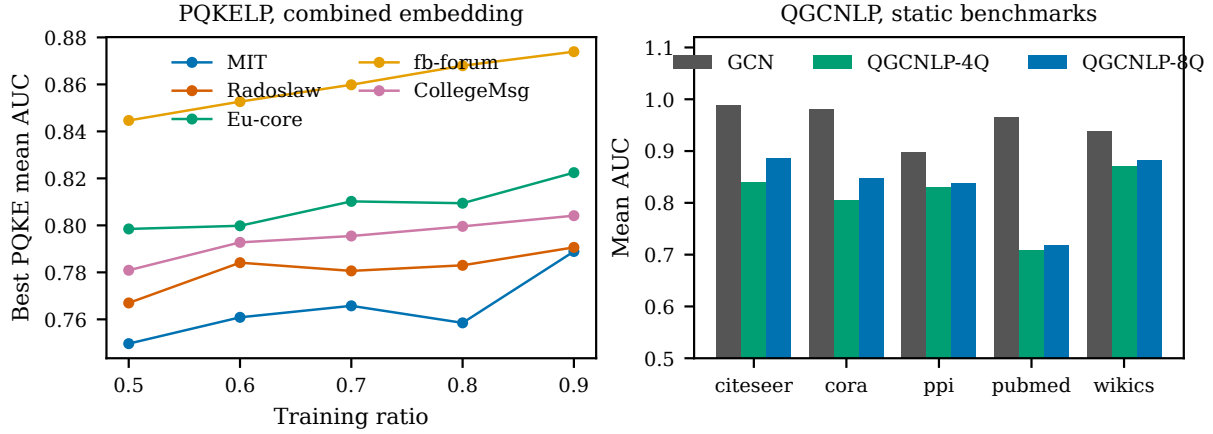


Figure 13: Left: best PQKE configuration of POKELP (combined embedding) across training ratios. Right: QGCNLP-4Q, QGCNLP-8Q and GCN on the five static benchmarks, mean AUC over five seeds.

Table 23: Quantum resources of every circuit executed, counted from the circuits as run (PennyLane tapes, without the start-state preparation). All simulations are exact state vectors without shots. For A2QTGN the amplitude embedding is one 8-qubit state preparation as simulated; parenthesised counts decompose it into R_Y , R_Z and CNOT gates. The PQK two-qubit gates are IsingXX, IsingYY and IsingZZ rotations. Runtimes are wall-clock on the machine of Section 5.6 and indicate order of magnitude only; a GCN epoch took 0.24 s under the same protocol; QGCNLP runtimes were recorded with four experiments sharing the machine. QLPCN gates are counted on a 34-node graph; on the eight networks the circuit has the qubits shown, its register and phase operators are dense or diagonal unitaries that are not decomposed, and the runtime is one scoring of all candidate pairs.

Circuit	Simulator	Qubits	Depth	1-qubit gates	2-qubit gates	Trainable	Measured	Recorded runtime
PQK (PQKLP, POKELP)	PennyLane	11	87	33	300	0	33: $\langle X \rangle, \langle Y \rangle, \langle Z \rangle$	16.6 s per 2,000 pairs
PQCLP, 4 qubits	PennyLane	4	27	34	18	12	$P(q_0=1)$	3.5 s per fit
PQCLP, 8 qubits	PennyLane	8	43	70	42	24	$P(q_0=1)$	37.5 s per fit
QGCNLP, 4 qubits	PennyLane	4	29	32	24	24	4 $\langle Z \rangle$	0.84 s per epoch
QGCNLP, 8 qubits	PennyLane	8	97	64	112	48	8 $\langle Z \rangle$	19.44 s per epoch
A2QTGN, 8 qubits	PennyLane	8	8 (1011)	0 (510)	7 (515)	0	8 $\langle Z \rangle$	497 s training
QLPCN	PennyLane	13–15	3	2	3 unitaries	0	probabilities	7–1,152 s per network

6.9 Statistical Analysis

The significance counts depend on which comparisons are pooled, and each count below states its scope. Under the observed-edge target, 1 of the 60 kernel-method comparisons was significant in favour of the quantum variant and 48 in favour of the classical counterpart. Including PQCLP, the corresponding counts over 90 observed-edge comparisons are 1 and 78. Over the 180 comparisons of all three methods under the observed-edge and new-links targets together, 1 difference was significant in favour of a quantum variant and 141 in favour of the classical counterpart. Under the relabelled target, 57 of the 60 kernel-method comparisons were significant in favour of the quantum variant and none in favour of the classical counterpart.

With ten paired seeds, the smallest attainable p -value is 0.002. This value was reached in every family, so the tests had enough power to detect effects that are consistent across seeds. Every matched comparison used ten seeds except one, POKELP with linear discriminant analysis on Radoslaw-email under the new-links target, which used nine because the classical fit did not converge on seed 10; that difference remained significant in favour of the classical counterpart ($q = 0.009$), and no count in Table 14 depends on it. Table 24 in Appendix A lists every comparison with its confidence interval, p -value and q -value. The QGCNLP results (three to five seeds) and

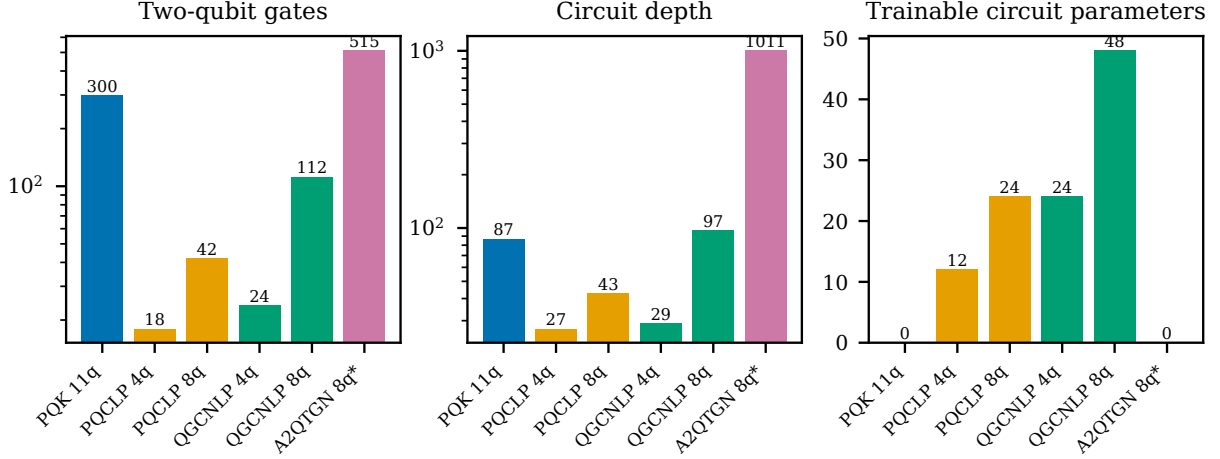


Figure 14: Two-qubit gate count, circuit depth (logarithmic scale) and number of trainable circuit parameters for every executed circuit except QLPCN, whose operators are unitaries not decomposed into gates (Table 23). The A2QTGN counts (marked *) decompose its amplitude embedding into R_Y , R_Z and CNOT gates.

the A2QTGN results (one run) are reported as means and were not tested.

7 Discussion

7.1 Why the Methods Performed as They Did

PQKLP and PQKELP. In both methods the circuit is a fixed nonlinear map applied to ten principal components, so its 33 outputs are a deterministic function of those ten numbers. The classical counterpart sees the uncompressed 20 or 1536 features. Against observed edges, the quantum arm scored lower in 28 of 30 comparisons for PQKLP and 25 of 30 for PQKELP. One possible explanation is that compression discards information that a fixed map cannot restore, but the experiment does not isolate this mechanism, because the observed-edge comparison changes compression and the circuit together. The significant PQKELP win on fb-forum is consistent with this reading: linear discriminant analysis on 1536 standardised features is a weak classifier on this network, and a ten-dimensional input may help it. Under the relabelled target the comparator receives the same ten components, which isolates the circuit, and there the circuit helped. The comparator of Section 6.6.3 separates the two effects, and it does not support that explanation: against the same ten components the circuit’s features were significantly worse in 24 of 30 comparisons for PQKLP and 28 of 30 for PQKELP and significantly better in none, whereas compression alone cost PQKLP 0.010 and *helped* PQKELP by 0.031. Most of the published gap is therefore contributed by the circuit rather than by the compression that precedes it. That PQKELP scored below PQKLP on all five networks, and that a single 512-dimensional embedding usually scored above the 1536-dimensional combined one, is also consistent with compression costing more as the input widens; it does not demonstrate that mechanism.

PQCLP. Under the tabulated setting, SPSA with a constant step of 0.001 moved no weight by more than 0.021 radians in 100 iterations, so both variants remained close to their random initialisation. This accounts for the near-chance accuracy in our experiments and for the similarity of the QNN and VQC means. With COBYLA the classifier trained, but it stayed well below classical twins that fit in a few hundredths of a second. Our experiments do not resolve whether the published values were obtained with SPSA as tabulated, with a different perturbation size or with a configuration closer to the COBYLA variant.

QGCNLP. In our implementation the circuit output is added to the convolution output, so each layer can pass the classical representation through largely unchanged. QGCNLP nevertheless scored below GCN on every static graph, whereas eight qubits improved on four on every graph. One possible explanation is that the quantum branch, as implemented, perturbs a representation that the convolution already provides, and that a wider projection perturbs it less. The runs do not test this explanation. Because held-out edges take part in message passing for all graph models, the absolute AUC values of that comparison are internal to our setup. Restricting propagation to training edges (Section 6.6.4) lowered every model on every graph, and unevenly: QGCNLP’s gap to the best classical encoder widened on CiteSeer and Cora and narrowed on the other three graphs, and the leading classical encoder changed on two graphs. Leakage of this kind therefore distorts comparisons between models, not only the absolute numbers, which is a reason to report the restricted variant alongside any full-edge result.

A2QTGN. With $\beta = 1$ and $\tau = 0.5$, the gate opens whenever a node’s input changes, so the adaptive arm differs from the always-re-encode arm mainly in how it blends old and new inputs. During evaluation, when the embeddings no longer change, it executed the circuit once per node, exactly as the encode-once arm did. Within this single-run experiment the differences among the arms were below 0.013, and the memorisation baselines reached more than twice their MRR. Under our sampling protocol those results reflect the protocol more than the encoder. The controlled run with the official candidate lists, edge features and five seeds (Section 6.6.5) sharpens the point rather than softening it: the adaptive, frozen, always-on and classical arms lie within the seed spread of one another even though the measured share of events triggering a circuit ranges from 24% to 100%, and all four rank far below a rule that predicts previously seen pairs. On this stream and training budget the quantum encoder is inert, and an adaptive schedule has no difference to exploit.

QLPCN. Its accuracy claim is modest and its complexity claim is analytic. Neither depends on protocol details that differ between papers, which may explain why both held up under reimplementations. Reimplementation did expose a numerical issue. At the shortest published walk times, the even-branch score of a pair of distinct nodes is typically below 10^{-17} , which double precision ranks partly by rounding and which no feasible number of circuit samples could estimate. Scores computed without cancellation brought QLPCN closer to the classical predictors. The published walk times therefore lie in a regime where sampled or double-precision scores and exact scores can rank pairs differently.

7.2 What the Quantum Models Appear to Learn

The projected-kernel features are a smooth nonlinear transformation of ten inputs. Under the relabelled target, linear and naive Bayes classifiers gained most from them, which is what a nonlinear feature expansion offers to classifiers that cannot form nonlinear decision boundaries themselves. The relabelled targets are built from the eigenstructure of these same features, so classifiers that receive them start with an advantage by construction. This pattern is consistent with a nonlinear feature expansion aligned with a constructed target. It does not show an advantage specific to quantum computation, and a classical random nonlinear map of the same width, relabelled in the same way, would test that distinction. Under SPSSA, the PQCLP classifiers learned little beyond their initialisation. Within the noise of our runs, the QGCNLP and A2QTGN encoders did not capture anything that their classical ablations missed.

7.3 Relation to Prior Literature

The reproduced orderings agreed with the published ones where the published effects are large, as with the weakness of PQCLP. They disagreed where the accuracy claims rest on comparisons made under protocols that the papers do not share, as the baseline audit of Section 3.7.1 would lead one to

expect. The relabelled datasets of Huang et al. [23] were designed to exhibit a separation between quantum and classical kernels, and when they serve as a link-prediction target they measure that separation rather than the formation of links. The strength of SeenBefore mirrors, for snapshot evaluation, the observation of Poursafaei et al. that memorisation baselines are competitive in streaming evaluation [18], and EdgeBank and Persistence showed the same effect in our A2QTGN run. We did not measure kernel concentration systematically, so these results neither support nor contradict the analysis of Thanasilp et al. [24] for these circuits.

7.4 Evidence from the New-Links Target

The new-links target is the most direct test of whether a method predicts links that have not occurred before, but its test sets are small and uneven: 3 pairs on MIT, 16 on Eu-core, 42 on fb-forum, 76 on Radoslaw-email and 128 on CollegeMsg. An AUC computed on 3 or 16 pairs is dominated by chance, so MIT and Eu-core provide weak evidence about new-link prediction. The conclusions drawn from this target rest mainly on Radoslaw-email, fb-forum and CollegeMsg, and larger or longer temporal networks would be needed to evaluate new-link prediction precisely.

7.5 Performance, Resources and Scalability

The executed circuits contain between 18 and 300 two-qubit gates, and none gave an accuracy gain against observed edges in our experiments. Exact simulation scales as 2^n in the number of qubits and linearly in the number of pairs, and it already took 16.6 seconds per 2,000 pairs for the eleven-qubit projected kernel. Scoring the $O(N^2)$ candidate pairs of a large network in this way would be impractical, and on hardware each pair would require 33 expectation values, each estimated from repeated shots. QLPCN’s advantage is asymptotic and assumes fault-tolerant Hamiltonian simulation. Our circuit is compiled from a dense eigendecomposition, whose cubic cost is classical, and nothing measured here tests the quantum speed-up. The A2QTGN gate is the only mechanism in the corpus that saves circuit executions, and it saves them only when node inputs stop changing.

7.6 Practical Use and Reproducibility

For practitioners, the results suggest a sequence of checks. The strength of classical baselines and of SeenBefore should be established on the protocol at hand, scoring should use observed edges (and new links where the test sets are large enough), and matched differences should be tested over several seeds. Competitive accuracy is not in itself evidence of a quantum advantage, particularly on a protocol in which high accuracy is easy to reach. Three reproducibility lessons emerged. Exact, seeded simulation ruled out shot noise as an explanation and made every run repeatable. A silent failure, a classical network returning a constant score, would have added apparent quantum wins had the scores not been checked for exact chance values. Finally, the choice of evaluation target, relabelled or observed, determined the sign of the reported effect.

8 Threats to Validity and Limitations

Reimplementation risk. A negative reproduction may reflect a limitation of the reimplementation rather than of the method. We reduced this risk by reproducing the published snapshot statistics exactly, validating the circuits against independent constructions, documenting every deviation (Table 10) and relying on internally controlled contrasts. We do not claim that any published value is incorrect. Our finding is that the published orderings did not carry over to this framework.

Implementation deviations. Several deviations limit specific claims. The scikit-learn neural classifiers limit absolute claims about the neural-network rows of PQKLP and PQKELP. The

shared rotation wall, the single-threaded embedding training and the SPSA perturbation size are our choices, and so are the residual branch and layer normalisation in QGCNLP.

Compression in the matched comparisons. Under the observed-edge and new-links targets, the classical counterparts of PQKLP, PQKELP and PQCLP use uncompressed features, whereas the quantum arms use a few principal components, so the published form of these comparisons measures compression and the quantum stage together. For the two projected-kernel methods this is now resolved by the compression-matched comparator (Section 6.6.3), which attributes most of the gap to the circuit; for PQCLP the comparator was not run, so its matched comparisons still confound the two. The comparator isolates the circuit from the compression, but not from the nonlinear expansion the circuit performs: a classical map of the same output width, which would test whether anything is specific to the quantum feature map, remains outstanding.

Reconstruction of the relabelled targets. The papers do not publish their relabelling code. We reproduced the tutorial construction that they cite, as that code behaves. If the papers departed from the tutorial, our relabelled values would differ from theirs, although the agreement of the targets with observed edges is a property of the construction itself.

Optimiser configuration. PQCLP tabulates SPSA, recommends COBYLA elsewhere and does not state the SPSA perturbation size. The reproduced behaviour depends strongly on this choice, which is the clearest case in which an unresolved configuration question could account for the distance between reproduced and published values.

Negative sampling. All snapshot results use uniformly sampled negatives drawn once per network. Other negative-sampling schemes, such as historical or degree-matched negatives, could change both absolute values and orderings.

Small new-links test sets. The new-links test sets contain 3 pairs on MIT and 16 on Eu-core. Results for these two networks under that target provide weak evidence and were not interpreted individually.

Full-edge message passing in QGCNLP. Our headline QGCNLP results, like the paper’s, pass messages over the full edge set, including held-out edges, for QGCNLP and its classical graph comparators alike, which inflates absolute performance. Section 6.6.4 measures that inflation on all five graphs: every model loses AUC, by 0.010 to 0.147, and unevenly enough to change which classical encoder ranks first on two graphs. The restricted variant uses three seeds, so its individual differences are descriptive rather than tested, and it was not run for the ogbl-ddi comparison.

Numerical precision of QLPCN scores. At the shortest walk times of the paper’s Table II, double precision cannot rank many QLPCN pairs reliably, in the executed circuit and in an eigen-decomposition alike. We report cancellation-free reference scores beside the circuit results. The reference was checked against 60-digit arithmetic on a small graph but not on the full networks.

A2QTGN protocol. The reproduction run of Table 13 uses one seed, a historical negative sampler instead of the official TGB candidate lists, and a local copy of the stream without edge features, so its absolute MRR values are not comparable with the published ones and its orderings are descriptive. The controlled run of Section 6.6.5 removes the first three of those limits, using the official candidate lists, the stream’s edge features and five seeds. Both runs remain on 25,000 evenly spaced events of tgbl-wiki, one of the paper’s five streams, with $\beta = 1$ and $\tau = 0.5$ because we did not locate the published values; the learned models are weak on that training budget, so the finding that the circuit’s schedule does not matter is established only in a regime where no learned model is competitive.

Number of runs and statistical power. The snapshot methods were run over ten seeds that redraw the split but not the candidate pairs, and one PQKELP new-links comparison has nine seeds because a classical fit failed to converge. QGCNLP was run over three to five seeds, A2QTGN once, and the QLPCN seeds redraw only the evaluation half of a single hold-out, whereas the paper uses ten-fold cross-validation. Significance tests cover the snapshot methods only, and with ten pairs the Wilcoxon test detects only effects that are consistent across nearly all seeds.

Exact simulation without hardware noise. Every result was obtained by exact, noise-free simulation. Device noise, finite shots and hardware connectivity were not modelled, and the hardware runs reported in the PQCLP and A2QTGN papers were not reproduced. Their effect on the observed gaps is unknown.

Scale and number of qubits. The temporal networks have at most 1899 nodes, and the circuits use at most eleven qubits, as the papers specify. Whether the observed behaviour persists on larger dynamic networks, or with wider encodings that compress less, was not tested.

Corpus and independence. Four of the five dynamic methods come from overlapping author teams, and one author of this study co-authored PQKLP (Section 3.6). The conclusions apply to this corpus rather than to quantum link prediction in general, and static-network quantum link predictors other than QLPCN were not reproduced.

Runtime measurement and software environment. Runtimes were recorded while other evaluation jobs were running and indicate orders of magnitude only. Package versions were recorded for all runs except the A2QTGN run (Section 5.6).

9 Conclusion and Future Work

This article examined whether the accuracy gains reported for quantum link prediction in dynamic networks persist when the published methods are reimplemented and evaluated under one protocol. We reimplemented five methods published between 2022 and 2026 and a static quantum-walk reference in a single framework, with exact simulation, shared data pipelines and matched classical counterparts, and we tested every matched comparison of the snapshot-based methods over ten seeds.

RQ1. Against observed edges, the quantum variants of the two projected-kernel methods had the higher mean AUC in 7 of 60 matched comparisons, and PQCLP in none of 30. Across these 90 comparisons, 1 difference was significant in favour of a quantum variant and 78 were significant in favour of the classical counterpart. For these methods the classical counterparts use uncompressed features, so the comparisons evaluate the published quantum pipelines, compression included, rather than the circuits in isolation. A comparator that gives the same classifiers the ten principal components the circuit receives separates the two for the projected-kernel methods (Section 6.6.3): against those inputs the circuit’s features were significantly worse in 24 of 30 comparisons for PQKLP and 28 of 30 for PQKELP and significantly better in none, while compression alone cost PQKLP 0.010 and helped PQKELP by 0.031. Most of the published gap is contributed by the circuit.

RQ2. No snapshot-based method exceeded the strongest classical configuration on any of the five temporal networks. QGCNLP scored below its classical comparators on every static and temporal network tested except ogbl-ddi, where it led GraphSAGE by 0.044 over three seeds, too few for a significance test. QLPCN reached approximate parity with classical path-based predictors, which is the claim that reproduced most closely; computed without double-precision cancellation, its scores came closer still.

RQ3. The published gains depend on the evaluation target. Against relabelled targets, which agree with observed edges on about half of all pairs, the same projected kernel won 59 of 60 comparisons. A parameter-free rule that predicts previously observed pairs reached an AUC of 0.93 to 0.99 on four of the five networks. Under the new-links target no quantum-favouring difference was significant, although the test sets for MIT and Eu-core are too small to be informative.

RQ4. The choice of optimiser moved PQCLP from near chance to a trained but still weaker classifier. More qubits improved QGCNLP without closing its gap to GCN. The executed circuits used up to 300 two-qubit gates, or 515 once the A2QTGN amplitude embedding is decomposed, without a corresponding accuracy gain against observed edges in our experiments.

Under this common protocol, the evidence does not establish a quantum accuracy advantage for link prediction in dynamic networks. These experiments also do not establish the absence of such an advantage, and we do not claim that they do. The conclusion is limited by exact simulation, small networks, deviations in several implementations and the small number of runs for QGCNLP and A2QTGN (Section 8).

The results point to several next steps. The compression-matched comparator reported here separates compression from the circuit for the two projected-kernel methods; the same comparator for PQCLP, and a classical random nonlinear map of the same output width, relabelled in the same way, would complete the attribution by testing whether anything is specific to the quantum feature map. Evaluations would benefit from reporting observed-edge results alongside any constructed target, from including the repeat-edge control, and from larger or longer temporal networks that give adequate new-links test sets. The A2QTGN re-evaluation with official candidate lists, edge features and five seeds, and the QGCNLP runs with message passing restricted to training edges, are reported in Sections 6.6.5 and 6.6.4; what remains for A2QTGN is a training budget and the remaining four streams on which its learned models are competitive at all. Noise-aware simulation, runs on quantum hardware with error mitigation, and an accounting of shots and circuit executions per prediction are needed before resource claims can be assessed. For QLPCN, walk times should be reported together with the number of samples needed to resolve the resulting scores. Finally, quantum graph models should be compared with current temporal graph networks, and the static quantum-walk methods outside this corpus remain candidates for reproduction.

Acknowledgements

Declarations

- **Funding.**
- **Competing interests.** S. Mishra is a co-author of PQKLP [27], one of the methods reproduced here, and of the survey by Singh et al. [32] discussed in Section 3.4. Section 3.6 discusses how this relationship bears on the independence of the evidence.
- **Author contributions.** A. K. Thakur carried out the reimplementations, the fidelity audit, the experiments and the analysis, and wrote the first draft of the manuscript. S. Mishra supervised the work: he conceived and directed the study, guided the experimental design and the reproduction protocol, reviewed the implementation and the results, and revised the manuscript. A. S. Parihar contributed to the study design, reviewed the methodology and the results, and revised the manuscript. All authors read and approved the final manuscript.
- **Data availability.** All networks are publicly available from the sources cited in Table 6.
- **Code availability.**
- **Ethics approval and consent to participate.**
- **Consent for publication.**

Appendix A All Matched Comparisons

Table 24 lists every matched comparison summarised in Table 14.

Table 24: All matched comparisons with paired statistics (ten seeds, except PQKLP with LDA on Radoslaw-email under the new-links target, which uses nine because the classical fit did not converge on seed 10). CI is the t -based 95% interval of the mean paired difference; p is the two-sided Wilcoxon signed-rank test; q is Benjamini–Hochberg within each block. The last column counts seeds on which the classical counterpart returned a constant score (AUC exactly 0.5).

Classifier	Network	Quantum	Classical	Δ	95% CI	p	q	Const.
<i>PQKLP, observed edges</i>								
NN	MIT	0.873	0.995	−0.122	[−0.138, −0.105]	0.002	0.003	0
NN	Radoslaw-email	0.925	0.977	−0.052	[−0.060, −0.044]	0.002	0.003	0
NN	Eu-core	0.974	0.994	−0.020	[−0.025, −0.016]	0.002	0.003	0
NN	fb-forum	0.901	0.948	−0.048	[−0.060, −0.035]	0.002	0.003	0
NN	CollegeMsg	0.838	0.862	−0.024	[−0.036, −0.012]	0.004	0.005	0
XGB	MIT	0.847	0.996	−0.148	[−0.169, −0.128]	0.002	0.003	0
XGB	Radoslaw-email	0.918	0.975	−0.057	[−0.064, −0.050]	0.002	0.003	0
XGB	Eu-core	0.961	0.989	−0.028	[−0.035, −0.021]	0.002	0.003	0
XGB	fb-forum	0.900	0.957	−0.057	[−0.068, −0.047]	0.002	0.003	0
XGB	CollegeMsg	0.838	0.873	−0.034	[−0.056, −0.013]	0.010	0.011	0
LR	MIT	0.856	0.986	−0.130	[−0.151, −0.109]	0.002	0.003	0
LR	Radoslaw-email	0.849	0.950	−0.102	[−0.121, −0.083]	0.002	0.003	0
LR	Eu-core	0.955	0.977	−0.022	[−0.029, −0.014]	0.004	0.005	0
LR	fb-forum	0.847	0.843	+0.004	[−0.016, +0.025]	0.695	0.719	0
LR	CollegeMsg	0.822	0.830	−0.008	[−0.024, +0.009]	0.275	0.306	0
RFC	MIT	0.899	0.998	−0.099	[−0.111, −0.088]	0.002	0.003	0
RFC	Radoslaw-email	0.958	0.982	−0.024	[−0.028, −0.020]	0.002	0.003	0
RFC	Eu-core	0.984	0.996	−0.012	[−0.016, −0.009]	0.002	0.003	0
RFC	fb-forum	0.923	0.962	−0.038	[−0.047, −0.029]	0.002	0.003	0
RFC	CollegeMsg	0.847	0.873	−0.026	[−0.041, −0.010]	0.006	0.007	0
LDA	MIT	0.866	0.986	−0.120	[−0.137, −0.103]	0.002	0.003	0
LDA	Radoslaw-email	0.852	0.950	−0.098	[−0.114, −0.081]	0.002	0.003	0
LDA	Eu-core	0.955	0.971	−0.016	[−0.023, −0.008]	0.004	0.005	0
LDA	fb-forum	0.851	0.841	+0.010	[−0.011, +0.031]	0.322	0.345	0
LDA	CollegeMsg	0.822	0.826	−0.004	[−0.022, +0.013]	0.846	0.846	0
GNB	MIT	0.818	0.987	−0.169	[−0.192, −0.145]	0.002	0.003	0
GNB	Radoslaw-email	0.806	0.940	−0.135	[−0.153, −0.117]	0.002	0.003	0
GNB	Eu-core	0.936	0.979	−0.043	[−0.049, −0.037]	0.002	0.003	0
GNB	fb-forum	0.812	0.866	−0.054	[−0.072, −0.037]	0.002	0.003	0
GNB	CollegeMsg	0.764	0.826	−0.062	[−0.094, −0.030]	0.002	0.003	0
<i>PQKLP, new links only</i>								
NN	MIT	0.750	0.900	−0.150	[−0.391, +0.091]	0.500	0.536	2
NN	Radoslaw-email	0.789	0.884	−0.095	[−0.134, −0.057]	0.002	0.005	0
NN	Eu-core	0.766	0.920	−0.155	[−0.217, −0.092]	0.002	0.005	0
NN	fb-forum	0.754	0.795	−0.041	[−0.074, −0.008]	0.037	0.065	0
NN	CollegeMsg	0.781	0.804	−0.023	[−0.034, −0.012]	0.004	0.009	0
XGB	MIT	0.475	0.800	−0.325	[−0.618, −0.032]	0.055	0.086	0
XGB	Radoslaw-email	0.761	0.867	−0.106	[−0.142, −0.070]	0.002	0.005	0
XGB	Eu-core	0.741	0.876	−0.134	[−0.192, −0.077]	0.002	0.005	0
XGB	fb-forum	0.708	0.738	−0.030	[−0.073, +0.012]	0.160	0.229	0
XGB	CollegeMsg	0.769	0.824	−0.055	[−0.077, −0.033]	0.002	0.005	0
LR	MIT	0.750	0.800	−0.050	[−0.363, +0.263]	1.000	1.000	2
LR	Radoslaw-email	0.776	0.876	−0.099	[−0.141, −0.058]	0.002	0.005	0
LR	Eu-core	0.762	0.905	−0.142	[−0.197, −0.088]	0.002	0.005	0
LR	fb-forum	0.739	0.776	−0.037	[−0.093, +0.018]	0.275	0.344	0
LR	CollegeMsg	0.773	0.792	−0.019	[−0.034, −0.003]	0.018	0.033	0
RFC	MIT	0.600	1.000	−0.400	[−0.729, −0.071]	0.062	0.094	0
RFC	Radoslaw-email	0.809	0.878	−0.069	[−0.107, −0.031]	0.004	0.009	0
RFC	Eu-core	0.810	0.934	−0.124	[−0.167, −0.081]	0.002	0.005	0
RFC	fb-forum	0.748	0.764	−0.017	[−0.058, +0.025]	0.449	0.499	0
RFC	CollegeMsg	0.768	0.800	−0.032	[−0.049, −0.014]	0.010	0.021	0
LDA	MIT	0.650	0.875	−0.225	[−0.587, +0.137]	0.250	0.326	2
LDA	Radoslaw-email	0.787	0.884	−0.098	[−0.136, −0.060]	0.002	0.005	0
LDA	Eu-core	0.677	0.897	−0.220	[−0.282, −0.159]	0.002	0.005	0
LDA	fb-forum	0.713	0.738	−0.025	[−0.086, +0.035]	0.557	0.576	0
LDA	CollegeMsg	0.759	0.773	−0.013	[−0.033, +0.007]	0.232	0.317	0
GNB	MIT	0.600	1.000	−0.400	[−0.641, −0.159]	0.016	0.031	0
GNB	Radoslaw-email	0.734	0.879	−0.145	[−0.184, −0.107]	0.002	0.005	0
GNB	Eu-core	0.802	0.889	−0.087	[−0.179, +0.005]	0.051	0.085	0
GNB	fb-forum	0.757	0.778	−0.021	[−0.067, +0.025]	0.287	0.345	0

Table 24: All matched comparisons with paired statistics (ten seeds, except PQKELP with LDA on Radoslaw-email under the new-links target, which uses nine because the classical fit did not converge on seed 10). CI is the t -based 95% interval of the mean paired difference; p is the two-sided Wilcoxon signed-rank test; q is Benjamini–Hochberg within each block. The last column counts seeds on which the classical counterpart returned a constant score (AUC exactly 0.5).

Classifier	Network	Quantum	Classical	Δ	95% CI	p	q	Const.
GNB	CollegeMsg	0.759	0.778	−0.019	[−0.051, +0.013]	0.322	0.372	0
<i>PQKELP, relabelled targets</i>								
NN	MIT	0.938	0.826	+0.112	[+0.047, +0.177]	0.002	0.002	0
NN	Radoslaw-email	0.945	0.891	+0.053	[+0.030, +0.077]	0.002	0.002	0
NN	Eu-core	0.946	0.925	+0.021	[+0.012, +0.030]	0.002	0.002	0
NN	fb-forum	0.933	0.899	+0.034	[+0.021, +0.047]	0.002	0.002	0
NN	CollegeMsg	0.948	0.940	+0.007	[−0.001, +0.015]	0.084	0.090	0
XGB	MIT	0.908	0.854	+0.054	[+0.028, +0.081]	0.002	0.002	0
XGB	Radoslaw-email	0.932	0.900	+0.032	[+0.024, +0.040]	0.002	0.002	0
XGB	Eu-core	0.937	0.923	+0.013	[+0.007, +0.020]	0.002	0.002	0
XGB	fb-forum	0.920	0.903	+0.017	[+0.003, +0.031]	0.020	0.022	0
XGB	CollegeMsg	0.946	0.940	+0.006	[−0.001, +0.013]	0.105	0.109	0
LR	MIT	0.946	0.704	+0.242	[+0.198, +0.286]	0.002	0.002	0
LR	Radoslaw-email	0.945	0.651	+0.294	[+0.271, +0.317]	0.002	0.002	0
LR	Eu-core	0.947	0.891	+0.056	[+0.035, +0.077]	0.002	0.002	0
LR	fb-forum	0.932	0.850	+0.082	[+0.060, +0.103]	0.002	0.002	0
LR	CollegeMsg	0.946	0.899	+0.047	[+0.022, +0.071]	0.004	0.005	0
RFC	MIT	0.938	0.890	+0.048	[+0.017, +0.079]	0.002	0.002	0
RFC	Radoslaw-email	0.943	0.921	+0.021	[+0.013, +0.030]	0.002	0.002	0
RFC	Eu-core	0.948	0.936	+0.011	[+0.006, +0.017]	0.002	0.002	0
RFC	fb-forum	0.929	0.920	+0.009	[+0.005, +0.013]	0.002	0.002	0
RFC	CollegeMsg	0.949	0.950	−0.0002	[−0.005, +0.005]	0.922	0.922	0
LDA	MIT	0.928	0.704	+0.224	[+0.181, +0.267]	0.002	0.002	0
LDA	Radoslaw-email	0.940	0.651	+0.290	[+0.266, +0.313]	0.002	0.002	0
LDA	Eu-core	0.942	0.889	+0.053	[+0.033, +0.072]	0.002	0.002	0
LDA	fb-forum	0.925	0.833	+0.092	[+0.072, +0.111]	0.002	0.002	0
LDA	CollegeMsg	0.946	0.894	+0.052	[+0.029, +0.076]	0.002	0.002	0
GNB	MIT	0.934	0.781	+0.152	[+0.101, +0.204]	0.002	0.002	0
GNB	Radoslaw-email	0.932	0.798	+0.134	[+0.098, +0.170]	0.002	0.002	0
GNB	Eu-core	0.930	0.894	+0.036	[+0.024, +0.049]	0.002	0.002	0
GNB	fb-forum	0.912	0.874	+0.038	[+0.026, +0.050]	0.002	0.002	0
GNB	CollegeMsg	0.941	0.909	+0.032	[+0.015, +0.049]	0.010	0.011	0
<i>PQKELP, observed edges</i>								
NN	MIT	0.725	0.943	−0.218	[−0.285, −0.151]	0.002	0.003	0
NN	Radoslaw-email	0.702	0.639	+0.062	[−0.069, +0.193]	0.275	0.318	3
NN	Eu-core	0.752	0.792	−0.040	[−0.181, +0.101]	0.846	0.906	1
NN	fb-forum	0.776	0.865	−0.089	[−0.153, −0.024]	0.014	0.018	0
NN	CollegeMsg	0.744	0.708	+0.036	[−0.012, +0.084]	0.131	0.164	0
XGB	MIT	0.750	0.992	−0.242	[−0.275, −0.210]	0.002	0.003	0
XGB	Radoslaw-email	0.780	0.962	−0.181	[−0.193, −0.169]	0.002	0.003	0
XGB	Eu-core	0.805	0.978	−0.173	[−0.188, −0.159]	0.002	0.003	0
XGB	fb-forum	0.861	0.935	−0.073	[−0.083, −0.063]	0.002	0.003	0
XGB	CollegeMsg	0.791	0.881	−0.090	[−0.117, −0.064]	0.002	0.003	0
LR	MIT	0.757	0.996	−0.239	[−0.274, −0.205]	0.002	0.003	0
LR	Radoslaw-email	0.768	0.951	−0.184	[−0.212, −0.156]	0.002	0.003	0
LR	Eu-core	0.765	0.976	−0.210	[−0.224, −0.196]	0.002	0.003	0
LR	fb-forum	0.841	0.915	−0.074	[−0.090, −0.058]	0.002	0.003	0
LR	CollegeMsg	0.773	0.823	−0.051	[−0.080, −0.021]	0.010	0.013	0
RFC	MIT	0.757	0.989	−0.232	[−0.265, −0.198]	0.002	0.003	0
RFC	Radoslaw-email	0.783	0.960	−0.177	[−0.198, −0.155]	0.002	0.003	0
RFC	Eu-core	0.809	0.975	−0.166	[−0.177, −0.154]	0.002	0.003	0
RFC	fb-forum	0.868	0.927	−0.059	[−0.068, −0.050]	0.002	0.003	0
RFC	CollegeMsg	0.800	0.868	−0.068	[−0.095, −0.041]	0.002	0.003	0
LDA	MIT	0.759	0.972	−0.213	[−0.247, −0.179]	0.002	0.003	0
LDA	Radoslaw-email	0.768	0.875	−0.108	[−0.136, −0.079]	0.002	0.003	0
LDA	Eu-core	0.765	0.805	−0.040	[−0.060, −0.020]	0.004	0.006	0
LDA	fb-forum	0.841	0.731	+0.110	[+0.087, +0.133]	0.002	0.003	0
LDA	CollegeMsg	0.771	0.773	−0.002	[−0.034, +0.030]	1.000	1.000	0
GNB	MIT	0.740	0.917	−0.177	[−0.219, −0.135]	0.002	0.003	0
GNB	Radoslaw-email	0.734	0.756	−0.022	[−0.051, +0.007]	0.232	0.279	0
GNB	Eu-core	0.736	0.825	−0.089	[−0.113, −0.065]	0.002	0.003	0
GNB	fb-forum	0.827	0.823	+0.004	[−0.015, +0.023]	0.492	0.547	0
GNB	CollegeMsg	0.758	0.757	+0.001	[−0.034, +0.036]	1.000	1.000	0
<i>PQKELP [min-max classical inputs], observed edges</i>								
NN	MIT	0.725	0.824	−0.099	[−0.256, +0.059]	0.131	0.157	2
NN	Radoslaw-email	0.702	0.637	+0.065	[−0.089, +0.218]	0.492	0.527	7

Table 24: All matched comparisons with paired statistics (ten seeds, except PQKELP with LDA on Radoslaw-email under the new-links target, which uses nine because the classical fit did not converge on seed 10). CI is the t -based 95% interval of the mean paired difference; p is the two-sided Wilcoxon signed-rank test; q is Benjamini-Hochberg within each block. The last column counts seeds on which the classical counterpart returned a constant score (AUC exactly 0.5).

Classifier	Network	Quantum	Classical	Δ	95% CI	p	q	Const.
NN	Eu-core	0.752	0.546	+0.206	[+0.091, +0.322]	0.027	0.034	9
NN	fb-forum	0.776	0.500	+0.276	[+0.249, +0.304]	0.002	0.003	10
NN	CollegeMsg	0.744	0.500	+0.244	[+0.212, +0.275]	0.002	0.003	10
XGB	MIT	0.750	0.992	-0.242	[-0.275, -0.210]	0.002	0.003	0
XGB	Radoslaw-email	0.780	0.961	-0.181	[-0.193, -0.168]	0.002	0.003	0
XGB	Eu-core	0.805	0.978	-0.173	[-0.188, -0.159]	0.002	0.003	0
XGB	fb-forum	0.861	0.935	-0.073	[-0.083, -0.063]	0.002	0.003	0
XGB	CollegeMsg	0.791	0.881	-0.090	[-0.117, -0.064]	0.002	0.003	0
LR	MIT	0.757	0.996	-0.239	[-0.272, -0.205]	0.002	0.003	0
LR	Radoslaw-email	0.768	0.963	-0.195	[-0.224, -0.167]	0.002	0.003	0
LR	Eu-core	0.765	0.975	-0.209	[-0.224, -0.195]	0.002	0.003	0
LR	fb-forum	0.841	0.935	-0.094	[-0.108, -0.080]	0.002	0.003	0
LR	CollegeMsg	0.773	0.851	-0.078	[-0.112, -0.044]	0.002	0.003	0
RFC	MIT	0.757	0.989	-0.232	[-0.265, -0.198]	0.002	0.003	0
RFC	Radoslaw-email	0.783	0.960	-0.177	[-0.198, -0.155]	0.002	0.003	0
RFC	Eu-core	0.809	0.975	-0.166	[-0.177, -0.154]	0.002	0.003	0
RFC	fb-forum	0.868	0.927	-0.059	[-0.068, -0.050]	0.002	0.003	0
RFC	CollegeMsg	0.800	0.868	-0.068	[-0.095, -0.041]	0.002	0.003	0
LDA	MIT	0.759	0.972	-0.213	[-0.247, -0.179]	0.002	0.003	0
LDA	Radoslaw-email	0.768	0.875	-0.108	[-0.136, -0.079]	0.002	0.003	0
LDA	Eu-core	0.765	0.805	-0.040	[-0.060, -0.020]	0.004	0.005	0
LDA	fb-forum	0.841	0.731	+0.110	[+0.087, +0.133]	0.002	0.003	0
LDA	CollegeMsg	0.771	0.773	-0.002	[-0.034, +0.030]	1.000	1.000	0
GNB	MIT	0.740	0.917	-0.177	[-0.219, -0.135]	0.002	0.003	0
GNB	Radoslaw-email	0.734	0.757	-0.022	[-0.051, +0.007]	0.232	0.268	0
GNB	Eu-core	0.736	0.825	-0.089	[-0.113, -0.065]	0.002	0.003	0
GNB	fb-forum	0.827	0.823	+0.004	[-0.015, +0.023]	0.492	0.527	0
GNB	CollegeMsg	0.758	0.757	+0.001	[-0.034, +0.036]	1.000	1.000	0
<i>PQKELP, new links only</i>								
NN	MIT	0.325	0.775	-0.450	[-0.740, -0.160]	0.016	0.028	1
NN	Radoslaw-email	0.646	0.780	-0.134	[-0.241, -0.027]	0.037	0.056	1
NN	Eu-core	0.508	0.814	-0.306	[-0.416, -0.197]	0.002	0.005	0
NN	fb-forum	0.602	0.633	-0.031	[-0.128, +0.066]	0.557	0.618	0
NN	CollegeMsg	0.673	0.570	+0.103	[+0.001, +0.206]	0.105	0.132	4
XGB	MIT	0.525	0.675	-0.150	[-0.573, +0.273]	0.457	0.527	3
XGB	Radoslaw-email	0.682	0.894	-0.212	[-0.283, -0.142]	0.002	0.005	0
XGB	Eu-core	0.428	0.844	-0.416	[-0.543, -0.289]	0.002	0.005	0
XGB	fb-forum	0.662	0.790	-0.128	[-0.180, -0.076]	0.002	0.005	0
XGB	CollegeMsg	0.721	0.822	-0.101	[-0.131, -0.072]	0.002	0.005	0
LR	MIT	0.500	0.950	-0.450	[-0.806, -0.094]	0.062	0.085	1
LR	Radoslaw-email	0.677	0.878	-0.200	[-0.267, -0.134]	0.002	0.005	0
LR	Eu-core	0.503	0.894	-0.391	[-0.515, -0.266]	0.002	0.005	0
LR	fb-forum	0.652	0.674	-0.022	[-0.119, +0.076]	0.770	0.824	0
LR	CollegeMsg	0.728	0.763	-0.035	[-0.061, -0.010]	0.010	0.018	0
RFC	MIT	0.500	0.950	-0.450	[-0.806, -0.094]	0.062	0.085	1
RFC	Radoslaw-email	0.712	0.916	-0.204	[-0.261, -0.146]	0.002	0.005	0
RFC	Eu-core	0.499	0.860	-0.361	[-0.474, -0.248]	0.002	0.005	0
RFC	fb-forum	0.716	0.813	-0.098	[-0.149, -0.046]	0.004	0.009	0
RFC	CollegeMsg	0.731	0.809	-0.079	[-0.097, -0.060]	0.002	0.005	0
LDA	MIT	0.500	0.900	-0.400	[-0.682, -0.118]	0.031	0.052	2
LDA	Radoslaw-email	0.672	0.853	-0.181	[-0.252, -0.110]	0.004	0.009	0
LDA	Eu-core	0.495	0.611	-0.116	[-0.270, +0.039]	0.125	0.150	0
LDA	fb-forum	0.644	0.639	+0.005	[-0.090, +0.101]	0.943	0.976	0
LDA	CollegeMsg	0.731	0.736	-0.005	[-0.037, +0.026]	1.000	1.000	0
GNB	MIT	0.550	0.825	-0.275	[-0.548, -0.002]	0.078	0.102	1
GNB	Radoslaw-email	0.688	0.755	-0.067	[-0.121, -0.013]	0.037	0.056	0
GNB	Eu-core	0.491	0.766	-0.275	[-0.424, -0.126]	0.006	0.013	0
GNB	fb-forum	0.698	0.772	-0.074	[-0.116, -0.032]	0.010	0.018	0
GNB	CollegeMsg	0.692	0.787	-0.094	[-0.131, -0.058]	0.002	0.005	0
<i>PQKELP [min-max classical inputs], new links only</i>								
NN	MIT	0.325	0.850	-0.525	[-0.867, -0.183]	0.020	0.034	2
NN	Radoslaw-email	0.646	0.565	+0.081	[-0.017, +0.179]	0.131	0.157	8
NN	Eu-core	0.508	0.738	-0.230	[-0.432, -0.027]	0.059	0.088	3
NN	fb-forum	0.602	0.532	+0.070	[-0.032, +0.173]	0.160	0.185	9
NN	CollegeMsg	0.673	0.501	+0.172	[+0.130, +0.214]	0.002	0.005	9
XGB	MIT	0.525	0.675	-0.150	[-0.573, +0.273]	0.457	0.490	3

Table 24: All matched comparisons with paired statistics (ten seeds, except PQKELP with LDA on Radoslaw-email under the new-links target, which uses nine because the classical fit did not converge on seed 10). CI is the t -based 95% interval of the mean paired difference; p is the two-sided Wilcoxon signed-rank test; q is Benjamini–Hochberg within each block. The last column counts seeds on which the classical counterpart returned a constant score (AUC exactly 0.5).

Classifier	Network	Quantum	Classical	Δ	95% CI	p	q	Const.
XGB	Radoslaw-email	0.682	0.894	−0.212	[−0.283, −0.142]	0.002	0.005	0
XGB	Eu-core	0.428	0.844	−0.416	[−0.543, −0.289]	0.002	0.005	0
XGB	fb-forum	0.662	0.790	−0.128	[−0.180, −0.076]	0.002	0.005	0
XGB	CollegeMsg	0.721	0.822	−0.101	[−0.131, −0.072]	0.002	0.005	0
LR	MIT	0.500	0.850	−0.350	[−0.729, +0.029]	0.125	0.156	3
LR	Radoslaw-email	0.677	0.905	−0.227	[−0.295, −0.160]	0.002	0.005	0
LR	Eu-core	0.503	0.889	−0.386	[−0.514, −0.258]	0.002	0.005	0
LR	fb-forum	0.652	0.712	−0.059	[−0.155, +0.036]	0.275	0.306	0
LR	CollegeMsg	0.728	0.785	−0.057	[−0.080, −0.033]	0.002	0.005	0
RFC	MIT	0.500	0.950	−0.450	[−0.806, −0.094]	0.062	0.089	1
RFC	Radoslaw-email	0.712	0.916	−0.204	[−0.261, −0.146]	0.002	0.005	0
RFC	Eu-core	0.499	0.860	−0.361	[−0.474, −0.248]	0.002	0.005	0
RFC	fb-forum	0.716	0.813	−0.098	[−0.149, −0.046]	0.004	0.008	0
RFC	CollegeMsg	0.731	0.809	−0.079	[−0.097, −0.060]	0.002	0.005	0
LDA	MIT	0.500	0.900	−0.400	[−0.682, −0.118]	0.031	0.052	2
LDA	Radoslaw-email	0.679	0.848	−0.169	[−0.237, −0.102]	0.002	0.005	0
LDA	Eu-core	0.495	0.611	−0.116	[−0.270, +0.039]	0.125	0.156	0
LDA	fb-forum	0.644	0.639	+0.005	[−0.090, +0.101]	0.943	0.976	0
LDA	CollegeMsg	0.731	0.736	−0.005	[−0.037, +0.026]	1.000	1.000	0
GNB	MIT	0.550	0.825	−0.275	[−0.548, −0.002]	0.078	0.107	1
GNB	Radoslaw-email	0.688	0.757	−0.068	[−0.123, −0.013]	0.037	0.059	0
GNB	Eu-core	0.491	0.766	−0.275	[−0.424, −0.126]	0.006	0.012	0
GNB	fb-forum	0.698	0.772	−0.074	[−0.116, −0.032]	0.010	0.018	0
GNB	CollegeMsg	0.692	0.787	−0.094	[−0.131, −0.058]	0.002	0.005	0
<i>PQKELP, relabelled targets</i>								
NN	MIT	0.903	0.535	+0.368	[+0.321, +0.415]	0.002	0.002	0
NN	Radoslaw-email	0.927	0.628	+0.300	[+0.235, +0.364]	0.002	0.002	1
NN	Eu-core	0.932	0.728	+0.204	[+0.097, +0.310]	0.002	0.002	0
NN	fb-forum	0.906	0.796	+0.111	[+0.082, +0.140]	0.002	0.002	0
NN	CollegeMsg	0.918	0.863	+0.055	[+0.033, +0.077]	0.002	0.002	0
XGB	MIT	0.929	0.725	+0.204	[+0.149, +0.259]	0.002	0.002	0
XGB	Radoslaw-email	0.943	0.905	+0.038	[+0.024, +0.052]	0.002	0.002	0
XGB	Eu-core	0.942	0.901	+0.041	[+0.029, +0.053]	0.002	0.002	0
XGB	fb-forum	0.930	0.875	+0.055	[+0.032, +0.078]	0.002	0.002	0
XGB	CollegeMsg	0.937	0.893	+0.044	[+0.030, +0.058]	0.002	0.002	0
LR	MIT	0.941	0.570	+0.370	[+0.334, +0.407]	0.002	0.002	0
LR	Radoslaw-email	0.947	0.624	+0.323	[+0.285, +0.361]	0.002	0.002	0
LR	Eu-core	0.949	0.652	+0.297	[+0.237, +0.357]	0.002	0.002	0
LR	fb-forum	0.941	0.711	+0.229	[+0.179, +0.280]	0.002	0.002	0
LR	CollegeMsg	0.948	0.820	+0.128	[+0.101, +0.155]	0.002	0.002	0
RFC	MIT	0.934	0.711	+0.223	[+0.165, +0.281]	0.002	0.002	0
RFC	Radoslaw-email	0.945	0.903	+0.042	[+0.027, +0.058]	0.002	0.002	0
RFC	Eu-core	0.946	0.888	+0.058	[+0.041, +0.076]	0.002	0.002	0
RFC	fb-forum	0.934	0.878	+0.055	[+0.034, +0.077]	0.002	0.002	0
RFC	CollegeMsg	0.944	0.910	+0.033	[+0.021, +0.046]	0.002	0.002	0
LDA	MIT	0.937	0.571	+0.367	[+0.330, +0.403]	0.002	0.002	0
LDA	Radoslaw-email	0.944	0.623	+0.321	[+0.283, +0.359]	0.002	0.002	0
LDA	Eu-core	0.945	0.651	+0.294	[+0.234, +0.355]	0.002	0.002	0
LDA	fb-forum	0.935	0.710	+0.226	[+0.176, +0.275]	0.002	0.002	0
LDA	CollegeMsg	0.943	0.818	+0.125	[+0.101, +0.148]	0.002	0.002	0
GNB	MIT	0.936	0.595	+0.341	[+0.292, +0.390]	0.002	0.002	0
GNB	Radoslaw-email	0.944	0.619	+0.324	[+0.283, +0.365]	0.002	0.002	0
GNB	Eu-core	0.944	0.677	+0.267	[+0.211, +0.323]	0.002	0.002	0
GNB	fb-forum	0.933	0.729	+0.204	[+0.156, +0.252]	0.002	0.002	0
GNB	CollegeMsg	0.940	0.834	+0.107	[+0.075, +0.138]	0.002	0.002	0
<i>PQCLP, observed edges</i>								
QNN/NN Local	MIT	0.452	0.989	−0.537	[−0.584, −0.489]	0.002	0.002	0
QNN/NN Local	Radoslaw-email	0.508	0.949	−0.441	[−0.526, −0.357]	0.002	0.002	0
QNN/NN Local	Eu-core	0.674	0.979	−0.305	[−0.424, −0.186]	0.002	0.002	0
QNN/NN Local	fb-forum	0.491	0.871	−0.380	[−0.495, −0.265]	0.002	0.002	0
QNN/NN Local	CollegeMsg	0.531	0.781	−0.251	[−0.386, −0.115]	0.004	0.004	0
VQC/SVM Local	MIT	0.453	0.988	−0.535	[−0.578, −0.492]	0.002	0.002	0
VQC/SVM Local	Radoslaw-email	0.510	0.946	−0.436	[−0.521, −0.351]	0.002	0.002	0
VQC/SVM Local	Eu-core	0.679	0.977	−0.298	[−0.414, −0.182]	0.002	0.002	0
VQC/SVM Local	fb-forum	0.499	0.857	−0.358	[−0.465, −0.251]	0.002	0.002	0
VQC/SVM Local	CollegeMsg	0.533	0.710	−0.177	[−0.305, −0.048]	0.006	0.006	0

Table 24: All matched comparisons with paired statistics (ten seeds, except PQKELP with LDA on Radoslaw-email under the new-links target, which uses nine because the classical fit did not converge on seed 10). CI is the t -based 95% interval of the mean paired difference; p is the two-sided Wilcoxon signed-rank test; q is Benjamini–Hochberg within each block. The last column counts seeds on which the classical counterpart returned a constant score (AUC exactly 0.5).

Classifier	Network	Quantum	Classical	Δ	95% CI	p	q	Const.
QNN/NN Global	MIT	0.514	0.999	−0.484	[−0.521, −0.448]	0.002	0.002	0
QNN/NN Global	Radoslaw-email	0.484	0.982	−0.498	[−0.533, −0.463]	0.002	0.002	0
QNN/NN Global	Eu-core	0.582	0.995	−0.413	[−0.462, −0.365]	0.002	0.002	0
QNN/NN Global	fb-forum	0.598	0.955	−0.356	[−0.443, −0.269]	0.002	0.002	0
QNN/NN Global	CollegeMsg	0.536	0.877	−0.341	[−0.440, −0.242]	0.002	0.002	0
VQC/SVM Global	MIT	0.515	0.995	−0.480	[−0.516, −0.444]	0.002	0.002	0
VQC/SVM Global	Radoslaw-email	0.485	0.952	−0.468	[−0.505, −0.431]	0.002	0.002	0
VQC/SVM Global	Eu-core	0.583	0.979	−0.396	[−0.442, −0.350]	0.002	0.002	0
VQC/SVM Global	fb-forum	0.599	0.884	−0.285	[−0.377, −0.193]	0.002	0.002	0
VQC/SVM Global	CollegeMsg	0.539	0.825	−0.286	[−0.378, −0.194]	0.002	0.002	0
QNN/NN LG	MIT	0.460	0.999	−0.539	[−0.560, −0.518]	0.002	0.002	0
QNN/NN LG	Radoslaw-email	0.508	0.981	−0.473	[−0.514, −0.433]	0.002	0.002	0
QNN/NN LG	Eu-core	0.521	0.994	−0.473	[−0.528, −0.419]	0.002	0.002	0
QNN/NN LG	fb-forum	0.544	0.946	−0.402	[−0.538, −0.266]	0.002	0.002	0
QNN/NN LG	CollegeMsg	0.554	0.877	−0.323	[−0.440, −0.206]	0.002	0.002	0
VQC/SVM LG	MIT	0.460	0.993	−0.533	[−0.554, −0.512]	0.002	0.002	0
VQC/SVM LG	Radoslaw-email	0.508	0.948	−0.439	[−0.479, −0.400]	0.002	0.002	0
VQC/SVM LG	Eu-core	0.523	0.981	−0.458	[−0.510, −0.406]	0.002	0.002	0
VQC/SVM LG	fb-forum	0.545	0.877	−0.332	[−0.474, −0.190]	0.002	0.002	0
VQC/SVM LG	CollegeMsg	0.555	0.823	−0.268	[−0.380, −0.156]	0.002	0.002	0
<i>PQCLP, new links only</i>								
QNN/NN Local	MIT	0.350	0.950	−0.600	[−0.929, −0.271]	0.014	0.015	1
QNN/NN Local	Radoslaw-email	0.516	0.861	−0.345	[−0.428, −0.262]	0.002	0.003	0
QNN/NN Local	Eu-core	0.461	0.952	−0.491	[−0.706, −0.275]	0.002	0.003	0
QNN/NN Local	fb-forum	0.513	0.736	−0.223	[−0.314, −0.132]	0.004	0.005	0
QNN/NN Local	CollegeMsg	0.451	0.705	−0.254	[−0.329, −0.178]	0.002	0.003	0
VQC/SVM Local	MIT	0.350	0.850	−0.500	[−0.913, −0.087]	0.043	0.043	1
VQC/SVM Local	Radoslaw-email	0.517	0.886	−0.370	[−0.444, −0.296]	0.002	0.003	0
VQC/SVM Local	Eu-core	0.463	0.928	−0.466	[−0.686, −0.245]	0.002	0.003	0
VQC/SVM Local	fb-forum	0.517	0.700	−0.183	[−0.278, −0.088]	0.002	0.003	0
VQC/SVM Local	CollegeMsg	0.451	0.538	−0.087	[−0.153, −0.022]	0.020	0.020	0
QNN/NN Global	MIT	0.400	0.950	−0.550	[−0.814, −0.286]	0.008	0.009	1
QNN/NN Global	Radoslaw-email	0.515	0.871	−0.356	[−0.390, −0.322]	0.002	0.003	0
QNN/NN Global	Eu-core	0.427	0.945	−0.519	[−0.677, −0.360]	0.002	0.003	0
QNN/NN Global	fb-forum	0.464	0.793	−0.329	[−0.458, −0.199]	0.002	0.003	0
QNN/NN Global	CollegeMsg	0.409	0.823	−0.415	[−0.479, −0.350]	0.002	0.003	0
VQC/SVM Global	MIT	0.400	0.900	−0.500	[−0.792, −0.208]	0.016	0.017	2
VQC/SVM Global	Radoslaw-email	0.514	0.878	−0.364	[−0.398, −0.329]	0.002	0.003	0
VQC/SVM Global	Eu-core	0.425	0.939	−0.514	[−0.669, −0.359]	0.002	0.003	0
VQC/SVM Global	fb-forum	0.467	0.797	−0.330	[−0.472, −0.189]	0.002	0.003	0
VQC/SVM Global	CollegeMsg	0.409	0.792	−0.383	[−0.447, −0.318]	0.002	0.003	0
QNN/NN LG	MIT	0.100	0.950	−0.850	[−1.023, −0.677]	0.002	0.003	1
QNN/NN LG	Radoslaw-email	0.455	0.875	−0.420	[−0.480, −0.361]	0.002	0.003	0
QNN/NN LG	Eu-core	0.453	0.942	−0.489	[−0.590, −0.388]	0.002	0.003	0
QNN/NN LG	fb-forum	0.528	0.780	−0.252	[−0.377, −0.128]	0.004	0.005	0
QNN/NN LG	CollegeMsg	0.401	0.802	−0.401	[−0.476, −0.327]	0.002	0.003	0
VQC/SVM LG	MIT	0.100	0.950	−0.850	[−1.023, −0.677]	0.002	0.003	1
VQC/SVM LG	Radoslaw-email	0.455	0.879	−0.424	[−0.490, −0.359]	0.002	0.003	0
VQC/SVM LG	Eu-core	0.455	0.928	−0.473	[−0.584, −0.363]	0.002	0.003	0
VQC/SVM LG	fb-forum	0.530	0.789	−0.259	[−0.374, −0.144]	0.002	0.003	0
VQC/SVM LG	CollegeMsg	0.402	0.784	−0.382	[−0.462, −0.302]	0.002	0.003	0

References

- [1] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011. doi: 10.1016/j.physa.2010.11.027.
- [2] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007. doi: 10.1002/asi.20591.

- [3] István A. Kovács, Katja Luck, Kerstin Spirohn, Yang Wang, Carl Pollis, Sadie Schlabach, Wenting Bian, Dae-Kyum Kim, Nishka Kishore, Tong Hao, Michael A. Calderwood, Marc Vidal, and Albert-László Barabási. Network-based prediction of protein interactions. *Nature Communications*, 10(1):1240, 2019. doi: 10.1038/s41467-019-09177-y.
- [4] Djihad Arrar, Nadjat Kamel, and Abdelaziz Lakhfif. A comprehensive survey of link prediction methods. *The Journal of Supercomputing*, 80(3):3902–3942, 2024. doi: 10.1007/s11227-023-05591-8.
- [5] Lada A Adamic and Eytan Adar. Friends and neighbors on the Web. *Social Networks*, 25(3):211–230, 2003. doi: 10.1016/s0378-8733(03)00009-1.
- [6] M. E. J. Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001. doi: 10.1103/physreve.64.025102.
- [7] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630, 2009. doi: 10.1140/epjb/e2009-00335-8.
- [8] Linyuan Lü, Ci-Hang Jin, and Tao Zhou. Similarity index based on local paths for link prediction of complex networks. *Physical Review E*, 80(4):046122, 2009. doi: 10.1103/physreve.80.046122.
- [9] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014. doi: 10.1145/2623330.2623732.
- [10] Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016. doi: 10.1145/2939672.2939754.
- [11] Bryan Perozzi, Vivek Kulkarni, Haochen Chen, and Steven Skiena. Don’t Walk, Skip!: On-line Learning of Multi-scale Network Embeddings. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 258–265, 2017. doi: 10.1145/3110025.3110086.
- [12] Giang Hoang Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen K. Ahmed, Eunye Koh, and Sungchul Kim. Continuous-Time Dynamic Network Embeddings. In *Companion Proceedings of The Web Conference 2018*, pages 969–976, 2018. doi: 10.1145/3184558.3191526.
- [13] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, 2016.
- [14] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs, 2017.
- [15] Petar Velickovi, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks, 2017.
- [16] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal Graph Networks for Deep Learning on Dynamic Graphs, 2020.
- [17] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1269–1278, 2019. doi: 10.1145/3292500.3330895.
- [18] Farimah Poursafaei, Shenyang Huang, Kellin Pelrine, and Reihaneh Rabbany. Towards Better Evaluation for Dynamic Link Prediction, 2022.

- [19] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017. doi: 10.1038/nature23474.
- [20] Maria Schuld and Nathan Killoran. Quantum Machine Learning in Feature Hilbert Spaces. *Physical Review Letters*, 122(4):040504, 2019. doi: 10.1103/physrevlett.122.040504.
- [21] Vojtech Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019. doi: 10.1038/s41586-019-0980-2.
- [22] João P. Moutinho, André Melo, Bruno Coutinho, István A. Kovács, and Yasser Omar. Quantum link prediction in complex networks. *Physical Review A*, 107(3):032605, 2023. doi: 10.1103/physreva.107.032605.
- [23] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean. Power of data in quantum machine learning. *Nature Communications*, 12(1):2631, 2021. doi: 10.1038/s41467-021-22539-9.
- [24] Supanut Thanasilp, Samson Wang, M. Cerezo, and Zoë Holmes. Exponential concentration in quantum kernel methods. *Nature Communications*, 15(1):5200, 2024. doi: 10.1038/s41467-024-49287-w.
- [25] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019. doi: 10.1088/2058-9565/ab4eb5.
- [26] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021. doi: 10.1038/s42254-021-00348-9.
- [27] Mukesh Kumar, Shivansh Mishra, and Bhaskar Biswas. PQKLP: Projected Quantum Kernel based Link Prediction in Dynamic Networks. *Computer Communications*, 196:249–267, 2022. doi: 10.1016/j.comcom.2022.10.006.
- [28] Nisha Singh, Mukesh Kumar, and Bhaskar Biswas. PQCLP: Parameterized quantum circuit based link prediction in dynamic networks. *Computer Networks*, 241:110210, 2024. doi: 10.1016/j.comnet.2024.110210.
- [29] Mukesh Kumar, Nisha Singh, and Bhaskar Biswas. PQKELP: Projected Quantum Kernel Embedding based Link Prediction in dynamic networks. *Expert Systems with Applications*, 265:125944, 2025. doi: 10.1016/j.eswa.2024.125944.
- [30] Nisha Singh, Mukesh Kumar, Jaideep Sharma, and Bhaskar Biswas. QGCNLP: Hybrid Quantum–classical Graph Convolutional Network based Link Prediction. *Physica A: Statistical Mechanics and its Applications*, 689:131449, 2026. doi: 10.1016/j.physa.2026.131449.
- [31] Nouhaila Innan, M. Murali Karthick, Simeon Kandan Sonar, Vivek Chaturvedi, and Muhammad Shafique. A2QTGN: Adaptive Amplitude Quantum-Integrated Temporal Graph Network for Dynamic Link Prediction, 2026.
- [32] Shashank Sheshar Singh, Sumit Kumar, Sunil Kumar Meena, Kuldeep Singh, Shivansh Mishra, and Albert Y. Zomaya. Quantum social network analysis: Methodology, implementation, challenges, and future directions. *Information Fusion*, 117:102808, 2025. doi: 10.1016/j.inffus.2024.102808.

- [33] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shah Nawaz Ahmed, Vishnu Ajith, M. Sohaib Alam, Guillermo Alonso-Linaje, B. Akash Narayanan, Ali Asadi, Juan Miguel Arrazola, Utkarsh Azad, Sam Banning, Carsten Blank, Thomas R Bromley, Benjamin A. Cordier, Jack Ceroni, Alain Delgado, Olivia Di Matteo, Amintor Dusko, Tanya Garg, Diego Guala, Anthony Hayes, Ryan Hill, Aroosa Ijaz, Theodor Isacsson, David Ittah, Soran Jahangiri, Prateek Jain, Edward Jiang, Ankit Khandelwal, Korbinian Kottmann, Robert A. Lang, Christina Lee, Thomas Loke, Angus Lowe, Keri McKiernan, Johannes Jakob Meyer, J. A. Montañez-Barrera, Romain Moyard, Zeyue Niu, Lee James O’Riordan, Steven Oud, Ashish Panigrahi, Chae-Yeun Park, Daniel Polatajko, Nicolás Quesada, Chase Roberts, Nahum Sá, Isidor Schoch, Borun Shi, Shuli Shu, Sukin Sim, Arshpreet Singh, Ingrid Strandberg, Jay Soni, Antal Száva, Slimane Thabet, Rodrigo A. Vargas-Hernández, Trevor Vincent, Nicola Vitucci, Maurice Weber, David Wierichs, Roeland Wiersema, Moritz Willmann, Vincent Wong, Shaoming Zhang, and Nathan Killoran. PennyLane: Automatic differentiation of hybrid quantum-classical computations, 2018.
- [34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019.
- [35] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80, 1945. doi: 10.2307/3001968.
- [36] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.
- [37] Carter Chiu and Justin Zhan. Deep Learning for Link Prediction in Dynamic Networks Using Weak Estimators. *IEEE Access*, 6:35937–35945, 2018. doi: 10.1109/access.2018.2845876.
- [38] J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992. doi: 10.1109/9.119632.
- [39] M. J. D. Powell. A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. In *Advances in Optimization and Numerical Analysis*, pages 51–67. Springer Netherlands, 1994. doi: 10.1007/978-94-015-8330-5_4.
- [40] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016. doi: 10.1145/2939672.2939785.
- [41] Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. Temporal Graph Benchmark for Machine Learning on Temporal Graphs, 2023.
- [42] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs, 2020.
- [43] Michael Broughton, Guillaume Verdon, Trevor McCourt, Antonio J. Martinez, Jae Hyeon Yoo, Sergei V. Isakov, Philip Massey, Ramin Halavati, Murphy Yuezheng Niu, Alexander Zlokapa, Evan Peters, Owen Lockwood, Andrea Skolik, Sofiene Jerbi, Vedran Dunjko, Martin Leib, Michael Streif, David Von Dollen, Hongxiang Chen, Shuxiang Cao, Roeland Wiersema, Hsin-Yuan Huang, Jarrod R. McClean, Ryan Babbush, Sergio Boixo, Dave Bacon, Alan K. Ho,

- Hartmut Neven, and Masoud Mohseni. TensorFlow Quantum: A Software Framework for Quantum Machine Learning, 2020.
- [44] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with Qiskit, 2024.
 - [45] Matthias Fey and Jan Eric Lenssen. Fast Graph Representation Learning with PyTorch Geometric, 2019.
 - [46] Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006. doi: 10.1007/s00779-005-0046-3.
 - [47] Radosaw Michalski, Sebastian Palus, and Przemysaw Kazienko. Matching Organizational Structure and Social Network Extracted from Email Communication. In *Business Information Systems Workshops*, Lecture Notes in Business Information Processing, pages 197–206. Springer, 2011. doi: 10.1007/978-3-642-21863-7_17.
 - [48] Ashwin Paranjape, Austin R. Benson, and Jure Leskovec. Motifs in Temporal Networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 601–610, 2017. doi: 10.1145/3018661.3018731.
 - [49] Tore Opsahl. Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Social Networks*, 35(2):159–167, 2013. doi: 10.1016/j.socnet.2011.07.001.
 - [50] Pietro Panzarasa, Tore Opsahl, and Kathleen M. Carley. Patterns and dynamics of users’ behavior and interaction: Network analysis of an online community. *Journal of the American Society for Information Science and Technology*, 60(5):911–932, 2009. doi: 10.1002/asi.21015.
 - [51] Jérôme Kunegis. KONECT: the Koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1343–1350, 2013. doi: 10.1145/2487788.2488173.
 - [52] Ryan Rossi and Nesreen Ahmed. The Network Data Repository with Interactive Graph Analytics and Visualization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015. doi: 10.1609/aaai.v29i1.9277.
 - [53] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective Classification in Network Data. *AI Magazine*, 29(3):93–106, 2008. doi: 10.1609/aimag.v29i3.2157.
 - [54] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting Semi-Supervised Learning with Graph Embeddings, 2016.
 - [55] Péter Mernyei and Ctina Cangea. Wiki-CS: A Wikipedia-Based Benchmark for Graph Neural Networks, 2020.
 - [56] Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 2017. doi: 10.1093/bioinformatics/btx252.
 - [57] Katja Luck, Dae-Kyum Kim, Luke Lambourne, Kerstin Spirohn, Bridget E. Begg, Wenting Bian, Ruth Brignall, Tiziana Cafarelli, Francisco J. Campos-Laborie, Benoit Charlotteaux, Dongsic Choi, Atina G. Côté, Meaghan Daley, Steven Deimling, Alice Desbuleux, Amélie Dricot, Marinella Gebbia, Madeleine F. Hardy, Nishka Kishore, Jennifer J. Knapp, István A. Kovács, Irma Lemmens, Miles W. Mee, Joseph C. Mellor, Carl Pollis, Carles Pons, Aaron D. Richardson, Sadie Schlabach, Bridget Teeking, Anupama Yadav, Mariana Babor, Dawit Balcha, Omer Basha, Christian Bowman-Colin, Suet-Feung Chin, Soon Gang Choi, Claudia

- Colabella, Georges Coppin, Cassandra D’Amata, David De Ridder, Steffi De Rouck, Miquel Duran-Frigola, Hanane Ennajdaoui, Florian Goebels, Liana Goehring, Anjali Gopal, Ghazal Haddad, Elodie Hatchi, Mohamed Helmy, Yves Jacob, Yoseph Kassa, Serena Landini, Roujia Li, Natascha van Lieshout, Andrew MacWilliams, Dylan Markey, Joseph N. Paulson, Sudharshan Rangarajan, John Rasla, Ashyad Rayhan, Thomas Rolland, Adriana San-Miguel, Yun Shen, Dayag Sheykhkarimli, Gloria M. Sheynkman, Eyal Simonovsky, Murat Taan, Alexander Tejeda, Vincent Tropepe, Jean-Claude Twizere, Yang Wang, Robert J. Weatheritt, Jochen Weile, Yu Xia, Xinping Yang, Esti Yeger-Lotem, Quan Zhong, Patrick Aloy, Gary D. Bader, Javier De Las Rivas, Suzanne Gaudet, Tong Hao, Janusz Rak, Jan Tavernier, David E. Hill, Marc Vidal, Frederick P. Roth, and Michael A. Calderwood. A reference map of the human binary protein interactome. *Nature*, 580(7803):402–408, 2020. doi: 10.1038/s41586-020-2188-x.
- [58] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1(1):2, 2007. doi: 10.1145/1217299.1217301.
- [59] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1361–1370, 2010. doi: 10.1145/1753326.1753532.
- [60] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python, 2012.
- [61] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. Karate Club: An API Oriented Open-Source Python Framework for Unsupervised Learning on Graphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3125–3132, 2020. doi: 10.1145/3340531.3412757.