PHOTE: PHOTONIC QUANTUM-WALK ENCODINGS FOR TEMPORAL LINK PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Jointly learning graph structure and temporal dependencies can make temporal link prediction computationally expensive. Precomputed features avoid structural-encoder optimization, yet their usefulness depends on which information they retain. We introduce **PHOTE**, a photonic framework combining non-trainable continuous-time quantum-walk encodings with a learned temporal predictor. One- and two-photon statistics at multiple evolution times are projected into identity-sensitive features or reduced to role summaries, yielding four encodings. Features are precomputed per snapshot; only a recurrent readout shared across nodes and a link scorer are trained. We evaluate PHOTE against nine temporal baselines, two classical positional encodings, and two heuristics on nine real networks and 53 synthetic settings from T-GRAB. The two-photon encoding achieves higher mean test average precision than the evaluated temporal baselines and classical encodings on eight of nine real networks. Across its four variants, PHOTE attains or ties the highest mean F_1 in five of seven synthetic task–metric comparisons. The identity-sensitive encodings also approach the optimal expected-count F_1 reference for stochastic periodicity. After feature precomputation, training takes 0.1–9.6 seconds per epoch, compared with 1.0–783.3 seconds for baselines on the same task families. One-photon measurements from a 24-mode processor yield AP comparable to that obtained with analytic features on the full test snapshots of a 13-node Guinea baboon contact network. Results support non-trainable photonic encodings for temporal prediction and demonstrate small-scale hardware feasibility. Although the present encodings are classically simulable, these results motivate exploring richer encodings on future multiphoton quantum processors.

1 INTRODUCTION

Temporal link prediction uses historical interactions to predict which pairs of nodes will interact in the future. We consider the discrete-time setting, in which a sequence of graph snapshots is used to predict the edges in the next snapshot. This requires using both graph structure and its evolution over time. Many temporal graph neural networks (GNNs) learn structural representations together with temporal dependencies, using neighborhood aggregation, attention, or memory updates (Rossi et al., 2020; Yu et al., 2023). Although these approaches achieve strong predictive performance, repeatedly computing and optimizing graph representations can incur substantial training time. Recent work has sought to reduce these costs through more efficient walk-based features and simpler temporal architectures (Lu et al., 2024; Li et al., 2025). An alternative is to construct non-trainable structural features for each snapshot and restrict learning to a temporal predictor. These features can be computed once and reused throughout training, avoiding repeated optimization of a structural encoder. This separation does not remove the initial cost of feature computation, and its usefulness depends on whether the fixed features retain enough structural information for accurate prediction. We investigate quantum walks as a source of such features. Their graph-dependent propagation statistics have been used to measure graph similarity, distinguish non-isomorphic graphs, and predict missing links in static networks (Rossi et al., 2015; Douglas & Wang, 2008; Goldsmith et al., 2023). Rather than comparing entire graphs or assigning link scores directly from walks, we use these statistics to represent individual nodes and learn from their feature histories.

In this paper, we introduce **PHOTE**, which combines a non-trainable photonic structural encoder with a learned classical temporal predictor. For each snapshot, the encoder constructs node features

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

from statistics describing the propagation of one or two photons through a graph-dependent optical transformation. We organize these statistics into two forms: identity-sensitive profiles that retain information about specific graph nodes, and compact role summaries that remove the correspondence between non-self feature values and those nodes. A shared recurrent network processes the feature history of each node, and a link scorer predicts future interactions. Only the recurrent network and link scorer are trained; the photonic encoder extracts structure independently from each snapshot. The underlying transformations admit a linear-optical implementation (Mezher et al., 2023), allowing us to study both analytically computed and measured features.

Our evaluation examines predictive performance and computational cost, the prediction requirements that favor different encodings, and the feasibility of using features measured on photonic hardware. We compare PHoTE with nine trained temporal GNNs, two classical positional encodings, and two heuristics on nine real interaction networks. We also evaluate 53 synthetic settings from T-GRAB (Dizaji et al., 2026), covering memorization, counting, stochastic periodicity, delayed dependencies, and long-range spatial-temporal dependencies. A small contact network provides a further test of the pipeline using photonic measurements.

Contributions. We introduce **PHoTE**, a temporal link prediction framework with four non-trainable photonic quantum-walk encodings, constructed from either one- or two-photon statistics using identity-sensitive projections or role summaries. Only a shared LSTM and an MLP link scorer are trained, reducing training time per epoch relative to the evaluated temporal GNN baselines after feature precomputation. Across synthetic task families, we identify complementary strengths: identity-sensitive encodings perform better on memorization, counting, and stochastic periodicity, while role summaries perform better on long-range memory and cause-and-effect. On eight of nine real networks, the two-photon encoding achieves higher mean average precision than the evaluated temporal GNNs and classical positional encodings. We also derive the optimal expected-count F_1 reference for stochastic periodicity when the target community assignment is known; the one- and two-photon encodings closely approach this reference. Finally, one-photon measurements from a 24-mode processor yield prediction performance comparable to analytic features on full test snapshots of a 13-node Guinea baboon contact network, demonstrating small-scale hardware feasibility.

2 Background

Temporal link prediction. Let $G_t = (V, E_t)$ denote the t-th snapshot of a temporal graph, with a fixed vertex set $V = \{1, \dots, M\}$ and a time-dependent edge set E_t. We represent each snapshot by its symmetric binary adjacency matrix $A_t \in \{0,1\}^{M\times M}$. Given the observed history $(G_1, \dots, G_t)$, the task for the predictor is to use the feature histories available up to time t to assign scores to candidate node pairs indicating whether they will form an edge in G_{t+1}.

Photonic states and linear-optical circuits. Consider a photonic processor with M optical modes, which serve as input and output channels for photons. A multimode Fock state $|n_1, \dots, n_M\rangle$ specifies a definite number n_i of photons in each mode (Kok et al., 2007; Heurtel et al., 2023). Injecting one photon into mode v, with all other modes empty, prepares the state $|v\rangle \equiv |0, \dots, 1, \dots, 0\rangle$, where the 1 is in position v. These M one-photon Fock states form an orthonormal basis, which we identify with the standard basis of $\mathbb{C}^M$. An ideal, lossless, passive linear-optical circuit consists of beam splitters and phase shifters and is described at the mode level by a unitary matrix $U \in \mathbb{C}^{M\times M}$ (Kok et al., 2007; Clements et al., 2016). For a one-photon input $|v\rangle$, the output state is $|\psi_{\text{out}}\rangle = U|v\rangle = \sum_{i=1}^{M} \alpha_i |i\rangle,\ \alpha_i = U[i, v]$. The coefficients α_i are complex probability amplitudes satisfying $\sum_i |\alpha_i|^2 = 1$, and $|\alpha_i|^2$ is the probability of detecting the photon in output mode i. The circuit can therefore produce a superposition of one-photon Fock states across the output modes. The interferometer settings determine U and hence the output amplitudes and detection probabilities. The same $M \times M$ mode transformation U determines the two-photon output amplitudes, but these amplitudes correspond to output-mode pairs rather than individual modes (Aaronson & Arkhipov, 2011).

Quantum walks and photonic implementation. A continuous-time quantum walk evolves complex probability amplitudes over the nodes of a graph (Farhi & Gutmann, 1998). The matrix governing this evolution is called the Hamiltonian. Using the snapshot adjacency matrix as the Hamiltonian gives the transformation $U_t(\tau) = \exp(-\mathrm{i}A_t\tau)$, where i is the imaginary unit and τ is the walk's evolution time. The snapshot index t and the evolution time τ have different meanings: t indexes the

changing graph, whereas τ controls a walk on one fixed snapshot. Since A_t is symmetric, $U_t(\tau)$ is unitary. Interference between amplitudes associated with graph walks of different lengths makes the output probabilities sensitive to both local connectivity and longer-range structure. Assigning one optical mode to each graph node allows $U_t(\tau)$ to be implemented by an M-mode linear interferometer, whose settings are determined by the graph and evolution time (Clements et al., 2016; Mezher et al., 2023). The same transformation can be probed using one- or two-photon inputs, which provide different output statistics.

One-photon probabilities. For a photon injected into the mode associated with node v, the amplitude for detection at output mode i is $\langle i|U_t(\tau)|v\rangle = U_t(\tau)[i, v]$, where $\langle i|$ is the conjugate transpose of $|i\rangle$. The corresponding detection probability is $Q_t(\tau)[i, v] = |\langle i|U_t(\tau)|v\rangle|^2 = |U_t(\tau)[i, v]|^2$. Collecting all input–output probabilities gives $Q_t(\tau) \in [0, 1]^{M\times M}$, with rows indexing output modes and columns indexing input modes. Thus, column v is the complete output probability distribution for input node v, and unitarity gives $\sum_{i=1}^{M} Q_t(\tau)[i, v] = 1$.

Two-photon coincidence probabilities. For each current edge $\{a, b\} \in E_t$ in turn, we inject one photon at node a and one at node b. The photons are indistinguishable. We measure the probability of detecting them at an unordered pair of output modes $\{i, j\}$, including $i = j$ when both photons are detected in the same mode. For compactness, let $U_{xy} = U_t(\tau)[x, y]$, and let δ_{xy} equal one when $x = y$ and zero otherwise. For $i \le j$, the edge-aggregated coincidence table is $E_t^{(2)}(\tau)[i, j] = \sum_{\{a,b\}\in E_t} \frac{|U_{ia}U_{jb}+U_{ib}U_{ja}|^2}{(1+\delta_{ij})(1+\delta_{ab})}$. The quantity $U_{ia}U_{jb} + U_{ib}U_{ja}$ is the permanent of the 2×2 matrix selected by rows i, j and columns a, b of $U_t(\tau)$. The two products represent the two indistinguishable ways in which the photons can reach the output pair. Adding their amplitudes before taking the squared modulus introduces the interference term $2\,\mathrm{Re}\left(U_{ia}U_{jb}\overline{U_{ib}U_{ja}}\right)$, where $\mathrm{Re}(\cdot)$ denotes the real part. This term can increase or decrease the coincidence probability depending on the relative phase. Such two-photon interference underlies the Hong–Ou–Mandel (HOM) effect (Hong et al., 1987). The denominator accounts for repeated input or output modes. Each input edge contributes one normalized probability distribution over unordered output pairs, so $\sum_{i\le j} E_t^{(2)}(\tau)[i, j] = |E_t|$, where $|E_t|$ is the number of edges in snapshot t. We extend the table symmetrically by defining $E_t^{(2)}(\tau)[i, j] = E_t^{(2)}(\tau)[j, i]$ for $i > j$.

3 Related work

Temporal graph learning and structural encodings. Temporal graph methods are commonly divided into discrete-time and continuous-time models. Discrete-time methods process sequences of graph snapshots, often combining graph convolutions with recurrent updates or evolving model parameters (Chen et al., 2021; Zhao et al., 2020; Pareja et al., 2019). Continuous-time methods operate directly on timestamped interactions and use temporal attention or event-driven memory (Xu et al., 2020; Rossi et al., 2020; Yu et al., 2023; Gravina et al., 2024). These approaches learn how graph structure and interaction history contribute to future links. Computational efficiency is an important design objective. EAGLE combines recent-neighbor information with temporal personalized PageRank to reduce complex multi-hop aggregation and memory-intensive components (Li et al., 2025). TPNet constructs relative encodings from time-decayed temporal walk matrices and uses random feature propagation to improve computational and storage efficiency (Lu et al., 2024). Structural features can also be supplied through fixed node descriptors, such as Laplacian positional encoding which represents nodes using graph-Laplacian eigenvectors and random-walk positional encoding which uses return probabilities from a classical random walk (Dwivedi et al., 2022; 2023). Other temporal models use walk-based or node-specific information. PINT combines injective temporal message passing with temporal walk-count positional features (Souza et al., 2022). Causal anonymous walks replace explicit node identities with walk-based occurrence information, allowing recurring temporal patterns to be recognized across different nodes (Wang et al., 2021). CRAFT instead learns node-specific embeddings and uses cross-attention to compare a candidate destination with the source node's recent interactions (Yi et al., 2025). Work on temporal graph evaluation has highlighted the importance of strong non-learned baselines, such as EdgeBank (Poursafaei et al., 2022), and introduced both diagnostic benchmarks targeting specific temporal reasoning abilities (Dizaji et al., 2026) and broader benchmark suites covering diverse datasets and prediction tasks (Huang et al., 2023).

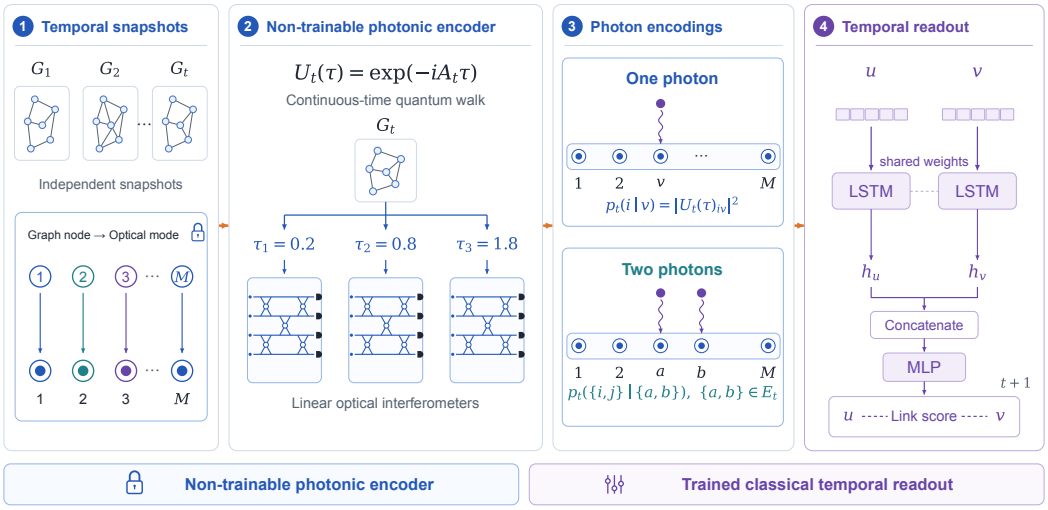


Figure 1: Overview of PHoTE.

Quantum graph features and photonic platforms. Quantum walks have been used for graph similarity and isomorphism testing. Continuous-time walks have been combined with quantum Jensen–Shannon divergence and graph kernels (Rossi et al., 2015; Bai et al., 2015), while discrete-time walks have been used to distinguish non-isomorphic graphs (Douglas & Wang, 2008). Quantum-walk-inspired methods also compare graphs without requiring a known correspondence between vertex labels (Schofield et al., 2020). The distinguishing power depends on the walk model, particle number, and statistics being compared. Interacting two-particle walks can improve discrimination between non-isomorphic strongly regular graphs (Gamble et al., 2010; Berry & Wang, 2011). Noninteracting multiparticle walks can distinguish additional pairs, although no fixed particle number suffices to distinguish all such graphs under the construction studied (Rudinger et al., 2012). Quantum walks have also been applied directly to link prediction. Goldsmith et al. (2023) define link scores from continuous-time classical and quantum-walk transition probabilities to reconstruct missing interactions in protein–protein interaction networks. Quantum-derived features have also been incorporated into classical graph-learning models. Thabet et al. (2023) introduce both precomputed and trainable positional encodings based on quantum walks and correlations between graph-associated qubits. Other approaches learn the walk dynamics: GQWformer uses discrete-time walks with learned coin operators (Yu et al., 2024), and CTQWformer learns a Hamiltonian from graph structure and node attributes, combining continuous-time walk probabilities with a graph Transformer and recurrent module for graph classification (Li et al., 2026). A fixed dynamical system followed by a trained readout is central to reservoir computing (Jaeger & Haas, 2004; Maass et al., 2002), including quantum reservoir computing (Fujii & Nakajima, 2017). In contrast to learned walk dynamics and reservoirs that retain temporal memory, PHoTE uses fixed walk parameters, encodes each snapshot independently, and learns temporal dependencies through a classical recurrent readout. On photonic platforms, linear-optical matrix encodings have been applied to graph problems using single-photon sources (Mezher et al., 2023), and graph search and isomorphism testing have been demonstrated on a silicon photonic quantum-walk processor (Qiang et al., 2021). Programmable photonic processors and simulation tools provide a basis for evaluating such transformations (Maring et al., 2024; Heurtel et al., 2023).

4 Methodology

Figure 1 gives an overview of PHoTE, which leverages continuous-time quantum walks to construct graph-dependent features with a direct linear optical implementation. The one-photon features use individual transition probabilities, whereas the two-photon features use coincidence probabilities. We use the same node-to-mode assignment across snapshots and construct each snapshot's encoding separately from its adjacency matrix. Evaluating the walk at several fixed evolution times samples different interference patterns determined by the snapshot's connectivity. The number L of evolution times and their values are hyperparameters fixed before readout training. In our experiments, we

use $L = 3$ and $(\tau_1, \tau_2, \tau_3) = (0.2, 0.8, 1.8)$. Appendix E.2 gives more details regarding different evolution times. The encodings do not contain trainable parameters.

One-photon encoding. In the one-photon encoding, a photon is injected into the mode associated with a selected node. At an evolution time τ, for input node v, we write column v of $Q_t(\tau)$ as $\mathbf{q}_{t,v}^{(1)}(\tau) = \big(Q_t(\tau)[1, v], \ldots, Q_t(\tau)[M, v]\big)^{\mathsf{T}} \in \mathbb{R}^M$. The complete one-photon encoding concatenates this vector at L evolution times:

$$\mathbf{x}_{t,v}^{(1)} = \big[\mathbf{q}_{t,v}^{(1)}(\tau_1) \,\|\, \mathbf{q}_{t,v}^{(1)}(\tau_2) \,\|\, \cdots \,\|\, \mathbf{q}_{t,v}^{(1)}(\tau_L)\big] \in \mathbb{R}^{LM}, \tag{1}$$

where $\|$ denotes vector concatenation.

Two-photon encoding. In the two-photon encoding, one photon is injected at each endpoint of a current edge, and coincidence measurements record the output-mode pairs. These coincidence distributions are summed over all current edges to construct node profiles. The vector assigned to node v at evolution time τ is $\mathbf{q}_{t,v}^{(2)}(\tau) = \big(E_t^{(2)}(\tau)[1, v], \ldots, E_t^{(2)}(\tau)[M, v]\big)^{\mathsf{T}}$. The complete two-photon encoding is

$$\mathbf{x}_{t,v}^{(2)} = \big[\mathbf{q}_{t,v}^{(2)}(\tau_1) \,\|\, \mathbf{q}_{t,v}^{(2)}(\tau_2) \,\|\, \cdots \,\|\, \mathbf{q}_{t,v}^{(2)}(\tau_L)\big] \in \mathbb{R}^{LM}. \tag{2}$$

At each evolution time, coordinate i is the sum of the coincidence probabilities for detector pair $\{i, v\}$ over all input edges. Because the table aggregates one distribution per edge, its columns do not generally sum to one.

Role summaries. The one- and two-photon encodings above retain a fixed correspondence between their coordinates and graph nodes. A role summary instead retains selected statistics that do not depend on the identities of the other nodes. Let $\mathbf{q} = (q_1, \ldots, q_M) \in \mathbb{R}^M$ be a one- or two-photon vector assigned to node v at one evolution time. Let $\mathbf{q}_{-v}$ denote the vector obtained by removing the self entry q_v. For a positive integer K_{top}, let $\mathrm{Top}_{K_{\mathrm{top}}}(\mathbf{q}_{-v})$ contain the K_{top} largest remaining entries, sorted in decreasing order. We also use the power sums $\mu_r(\mathbf{q}) = \sum_{i=1}^{M} q_i^r, r \in \{2, 3, 4\}$. The role summary at one evolution time is $\mathbf{r}_v(\mathbf{q}) = \big[q_v \,\|\, \mathrm{Top}_{K_{\mathrm{top}}}(\mathbf{q}_{-v}) \,\|\, \mu_2(\mathbf{q}) \,\|\, \mu_3(\mathbf{q}) \,\|\, \mu_4(\mathbf{q})\big]$. This vector has $K_{\mathrm{top}} + 4$ entries. For $s \in \{1, 2\}$, the complete role summary is

$$\mathbf{z}_{t,v}^{(s)} = \big[\mathbf{r}_v\big(\mathbf{q}_{t,v}^{(s)}(\tau_1)\big) \,\|\, \mathbf{r}_v\big(\mathbf{q}_{t,v}^{(s)}(\tau_2)\big) \,\|\, \cdots \,\|\, \mathbf{r}_v\big(\mathbf{q}_{t,v}^{(s)}(\tau_L)\big)\big] \in \mathbb{R}^{L(K_{\mathrm{top}}+4)}. \tag{3}$$

We call $\mathbf{z}_{t,v}^{(1)}$ **Q1-role** and $\mathbf{z}_{t,v}^{(2)}$ **Q2-role**, and use $K_{\mathrm{top}} = 4$ for both in all experiments. A node relabeling simultaneously permutes the rows and columns of $Q_t(\tau)$ and $E_t^{(2)}(\tau)$. For each node and its relabeled counterpart, the self entry retains its value, while the non-self entries are merely reordered. The sorted non-self values and the power sums are therefore unchanged, yielding identical role summaries. If a relabeling leaves G_t unchanged and maps node v to node w, the two nodes receive identical role summaries. The temporal model still receives a separate feature sequence for every node; only the correspondence between non-self coordinates and destination nodes is removed.

Projection of the encodings. The full one- and two-photon encodings have dimension LM. To reduce the input dimension of the temporal readout, we apply a fixed random linear projection to D dimensions, where D is a hyperparameter. In our experiments, we set $D = M$, matching the dimension of the classical positional encodings. For each photon encoding, we independently draw a projection matrix $R^{(s)} \in \mathbb{R}^{D \times LM}$ and compute $\widetilde{\mathbf{x}}_{t,v}^{(s)} = R^{(s)}\mathbf{x}_{t,v}^{(s)},\ s \in \{1, 2\}$. The entries of each projection matrix are sampled independently from $\mathcal{N}(0, \sigma^2)$, with $\sigma^2 = 1/D$. Each matrix is then held fixed across all nodes and snapshots throughout training and evaluation. Both role-summary encodings are used without projection. Before projection, the one- and two-photon encodings retain a coordinate for each destination node at each evolution time. The fixed projection mixes these coordinates but remains sensitive to their node ordering. We therefore refer to these projected encodings as identity-sensitive.

Temporal readout. Within each experimental setting, the four quantum-walk encodings use the same recurrent architecture. A shared LSTM network processes each node's sequence of features to learn temporal dependencies. At every time step, for each candidate node pair (u, w), the two nodes' hidden states produced by the LSTM are concatenated and passed to a small two-layer MLP, which produces a score indicating whether the pair will form an edge in the next snapshot. Only the LSTM and MLP are trained; no gradients are propagated through the quantum walk or the feature-construction steps. The walk, the random projection, and the role-summary operations remain fixed.

5 EXPERIMENT SETUP

Comparison methods. Inspired by T-GRAB (Dizaji et al., 2026) and UTG (Huang et al., 2024), we evaluate five models through the discrete-time pipeline (DTDG): GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), GC-LSTM (Chen et al., 2021), TGCN (Zhao et al., 2020), and EvolveGCN (EGCN) (Pareja et al., 2019); four continuous-time models (CTDG): TGAT (Xu et al., 2020), TGN (Rossi et al., 2020), DyGFormer (DyGF) (Yu et al., 2023), and CTAN (Gravina et al., 2024); as well as two non-learned heuristics (Persistence and EdgeBank (Poursafaei et al., 2022)). The discrete-time models process graph snapshots, whereas the continuous-time models process timestamped edge events. Persistence assigns a score of one to edges in the preceding snapshot. EdgeBank assigns one to edges observed anywhere in the preceding history for synthetic tasks, or within the preceding $\lceil 0.15T \rceil$ snapshots for real-data experiments, where T is the total number of snapshots in the stream. We run all nine models using the official T-GRAB implementation and its published configurations. These are complete trained models and do not use our temporal readout. We further evaluate two classical positional encodings: Laplacian positional encoding (LapPE) and random-walk positional encoding (RWPE). Both are computed independently for each snapshot and use the same temporal-readout architecture as our four quantum-walk encodings.

Datasets and splits. We use real temporal interaction networks to evaluate the predictive performance of PHOTE against the baselines on real-world data. Dept1 to Dept4 are department subnetworks of the email-Eu-core-temporal network, which records email exchanges inside a European research institution (Paranjape et al., 2017). Radoslaw is the internal email network of a mid-sized manufacturing company (Michalski et al., 2011). MIT Reality contains Bluetooth proximity contacts between members of a university community, recorded in the Reality Mining study (Eagle & Pentland, 2006). SocialEvo contains proximity contacts from the Social Evolution study of an undergraduate dormitory (Madan et al., 2012; Huang et al., 2024). Tgbn-trade records agricultural trade between countries, aggregated by year (Huang et al., 2023). Tgbn-genre is a bipartite network connecting users to the music genres of the songs they listen to, aggregated into weekly snapshots (Huang et al., 2023). We additionally use a 13-node Guinea baboon contact network (Gelardi et al., 2020) to assess feasibility on a 24-mode photonic quantum device. All networks are represented as sequences of undirected, unweighted graph snapshots and split chronologically into $70\%/15\%/15\%$ for training, validation and testing. Table 4 in Appendix A.1 lists their statistics.

We adopt all synthetic settings of T-GRAB (Dizaji et al., 2026), which cover five task families: `memorization`, `counting`, `stochastic periodicity`, `cause-and-effect`, and `long-range memory`; Appendix B.1 details each task. We regenerate all instances with the official generators, with $M = 100$ nodes per graph. Memorization, counting, and stochastic periodicity sequences contain 48 complete periods, split chronologically into 40 for training, 4 for validation, and 4 for testing; the cause-and-effect and long-range memory sequences are split chronologically into $80\%/10\%/10\%$.

Metrics. For the synthetic tasks, an edge is predicted when its score is at least 0.5, following the official T-GRAB evaluation. We report the F_1 score. For memorization, counting, and stochastic periodicity tasks, we compute F_1 over all unordered pairs of distinct nodes in each test snapshot and then average over the test split. For counting, we also report change-point F_1, evaluated only on the test snapshots at which the active graph changes. For long-range and cause-and-effect tasks, we follow T-GRAB and evaluate the $M - 1$ candidate edges incident to the designated memory node. For all the real interaction networks, we compute average precision (AP) separately in each test snapshot and report the mean over the test split. Unless stated otherwise, each trainable configuration is run with three random seeds. We report the mean and standard deviation across seeds.

6 RESULTS

6.1 REAL INTERACTION NETWORKS

We evaluate the real networks with AP instead of F_1: F_1 requires a decision threshold that need not suit every model and network, whereas AP summarizes precision across recall levels as the score threshold varies. AP is also standard in temporal link prediction (Rossi et al., 2020; Yu et al., 2023). Table 1 shows the two-photon encoding achieves the highest mean AP among the methods shown on eight of the nine networks. These results show that non-trainable photonic encodings can support

Table 1: Test average precision (AP) on real temporal interaction networks. The best GNN column reports the highest mean among all the temporal-GNN baseline results: DyGFormer for Dept1, Dept3, Dept4, and Radoslaw; GC-LSTM for MIT Reality and tgbn-genre; TGCN for Dept2 and SocialEvo; GCN for tgbn-trade. Top three means in each row are marked **First**, Second and *Third*.

	Temporal GNN	*Classical positional encodings*		*Quantum-walk encodings* (PHoTE)			
network	best GNN	LapPE	RWPE	one-photon	two-photon	Q1-role	Q2-role
Eu Dept1	0.139±0.027	0.071±0.007	0.138±0.005	0.170±0.006	**0.201**±0.012	*0.158*±0.010	0.156±0.016
Eu Dept2	0.195±0.011	0.079±0.005	0.129±0.009	*0.181*±0.022	**0.203**±0.010	0.152±0.038	0.131±0.019
Eu Dept3	**0.117**±0.012	0.028±0.002	0.042±0.003	0.098±0.004	*0.078*±0.003	0.038±0.007	0.041±0.006
Eu Dept4	*0.180*±0.003	0.071±0.024	0.112±0.007	0.215±0.006	**0.225**±0.001	0.162±0.003	0.174±0.007
Radoslaw	*0.235*±0.001	0.099±0.004	0.156±0.016	0.265±0.007	**0.280**±0.018	0.140±0.005	0.193±0.016
MIT Reality	0.284±0.009	0.181±0.075	0.196±0.065	*0.210*±0.025	**0.320**±0.022	0.186±0.020	0.158±0.045
SocialEvo	0.257±0.008	0.246±0.007	*0.325*±0.004	0.284±0.003	**0.344**±0.010	0.277±0.016	0.332±0.009
tgbn-trade	0.837±0.011	0.794±0.017	0.901±0.001	0.855±0.021	**0.915**±0.004	0.857±0.009	*0.867*±0.019
tgbn-genre	0.184±0.008	0.151±0.020	*0.169*±0.005	0.156±0.005	**0.222**±0.024	0.158±0.010	0.164±0.006

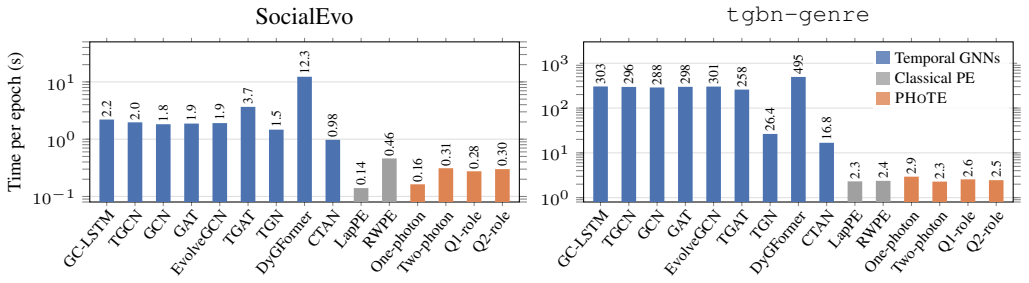


Figure 2: Mean training time per epoch (seconds, log scale) on SocialEvo and `tgbn-genre`.

competitive link prediction against trained temporal GNNs, even though only the temporal readout is trained. Dept3 is the only network on which a temporal GNN leads. Isolated nodes account for 72% of this network's node-snapshot pairs in the test period. Because the two-photon encoding uses current edges as inputs, no injected photon can reach an isolated node, whose two-photon profile is therefore all zero. A photon injected directly at an isolated node instead remains there, giving a nonzero, node-specific one-photon profile. This difference may help explain the higher mean of one-photon than two-photon on Dept3. The two-photon encoding has a higher mean than Q2-role on all nine networks, and the one-photon encoding has a higher mean than Q1-role on seven of them. The one- and two-photon profiles give each destination node the same feature position in every snapshot; the fixed projection mixes these coordinates but is identical across snapshots, so the temporal readout can still use node-specific information. The role summaries no longer record which node each non-self value comes from. The results suggest that retaining this correspondence is useful for real interaction networks, where prediction can depend on repeated interactions with specific partners. Persistence remains stronger on SocialEvo (0.355) and tgbn-genre (0.259). Appendix A.2 provides the full baseline results. As a complementary evaluation, we evaluate our encoding pipelines and additional GNN baselines on SocialEvo under the UTG protocol, using mean reciprocal rank (MRR) to assess candidate rankings. We compare these results with the published UTG results (Huang et al., 2024); Appendix A.3 reports the results. For `tgbn-trade` and `tgbn-genre`, Appendix A.4 describes how we convert the TGB node-affinity datasets into snapshot link prediction datasets.

Across all the real interaction networks, mean training time per epoch ranges from 0.118 s (Q2-role on `tgbn-trade`) to 9.439 s (two-photon on the 5-minute baboon sequence) for PHoTE. The temporal GNN baselines range from 0.976 s (CTAN on SocialEvo) to 494.768 s (DyGFormer on `tgbn-genre`). Figure 2 illustrates the comparison on two representative datasets, SocialEvo and `tgbn-genre`. Table 18 and Table 19 in Appendix F.2 provide the full per-dataset timings.

6.2 T-GRAB SYNTHETIC BENCHMARK RESULTS

Table 2 summarizes the test F_1 scores over the 53 synthetic settings from T-GRAB. The five task families give seven task–metric columns: memorization, counting, counting at change points (CP), stochastic periodicity (sto) at $p = 0.9$ and $p = 0.5$, long-range memory, and cause-and-effect. In counting, each period consists of n consecutive repetitions of a fixed graph A, followed by n consecutive repetitions of a fixed graph B. A change point is a snapshot at which the active graph

Table 2: Mean test scores over the 53 synthetic settings from T-GRAB, grouped by task family, and mean training time per epoch. For each seed the per-setting test scores are averaged within the family; each entry reports the mean and the sample standard deviation of that family average over the three seeds. Counting (CP) includes only change-point snapshots. The last column averages the training time per epoch (seconds) over the six task groups (except counting (CP)) of Table 17 in Appendix F.2. The top three entries of each column are marked **First**, Second and *Third*.

	memorization	counting	counting (CP)	sto p=0.9	sto p=0.5	long-range	cause-and-effect	time (s)
Non-learned baselines								
Persistence	0.009±0.000	0.858±0.000	0.000±0.000	0.296±0.000	0.168±0.000	0.033±0.000	0.149±0.000	–
EdgeBank	0.198±0.000	0.665±0.000	0.665±0.000	0.478±0.000	0.301±0.000	0.059±0.000	0.275±0.000	–
Discrete-time GNNs								
GCN	0.623±0.005	0.664±0.000	0.506±0.006	0.429±0.001	0.269±0.002	0.238±0.003	0.383±0.001	30.5
GAT	0.674±0.005	0.694±0.003	0.570±0.001	0.600±0.008	0.320±0.021	0.267±0.002	0.385±0.001	30.2
GC-LSTM	0.921±0.003	0.897±0.005	0.707±0.005	*0.933*±0.003	*0.625*±0.006	0.377±0.010	0.616±0.008	39.0
TGCN	0.879±0.004	0.889±0.009	0.671±0.015	0.857±0.005	0.405±0.004	0.388±0.016	0.629±0.012	33.2
EvolveGCN	0.092±0.010	0.363±0.039	0.150±0.013	0.462±0.000	0.290±0.000	0.078±0.001	0.366±0.003	31.2
Continuous-time GNNs								
TGAT	0.814±0.046	0.906±0.003	0.722±0.024	0.800±0.040	0.552±0.056	0.743±0.008	0.997±0.002	54.2
TGN	0.753±0.016	0.788±0.060	0.609±0.049	0.752±0.028	0.485±0.006	0.732±0.067	**0.999**±0.001	31.1
DyGFormer	*0.927*±0.013	0.951±0.005	0.759±0.009	0.668±0.038	0.345±0.001	*0.757*±0.026	*0.989*±0.002	329.6
CTAN	0.557±0.020	0.726±0.024	0.597±0.018	0.621±0.020	0.408±0.037	0.052±0.002	0.245±0.002	22.8
Quantum-walk encodings (PHoTE)								
one-photon	**0.988**±0.004	0.924±0.007	**0.812**±0.015	0.936±0.001	**0.657**±0.000	0.553±0.032	0.910±0.009	5.4
two-photon	0.946±0.068	*0.937*±0.014	*0.779*±0.017	**0.937**±0.000	**0.657**±0.000	0.679±0.014	0.960±0.005	6.5
Q1-role	0.910±0.067	0.721±0.002	0.555±0.004	0.583±0.015	0.322±0.033	**0.922**±0.014	0.976±0.013	6.6
Q2-role	0.909±0.066	0.738±0.003	0.586±0.008	0.461±0.027	0.374±0.014	0.907±0.043	0.978±0.018	6.4
Classical positional encodings								
LapPE	0.827±0.142	**0.957**±0.004	0.809±0.006	0.639±0.056	0.456±0.003	0.659±0.014	0.864±0.028	6.2
RWPE	0.924±0.006	0.654±0.010	0.508±0.005	0.445±0.010	0.275±0.007	0.738±0.033	0.965±0.005	5.6

switches. The counting column averages F_1 over all test snapshots, whereas counting (CP) evaluates only the change-point snapshots. For stochastic periodicity, the 100 nodes are partitioned into three communities of sizes $(34, 33, 33)$. The two stochastic-periodicity columns correspond to within-community edge probabilities $p = 0.9$ and $p = 0.5$: each pair of nodes in the same community is connected with probability p. The between-community edge probability q is 0.01 in both settings.

For each seed, we first average the scores equally over all settings in the family, then report the mean and sample standard deviation of the three resulting family averages. At least one of our four photonic encodings achieves or ties the highest mean in five of the seven task–metric comparisons, while LapPE leads on average counting F_1 and TGN leads on cause-and-effect. Stochastic periodicity can also be compared with a theoretical reference: even a predictor that knows the community assignment generating the target snapshot cannot tell which individual edges the next random sample will contain. Proposition 1 therefore computes F_1 from expected edge counts rather than averaging F_1 over individual sampled graphs, and derives the maximum of this expected-count score for the benchmark's community sizes and edge probabilities. The identity-sensitive one- and two-photon encodings closely approach this reference at both densities.

The family averages also show how the four photonic encodings differ. The identity-sensitive one- and two-photon encodings have higher means than both role summaries on memorization, counting, and stochastic periodicity. This pattern is consistent with destination-node information helping the readout distinguish recurring graphs and community assignments. In counting, this information can help identify which graph is being repeated before predicting a switch. In stochastic periodicity, it can help distinguish community assignments even though individual edges are sampled again in each snapshot. Q1-role and Q2-role instead have higher means than the identity-sensitive encodings on long-range memory and cause-and-effect. In these tasks, the readout must remember that a node was an endpoint of an earlier path or edge, even when the other participating nodes change across snapshots. The results are consistent with role summaries making these recurring structural events easier to recognize without requiring the same destination coordinates. Appendix E.3 illustrates this with an example of an isolated edge whose partner changes between snapshots: the one- and two-photon values move to a different coordinate, whereas the role summaries remain identical. Because some settings contain low-performing runs, these differences are also sensitive to variation between training runs; Appendix B.2 reports every setting of each task family, including these runs.

Proposition 1 (Expected-count F_1 for stochastic periodicity)**.** *Let M_{in} and M_{out} be the numbers of node pairs within and between communities, p and q be their respective edge probabilities. For the*

Table 3: Test AP on the 13-node baboon contact network. GC-LSTM is the best temporal GNN at both durations. The four quantum-walk columns use analytic features; `sim:belenos` and `qpu:belenos` replace the test features with simulator and hardware outputs, respectively. Hardware measurements cover the one-photon encoding at 15-minute aggregation only. The top three means in each row are marked **First**, Second and *Third*.

	Temporal GNN	*Classical PE*		*Quantum-walk encodings (*PHOTE*)*				*sim:belenos*		*qpu:belenos*
Window	best GNN	LapPE	RWPE	one-photon	two-photon	Q1-role	Q2-role	one-photon	two-photon	one-photon
15 min	0.399±0.006	0.309±0.002	0.390±0.005	**0.414**±0.006	0.412±0.015	0.380±0.003	0.388±0.003	0.401±0.014	*0.402*±0.006	0.398±0.012
5 min	*0.311*±0.005	0.292±0.005	0.275±0.008	0.308±0.016	**0.330**±0.011	0.283±0.010	0.270±0.011	0.300±0.007	0.312±0.008	–

community sizes and edge probabilities used in this benchmark, a deterministic predictor given the community assignment generating the target snapshot maximizes the expected-count F_1 by predicting every within-community pair and no between-community pair. Writing Rec *for recall computed from expected edge counts, the resulting score is* $F_1^\star = \frac{2p\,\mathrm{Rec}}{p+\mathrm{Rec}}$, *where* $\mathrm{Rec} = \frac{pM_{\mathrm{in}}}{pM_{\mathrm{in}}+qM_{\mathrm{out}}}$.

For three communities of sizes $(34, 33, 33)$ and $q = 0.01$, Proposition 1 gives $F_1^\star = 0.9372$ at $p = 0.9$ and $F_1^\star = 0.6576$ at $p = 0.5$. These values maximize the expected-count score but do not upper-bound F_1 on every finite sample. For these parameters, adding any within-community pair increases the expected-count score, whereas adding between-community pairs to the full within-community set lowers it; Appendix E.1 gives the complete proof.

The last column of Table 2 averages the per-epoch training time over the settings of each synthetic task group; Table 17 in Appendix F.2 gives the timings for every method and task group. Across the six groups, these mean times range from 1.847 s (one-photon on sto p=0.5) to 9.573 s (Q1-role on long-range) for PHOTE, and from 15.848 s (CTAN on sto p=0.5) to 783.335 s (DyGFormer on sto p=0.9) for the temporal GNN baselines.

6.3 BABOON INTERACTION DATASET

We use the 13-node Guinea baboon contact network to assess whether measured photonic features can support temporal link prediction at a size compatible with the 24-mode processor considered here. The original contact data have a temporal resolution of 20 seconds; we aggregate them using 15-minute and 5-minute windows. We evaluate analytic features and sampled outputs from Quandela's Belenos simulator on the full test sets at both aggregation durations. Physical one-photon measurements also cover the full test set of the 15-minute aggregation. Table 3 shows that analytic one-photon achieves the highest mean AP at 15-minute aggregation, while analytic two-photon leads at 5-minute aggregation among the methods shown. At both aggregation durations, at least one non-trainable photonic encoding therefore achieves higher mean AP than the best evaluated temporal GNN. On the full 15-minute test set, the hardware one-photon result is close to the simulator result and the best temporal GNN, although its mean AP is lower than that of analytic one-photon. These results support the small-scale feasibility of using measured one-photon features throughout the test sequence. Comparable prediction performance does not imply equality of the optical output distributions. Appendix D describes the sampling protocol and the hardware specifications.

7 LIMITATIONS AND CONCLUSION

Limitations. We use 100-node synthetic graphs and nine primary real networks with 74 to 1,505 nodes; scalability to substantially larger networks remains untested. The optical implementation requires one mode per node. Most results rely on analytic probabilities or simulation. Hardware measurements cover full test snapshots of the baboon network's 15-minute sequence, but are limited to one-photon inputs. Evaluations on larger graphs and two-photon hardware experiments remain future work.

Conclusion. PHOTE separates non-trainable photonic structural encoding from classical temporal learning, supporting competitive link prediction with lower training time after feature precomputation than the evaluated temporal GNNs. The task-dependent strengths of identity-sensitive profiles and role summaries highlight the importance of which structural information an encoding retains. The hardware study provides a small-scale demonstration that measured one-photon features can support this prediction pipeline across the full test sequence.

AI USE STATEMENT

We used generative AI tools to assist with experimental code implementation and deployment, figure and table preparation, language polishing, and reviewing the manuscript and its mathematical claims. We have reviewed all AI-assisted work and take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 4 describes the encodings and temporal readout, and Section 5 specifies the experimental setup and evaluation metrics. Data preprocessing, benchmark construction, and data splits are described in Appendices A and B. Appendix D documents the simulator and hardware experiments, while Appendix F provides implementation details and the runtime measurement protocol. The complete proof of Proposition 1 is given in Appendix E.1.

REFERENCES

Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, pp. 333–342, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450306911. doi: 10.1145/1993636.1993682. URL https://doi.org/10.1145/1993636.1993682.

Lu Bai, Luca Rossi, Andrea Torsello, and Edwin R. Hancock. A quantum Jensen–Shannon graph kernel for unattributed graphs. *Pattern Recognition*, 48(2):344–355, 2015. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2014.03.028. URL https://www.sciencedirect.com/science/article/pii/S0031320314001265.

Scott D. Berry and Jingbo B. Wang. Two-particle quantum walks: Entanglement and graph isomorphism testing. *Phys. Rev. A*, 83:042317, Apr 2011. doi: 10.1103/PhysRevA.83.042317. URL https://link.aps.org/doi/10.1103/PhysRevA.83.042317.

Jinyin Chen, Xueke Wang, and Xuanheng Xu. GC-LSTM: Graph convolution embedded LSTM for dynamic link prediction, 2021. URL https://arxiv.org/abs/1812.04206.

William R Clements, Peter C Humphreys, Benjamin J Metcalf, W Steven Kolthammer, and Ian A Walmsley. Optimal design for universal multiport interferometers. *Optica*, 3(12):1460–1465, 2016.

Alireza Dizaji, Benedict Aaron Tjandra, Mehrab Hamidi, Shenyang Huang, and Guillaume Rabusseau. T-GRAB: A synthetic diagnostic benchmark for learning on temporal graphs. KDD '26, pp. 2628–2639, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722585. doi: 10.1145/3770854.3785699. URL https://doi.org/10.1145/3770854.3785699.

Brendan L Douglas and Jingbo B Wang. A classical approach to the graph isomorphism problem using quantum walks. *Journal of Physics A: Mathematical and Theoretical*, 41(7):075303, feb 2008. doi: 10.1088/1751-8113/41/7/075303. URL https://doi.org/10.1088/1751-8113/41/7/075303.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=wTTjnvGphYj.

Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1960.

Edward Farhi and Sam Gutmann. Quantum computation and decision trees. *Phys. Rev. A*, 58:915–928, Aug 1998. doi: 10.1103/PhysRevA.58.915. URL https://link.aps.org/doi/10.1103/PhysRevA.58.915.

Keisuke Fujii and Kohei Nakajima. Harnessing disordered-ensemble quantum dynamics for machine learning. *Phys. Rev. Appl.*, 8:024030, Aug 2017. doi: 10.1103/PhysRevApplied.8.024030. URL https://link.aps.org/doi/10.1103/PhysRevApplied.8.024030.

John King Gamble, Mark Friesen, Dong Zhou, Robert Joynt, and S. N. Coppersmith. Two-particle quantum walks applied to the graph isomorphism problem. *Phys. Rev. A*, 81:052313, May 2010. doi: 10.1103/PhysRevA.81.052313. URL https://link.aps.org/doi/10.1103/PhysRevA.81.052313.

Valeria Gelardi, Jeanne Godard, Dany Paleressompoulle, Nicolas Claidiere, and Alain Barrat. Measuring social networks in primates: wearable sensors versus direct observations. *Proceedings. Mathematical, Physical, and Engineering Sciences*, 476(2236):20190737, 2020.

Mark Goldsmith, Harto Saarinen, Guillermo García-Pérez, Joonas Malmi, Matteo AC Rossi, and Sabrina Maniscalco. Link prediction with continuous-time classical and quantum walks. *Entropy*, 25(5):730, 2023.

Alessio Gravina, Giulio Lovisotto, Claudio Gallicchio, Davide Bacciu, and Claas Grohnfeldt. Long range propagation on continuous-time dynamic graphs, 2024. URL https://arxiv.org/abs/2406.02740.

Nicolas Heurtel, Andreas Fyrillas, Grégoire de Gliniasty, Raphaël Le Bihan, Sébastien Malherbe, Marceau Pailhas, Eric Bertasi, Boris Bourdoncle, Pierre-Emmanuel Emeriau, Rawad Mezher, Luka Music, Nadia Belabas, Benoît Valiron, Pascale Senellart, Shane Mansfield, and Jean Senellart. Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing. *Quantum*, 7:931, February 2023. ISSN 2521-327X. doi: 10.22331/q-2023-02-21-931. URL https://doi.org/10.22331/q-2023-02-21-931.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

Chong-Ki Hong, Zhe-Yu Ou, and Leonard Mandel. Measurement of subpicosecond time intervals between two photons by interference. *Physical Review Letters*, 59(18):2044, 1987.

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. URL https://arxiv.org/abs/2307.01026.

Shenyang Huang, Farimah Poursafaei, Reihaneh Rabbany, Guillaume Rabusseau, and Emanuele Rossi. UTG: Towards a unified view of snapshot and event based models for temporal graphs, 2024. URL https://arxiv.org/abs/2407.12269.

Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017. URL https://arxiv.org/abs/1609.02907.

Pieter Kok, W. J. Munro, Kae Nemoto, T. C. Ralph, Jonathan P. Dowling, and G. J. Milburn. Linear optical quantum computing with photonic qubits. *Rev. Mod. Phys.*, 79:135–174, Jan 2007. doi: 10.1103/RevModPhys.79.135. URL https://link.aps.org/doi/10.1103/RevModPhys.79.135.

Haoyang Li, Yuming Xu, Yiming Li, Hanmo Liu, Darian Li, Chen Jason Zhang, Lei Chen, and Qing Li. When speed meets accuracy: an efficient and effective graph model for temporal link prediction. *Proc. VLDB Endow.*, 18(10):3396–3405, June 2025. ISSN 2150-8097. doi: 10.14778/3748191.3748203. URL https://doi.org/10.14778/3748191.3748203.

Zhan Li, Wuqing Yu, Yusen Wu, and Chuan Wang. CTQWformer: A CTQW-based transformer for graph classification, 2026. URL https://arxiv.org/abs/2605.09486.

Xiaodong Lu, Leilei Sun, Tongyu Zhu, and Weifeng Lv. Improving temporal link prediction via temporal walk matrix projection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Ti3ciyqlS3.

Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.*, 14 (11):2531–2560, November 2002. ISSN 0899-7667. doi: 10.1162/089976602760407955. URL https://doi.org/10.1162/089976602760407955.

Anmol Madan, Manuel Cebrian, Sai Moturu, Katayoun Farrahi, and Sandy Pentland. Sensing the "health state" of a community. *IEEE Pervasive Computing*, 11(4):36–45, October 2012. ISSN 1536-1268. doi: 10.1109/MPRV.2011.79. URL https://doi.org/10.1109/MPRV.2011.79.

Nicolas Maring, Andreas Fyrillas, Mathias Pont, Edouard Ivanov, Petr Stepanov, Nico Margaria, William Hease, Anton Pishchagin, Aristide Lemaître, Isabelle Sagnes, et al. A versatile single-photon-based quantum computing platform. *Nature Photonics*, 18(6):603–609, 2024.

Rawad Mezher, Ana Filipa Carvalho, and Shane Mansfield. Solving graph problems with single photons and linear optics. *Phys. Rev. A*, 108:032405, Sep 2023. doi: 10.1103/PhysRevA.108.032405. URL https://link.aps.org/doi/10.1103/PhysRevA.108.032405.

Radosław Michalski, Sebastian Palus, and Przemysław Kazienko. Matching organizational structure and social network extracted from email communication. In Witold Abramowicz (ed.), *Business Information Systems*, pp. 197–206, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21863-7.

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*, WSDM '17, pp. 601–610, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346757. doi: 10.1145/3018661.3018731. URL https://doi.org/10.1145/3018661.3018731.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. EvolveGCN: Evolving graph convolutional networks for dynamic graphs, 2019. URL https://arxiv.org/abs/1902.10191.

Farimah Poursafaei, Shenyang Huang, Kellin Pelrine, and Reihaneh Rabbany. Towards better evaluation for dynamic link prediction. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Xiaogang Qiang, Yizhi Wang, Shichuan Xue, Renyou Ge, Lifeng Chen, Yingwen Liu, Anqi Huang, Xiang Fu, Ping Xu, Teng Yi, Fufang Xu, Mingtang Deng, Jingbo B. Wang, Jasmin D. A. Meinecke, Jonathan C. F. Matthews, Xinlun Cai, Xuejun Yang, and Junjie Wu. Implementing graph-theoretic quantum algorithms on a silicon photonic quantum walk processor. *Science Advances*, 7(9):eabb8375, 2021. doi: 10.1126/sciadv.abb8375. URL https://www.science.org/doi/abs/10.1126/sciadv.abb8375.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs, 2020. URL https://arxiv.org/abs/2006.10637.

Luca Rossi, Andrea Torsello, and Edwin R. Hancock. Measuring graph similarity through continuous-time quantum walks and the quantum Jensen-Shannon divergence. *Phys. Rev. E*, 91: 022815, Feb 2015. doi: 10.1103/PhysRevE.91.022815. URL https://link.aps.org/doi/10.1103/PhysRevE.91.022815.

Kenneth Rudinger, John King Gamble, Mark Wellons, Eric Bach, Mark Friesen, Robert Joynt, and S. N. Coppersmith. Noninteracting multiparticle quantum random walks applied to the graph isomorphism problem for strongly regular graphs. *Phys. Rev. A*, 86:022334, Aug 2012. doi: 10.1103/PhysRevA.86.022334. URL https://link.aps.org/doi/10.1103/PhysRevA.86.022334.

Callum Schofield, Jingbo B. Wang, and Yuying Li. Quantum walk inspired algorithm for graph similarity and isomorphism. *Quantum Information Processing*, 19(9), August 2020. ISSN 1570-0755. URL https://doi.org/10.1007/s11128-020-02758-7.

Amauri H. Souza, Diego Mesquita, Samuel Kaski, and Vikas Garg. Provably expressive temporal graph networks, 2022. URL https://arxiv.org/abs/2209.15059.

Slimane Thabet, Romain Fouilland, Mehdi Djellabi, Igor Sokolov, Sachin Kasture, Louis-Paul Henry, and Loïc Henriet. Enhancing graph neural networks with quantum computed encodings, 2023. URL https://arxiv.org/abs/2310.20519.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018. URL https://arxiv.org/abs/1710.10903.

Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=KYPz4YsCPj.

Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs, 2020. URL https://arxiv.org/abs/2002.07962.

Lu Yi, Runlin Lei, Fengran Mo, Yanping Zheng, Zhewei Wei, and Yuhang Ye. Future link prediction without memory or aggregation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=kHDUCUqPEc.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library, 2023. URL https://arxiv.org/abs/2303.13047.

Lei Yu, Hongyang Chen, Jingsong Lv, and Linyao Yang. GQWformer: A quantum-based transformer for graph representation learning, 2024. URL https://arxiv.org/abs/2412.02285.

Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2020. doi: 10.1109/TITS.2019.2935152.

Table 4: Statistics of the real temporal interaction networks.

Dataset	Nodes	Edges	Unique Edges	Surprise	Granularity	Snapshots
Eu Dept1	309	35,490	1,938	0.185	1 day	517
Eu Dept2	162	25,254	1,045	0.240	1 day	521
Eu Dept3	89	8,147	973	0.341	1 day	436
Eu Dept4	142	24,917	833	0.266	1 day	509
Radoslaw	167	37,226	3,250	0.136	1 day	240
MIT Reality	96	29,165	2,539	0.016	1 day	233
SocialEvo	74	70,869	2,497	0.028	1 day	244
tgbn-trade	255	295,476	19,725	0.116	1 year	31
tgbn-genre	1,505	1,854,242	133,758	0.443	7 days	227
Baboons (15 min)	13	18,358	78	0.000	15 min	1,816
Baboons (5 min)	13	24,853	78	0.000	5 min	5,192

Table 5: Test average precision (AP) of two non-learned heuristics and the nine trained temporal-GNN baselines on nine interaction networks and the 13-node baboon contact network at 15-minute and 5-minute aggregation. Persistence and EdgeBank are deterministic and have no seed variation. The top three means in each row are marked **First**, Second and *Third*; tied entries share a rank.

	Non-learned		*Discrete-time GNNs*					*Continuous-time GNNs*			
network	Persistence	EdgeBank	GCN	GAT	GC-LSTM	TGCN	EvolveGCN	TGAT	TGN	DyGFormer	CTAN
Eu Dept1	0.096	0.073	0.097±0.010	0.093±0.005	*0.127*±0.002	0.113±0.020	0.071±0.003	0.131±0.003	0.104±0.006	**0.139**±0.027	0.108±0.002
Eu Dept2	0.121	0.091	0.165±0.003	0.142±0.006	0.188±0.006	**0.195**±0.011	0.104±0.010	0.130±0.007	0.128±0.006	0.163±0.011	*0.180*±0.011
Eu Dept3	0.045	0.045	*0.075*±0.007	0.071±0.003	0.081±0.004	0.049±0.008	0.019±0.004	*0.075*±0.007	0.059±0.006	**0.117**±0.012	0.062±0.020
Eu Dept4	0.126	0.104	0.145±0.004	0.134±0.007	0.146±0.010	0.140±0.011	0.097±0.002	0.165±0.010	0.153±0.021	**0.180**±0.003	*0.160*±0.018
Radoslaw	*0.180*	0.108	0.160±0.010	0.153±0.006	0.178±0.015	0.141±0.005	0.065±0.014	0.187±0.010	0.159±0.015	**0.235**±0.001	0.162±0.057
MIT Reality	0.247	0.100	*0.222*±0.032	0.174±0.017	**0.284**±0.009	0.191±0.024	0.117±0.037	0.195±0.019	0.142±0.004	0.158±0.033	0.148±0.061
SocialEvo	**0.355**	0.182	0.212±0.007	0.218±0.006	0.160±0.007	0.257±0.008	0.205±0.030	0.176±0.008	0.215±0.011	0.174±0.007	*0.221*±0.004
tgbn-trade	**0.853**	0.769	0.837±0.011	0.791±0.052	0.781±0.028	0.793±0.027	0.770±0.059	*0.803*±0.028	0.792±0.046	0.748±0.023	0.730±0.028
tgbn-genre	**0.259**	0.132	0.095±0.016	0.068±0.010	0.184±0.008	*0.174*±0.006	0.126±0.024	0.112±0.006	0.123±0.006	0.114±0.013	0.117±0.007
Baboons 15 min	0.240	0.125	0.311±0.009	0.299±0.009	**0.399**±0.006	0.382±0.005	0.185±0.010	0.364±0.007	0.354±0.005	0.338±0.015	*0.368*±0.009
Baboons 5 min	0.206	0.059	0.247±0.008	0.235±0.004	**0.311**±0.005	0.294±0.004	0.101±0.006	0.268±0.005	0.257±0.006	*0.273*±0.001	0.264±0.014

A Real-data statistics and detailed baseline results

A.1 Data preprocessing and dataset statistics

Table 4 lists the statistics of the nine interaction networks and both baboon aggregations.

Data preprocessing. For each of the nine interaction networks, we sort the interactions by time and divide them into non-overlapping windows of fixed duration. We use one-day windows for the email and proximity datasets, one-year windows for tgbn-trade, and one-week windows for tgbn-genre. Within each window, all repeated interactions between the same node pair are collapsed into a single undirected, unweighted edge. Edge directions and interaction weights are discarded where present. Each nonempty window becomes one graph snapshot. The aggregation duration follows the temporal resolution and prediction protocol of each data source rather than being forced to a common value. The tgbn-trade data are released as annual reports, so one year is their finest available temporal resolution. For tgbn-genre, the one-week window follows the original TGB task. We also report the surprise index (i.e., $\frac{|E_{\text{test}} \setminus E_{\text{train}}|}{|E_{\text{test}}|}$) defined in (Poursafaei et al., 2022), which computes the ratio of test edges that are not seen during training.

Baboon interaction dataset. The nine interaction networks require more optical modes than are available in the processor configuration considered here. To evaluate whether the complete pipeline remains effective when its photonic features are obtained from measurements on a real 24-mode quantum device, we use a proximity network with 13 nodes, comprising 28 days of face-to-face contact data among Guinea baboons (Gelardi et al., 2020). We aggregate the original 63,095 time-stamped contact records into snapshots using 15-minute and 5-minute windows, shown in the last two rows of Table 4. After removing empty windows, the resulting sequences contain 1,816 and 5,192 snapshots, respectively. The remaining snapshots are arranged in chronological order.

Data splits. The real interaction networks and Baboon interaction dataset use chronological 70%/15%/15% splits.

A.2 Detailed baseline results

Table 5 reports the available results for two non-learned heuristics and nine trained temporal GNNs on the nine interaction networks and both baboon aggregations.

Table 6: Test MRR on SocialEvo under the UTG protocol. Results under *Temporal GNNs (reported by UTG)* are taken from Table 2 of Huang et al. (2024), where EvolveGCN is called EGCNo; all other results are from our experiments with the same data split, negative samples and metric. EdgeBank denotes the time-window variant. Trained entries report means and standard deviations over five runs. The top three means among all methods shown are marked **First**, Second and *Third*.

	Non-learned		*Temporal GNNs (reported by UTG)*					*Temporal GNNs (our runs)*			
Network	Persistence	EdgeBank	GCN	GC-LSTM	EvolveGCN	TGN	DyGFormer	GAT	TGCN	TGAT	CTAN
SocialEvo	0.220	0.070	0.289±0.008	0.286±0.003	0.253±0.006	0.283±0.009	**0.366**±0.004	0.287±0.004	0.321±0.004	0.252±0.006	*0.350*±0.014

	Classical PE		*Quantum-walk encodings (PHOTE)*			
Network	LapPE	RWPE	one-photon	two-photon	Q1-role	Q2-role
SocialEvo	0.248±0.008	0.245±0.011	0.351±0.006	0.313±0.003	0.222±0.005	0.252±0.019

A.3 SUPPLEMENTARY INFORMATION ON SOCIALEVO

Repeated contacts and Persistence. SocialEvo records proximity contacts among students in a university dormitory. Contacts between the same residents often recur, which explains the strong Persistence result: predicting that a recently observed contact will recur is already useful on this dataset. Two-photon has a higher mean AP than every evaluated GNN and classical encoding, but not than Persistence (0.344 ± 0.010 vs. 0.355; Tables 1 and 5).

Evaluation under the UTG protocol. We additionally evaluate SocialEvo with the data split, negative samples and mean reciprocal rank (MRR) metric of UTG (Huang et al., 2024). MRR averages the reciprocal rank of the true destination when ranked against the sampled negative destinations of each test edge; higher values indicate better ranking of true links. The results for GCN, GC-LSTM, EvolveGCN, TGN and DyGFormer are those reported in Table 2 of Huang et al. (2024). We train GAT, TGCN, TGAT and CTAN, which are not reported there, and our six encoding pipelines, for which only the temporal readout and link scorer are trained, under the same protocol, and select every checkpoint by validation MRR. Table 6 reports these results separately from the AP-based experiments. One-photon has the highest mean among our encodings, 0.351 ± 0.006. This is level with CTAN (0.350 ± 0.014), below the 0.366 ± 0.004 that Huang et al. (2024) report for DyGFormer, and above all other methods in the table. These results further support the effectiveness of PHOTE under UTG's ranking-based evaluation.

A.4 LINK-PREDICTION DATASETS DERIVED FROM TGB

TGB (Huang et al., 2023) provides datasets for dynamic link-property prediction (`tgbl-*`) and dynamic node-property prediction (`tgbn-*`). We use `tgbn-trade` and `tgbn-genre`, the two node-property datasets with the fewest nodes in the original TGB benchmark, and adapt the two datasets for discrete-time link prediction.

Original datasets and tasks. `tgbn-trade` records international agricultural trade from 1986 to 2016. Each node represents a nation, and a directed edge from u to v records the total value of agricultural products traded from u to v in a given year. The dataset contains 255 nodes and 468,245 timestamped, weighted, directed edges.

`tgbn-genre` is a weighted, directed bipartite network between users and music genres. An edge from a user to a genre records a listening event, with a timestamp in seconds and a weight indicating the listened-to song's degree (percentage) of membership in that genre. The dataset contains 1,505 nodes (992 users and 513 genres) and 17,858,395 timestamped, weighted edges.

In TGB, both datasets are used for *node affinity prediction*. For a target node u, the label is a vector of accumulated edge weights toward candidate destination nodes over a fixed future period, normalized to sum to one. For `tgbn-trade`, this represents the distribution of a nation's agricultural trade across partner nations in the next year. For `tgbn-genre`, it represents the normalized distribution of a user's genre-weighted listening activity over the next week.

Conversion to snapshot datasets. We retain the full node sets and construct snapshot graphs from the original timestamped edge records in three steps. (i) *Snapshots.* We group `tgbn-trade` edges by calendar year, producing 31 snapshots. For `tgbn-genre`, we use consecutive, non-overlapping seven-day windows beginning at the earliest edge timestamp, producing 227 snapshots. All resulting windows are nonempty. (ii) *Links.* Two distinct nodes are linked in a snapshot if at least one edge between them, in either direction, occurs within its window. We discard edge weights and directions,

collapse repeated interactions into a single link, and remove self-loops. Each snapshot is therefore an undirected, unweighted simple graph. For `tgbn-genre`, every observed link joins a user to a genre, so the snapshot graphs remain bipartite. (iii) *Splits.* We split snapshots chronologically into training, validation, and test sets using an approximately $70\%/15\%/15\%$ allocation. This gives $21/5/5$ yearly snapshots for `tgbn-trade` and $159/34/34$ weekly snapshots for `tgbn-genre`.

Link-prediction task. To predict the links in snapshot t, a model uses only the preceding snapshots $1, \ldots, t-1$. At evaluation, the candidate set contains all unordered pairs of distinct nodes. Pairs linked in snapshot t are positives, and all remaining pairs are negatives; no negative sampling is used for evaluation. We compute average precision (AP) separately for each test snapshot and report the mean over the test snapshots.

For `tgbn-genre`, the candidate set is not restricted to user–genre pairs: user–user and genre–genre pairs are also scored and are always negatives. Training negatives are likewise sampled over the full node set without enforcing the bipartite constraint. Thus, unlike TGB's node-affinity task, our task predicts whether a link exists in the next snapshot rather than the normalized distribution of future edge weights.

B RESULTS FOR INDIVIDUAL SYNTHETIC SETTINGS

B.1 BENCHMARK CONSTRUCTION AND DATA SPLITS

We describe the synthetic benchmarks from T-GRAB (Dizaji et al., 2026) used in our evaluation. The benchmark organizes these tasks into three main families: *Periodicity*, *Cause-and-effect*, and *Long-range memory*. Periodicity is further divided into three subtasks: `Memorization`, `Counting`, and `Stochastic periodicity`. We treat these three subtasks separately, giving five task families in total.

Periodicity. We use the benchmark notation $\mathrm{det}(k, n)$ for deterministic periodicity and $\mathrm{sto}(k, n)$ for stochastic periodicity, where $k, n \in \mathbb{N}$. The sequence cycles through k fixed graphs ($\mathrm{det}(k, n)$) or graph-generating distributions ($\mathrm{sto}(k, n)$). Each one repeats n consecutive times, giving kn snapshots per cycle. We consider three subtasks:

- `Memorization`, $\mathrm{det}(k, 1)$: We generate k fixed Erdős–Rényi (ER) graphs (Erdős & Rényi, 1960), each with edge probability 0.01. In one cycle, we present each snapshot once sequentially. We use $k \in \{2, 4, 8, 16, 32, 64, 128, 256\}$. Increasing k enlarges the set of graph structures that the model must distinguish and remember.
- `Counting`, $\mathrm{det}(2, n)$: We generate two fixed ER graphs. In one cycle, we repeat the first graph for n times, then switch to the second graph and also repeat n times. We use $n \in \{2, 4, 8, 16, 32, 64, 128\}$. Larger values of n require the model to track longer repetitions before predicting when the graph will switch.
- `Stochastic periodicity`, $\mathrm{sto}(k, 1)$: We define k stochastic block model (SBM) (Holland et al., 1983) distributions. In one cycle, we sample one graph from each of the k distributions sequentially. Each distribution assigns the 100 nodes to three communities of sizes $(34, 33, 33)$ and uses an inter-community edge probability of 0.01, but differs in how the nodes are assigned. We consider intra-community edge probabilities $p = 0.9$ and $p = 0.5$, with the latter producing the more difficult setting, and use $k \in \{2, 4, 8, 16, 32, 64\}$. This subtask tests whether a model can identify recurring stochastic structures rather than memorize fixed graphs.

Cause-and-effect, $\mathbf{CE}(\boldsymbol{\ell})$**.** One node is designated as the memory node. At each snapshot, we sample a new ER graph over the remaining 99 nodes. At time t, the memory node connects to the endpoints of the edges that appeared at time $t - \ell$. We use $\ell \in \{1, 2, 4, 8, 16, 32\}$. Increasing ℓ raises the temporal memory requirement because the model must retain edge information over a longer delay.

Long-range memory, $\mathbf{LR}(\boldsymbol{\ell}, \boldsymbol{d})$**.** We designate one fixed source node and one fixed memory node. At each snapshot, we uniformly sample nodes to construct three new paths of length d from the source node. The paths are node-disjoint except for the source node. After ℓ snapshots, the memory node connects to the end nodes of these paths. We use $\ell \in \{1, 4, 16, 32\}$ and $d \in \{1, 2, 4, 8, 16\}$. This task jointly tests temporal and spatial reasoning: ℓ determines how long the path information must be retained, while d determines how far it must be propagated across the graph.

Together, we produce 53 benchmark settings.

Data splits. The periodicity task contains 48 complete periods. We split them chronologically into 40 periods for training, four for validation, and four for testing. A memorization or stochastic period contains k snapshots, whereas a counting period contains $2n$ snapshots. The long-range and cause-and-effect tasks contain 4000 snapshots plus the task delay (i.e., ℓ in both tasks) and are split chronologically in proportions $80\%/10\%/10\%$. The real interaction networks and Baboon interaction dataset use chronological $70\%/15\%/15\%$ splits.

B.2 RESULT REPORTING AND LOW-PERFORMING RUNS

Tables 7–12 report every method on all 53 synthetic settings. Appendix C presents the same results as curves grouped by task family (Figures 3–7).

For the temporal-GNN baselines and non-trainable encoding pipelines, we also report configurations in which one or two of the three seeds produce low-performing runs, identified post hoc. A seed

Table 7: Test F_1 for memorization $\det(k, 1)$ at M=100. Each entry is the mean and standard deviation over three seeds, computed over all node pairs in each test snapshot and then averaged over the test split. The top three means in each column are marked **First**, Second and *Third*; tied entries share a rank.

	memorization $\det(k, 1)$							
	k=2	k=4	k=8	k=16	k=32	k=64	k=128	k=256
Non-learned baselines								
Persistence	0.000±0.000	0.010±0.000	0.010±0.000	0.008±0.000	0.010±0.000	0.011±0.000	0.009±0.000	0.010±0.000
EdgeBank	0.665±0.000	0.404±0.000	0.227±0.000	0.127±0.000	0.070±0.000	0.042±0.000	0.027±0.000	0.021±0.000
Discrete-time GNNs								
GCN	0.830±0.003	0.802±0.003	0.763±0.008	0.726±0.021	0.674±0.004	0.585±0.002	0.411±0.005	0.189±0.014
GAT	0.913±0.027	0.904±0.011	0.866±0.011	0.792±0.013	0.717±0.002	0.596±0.006	0.415±0.002	0.187±0.019
GC-LSTM	0.985±0.004	0.965±0.003	0.968±0.011	0.976±0.003	0.970±0.007	0.957±0.003	0.875±0.004	0.669±0.011
TGCN	**0.995**±0.002	0.970±0.003	0.945±0.022	0.967±0.004	0.964±0.015	0.926±0.001	0.800±0.021	0.462±0.002
EvolveGCN	0.101±0.001	0.191±0.065	0.101±0.012	0.140±0.008	0.138±0.041	0.028±0.007	0.020±0.000	0.020±0.000
Continuous-time GNNs								
TGAT	0.949±0.006	0.941±0.062	0.937±0.002	0.950±0.015	0.972±0.002	0.940±0.019	0.486±0.307	0.341±0.018
TGN	0.913±0.011	0.932±0.021	0.946±0.011	0.924±0.003	0.856±0.014	0.725±0.019	0.525±0.047	0.203±0.053
DyGFormer	0.947±0.006	0.952±0.015	0.964±0.002	0.953±0.009	0.863±0.079	0.931±0.014	0.936±0.008	*0.865*±0.026
CTAN	0.827±0.037	0.431±0.164	0.644±0.028	0.701±0.024	0.619±0.114	0.626±0.043	0.497±0.069	0.112±0.082
Quantum-walk encodings (PHoTE)								
one-photon	0.960±0.016	*0.980*±0.012	0.989±0.005	**0.996**±0.003	*0.995*±0.003	0.992±0.001	0.990±0.008	**0.998**±0.000
Q1-role	0.713±0.036	0.955±0.009	**0.997**±0.003	*0.991*±0.007	**0.998**±0.001	0.980±0.018	0.976±0.041	0.673±0.566
two-photon	0.935±0.051	**0.991**±0.006	*0.987*±0.004	*0.991*±0.005	0.996±0.001	**0.997**±0.001	**0.998**±0.000	0.672±0.565
Q2-role	0.722±0.028	0.937±0.019	*0.987*±0.009	0.993±0.003	0.992±0.002	0.990±0.005	*0.989*±0.003	0.664±0.558
Classical positional encodings								
LapPE	*0.972*±0.005	0.990±0.006	0.983±0.005	0.986±0.008	0.673±0.563	0.671±0.564	0.672±0.565	0.672±0.565
RWPE	0.609±0.034	0.890±0.062	0.982±0.015	0.987±0.008	0.992±0.003	*0.991*±0.002	0.978±0.002	0.960±0.012

counts as low-performing when it recovers no more than a quarter of the gap between the EdgeBank score and the best seed in the same configuration. Configurations whose best seed is less than 0.20 above the EdgeBank score are not counted. Each tuple below gives the test F_1 scores for the three seeds. All three runs remain included in the reported means and standard deviations.

Memorization. Table 7 shows that the one-photon encoding maintains high mean scores across all tested cycle lengths, ranging from 0.960 to 0.998. The two-photon encoding scores between 0.987 and 0.998 for $k \in \{4, 8, 16, 32, 64, 128\}$, but its means are lower at $k = 2$ and 256. At $k = 2$, TGCN has the highest mean, 0.995, compared with 0.960 for one-photon and 0.935 for two-photon. The role summaries also perform less well at this shortest cycle, with Q1-role and Q2-role scoring 0.713 and 0.722. Both improve at longer cycles, reaching at least 0.976 at every tested $k \in \{8, 16, 32, 64, 128\}$.

Several trained GNNs have lower scores at longer cycles. At $k = 256$, GC-LSTM scores 0.669, while DyGFormer has the highest mean among the trained GNNs at 0.865. One-photon leads overall with 0.998, and RWPE scores 0.960. However, not all quantum-walk encodings maintain similarly high means: two-photon, Q1-role, and Q2-role score 0.672, 0.673, and 0.664, respectively. These three configurations each contain one low-performing run, while the other two runs score at least 0.975.

Specifically, at $\det(256, 1)$, the three fixed seeds yield $(1.000, 0.020, 0.999)$ for Q1-role, $(0.020, 0.998, 0.998)$ for two-photon, and $(0.020, 0.998, 0.975)$ for Q2-role. LapPE also contains one low-performing run at each of four cycle lengths: its three-seed scores are $(0.022, 0.998, 0.998)$ at $\det(32, 1)$, $(0.999, 0.995, 0.020)$ at $\det(64, 1)$, $(0.020, 1.000, 0.998)$ at $\det(128, 1)$, and $(0.998, 0.997, 0.020)$ at $\det(256, 1)$. CTAN contains two low-performing runs at $\det(4, 1)$, with scores $(0.311, 0.619, 0.364)$, while TGAT contains one at $\det(128, 1)$, with scores $(0.823, 0.411, 0.222)$. Thus, near-perfect performance in individual runs does not imply consistently high performance across the three fixed seeds.

Counting. Average F_1 and change-point F_1 measure different behavior. The former includes snapshots within a repeated block, whereas the latter evaluates only the transitions between the two graphs. Each graph is repeated for n consecutive snapshots.

This distinction is visible in Table 8. Persistence has the highest average score at $n = 32$, 64, and 128, despite scoring zero at every change point. LapPE leads on average F_1 at $n = 2$ and 16, while DyGFormer leads at $n = 4$ and 8. On the change-point metric, LapPE leads at $n = 2$; DyGFormer leads at $n = 4$ and 8; and one-photon leads at $n = 16$, 32, and 64. At $n = 128$, EdgeBank has

Table 8: Test F_1 for counting $\det(2, n)$. The upper block averages F_1 over all test snapshots; the lower block reports F_1 on the eight snapshots where the active graph changes. Entries are means and standard deviations over three seeds. Persistence has average $F_1 = 1 - 1/n$ and change-point $F_1 = 0$; EdgeBank scores 0.665 under both metrics. The top three means in each column of each block are marked **First**, Second and *Third*; tied entries share a rank.

	average F_1						
	n=2	n=4	n=8	n=16	n=32	n=64	n=128
Non-learned baselines							
Persistence	0.500±0.000	0.750±0.000	0.875±0.000	0.938±0.000	**0.969**±0.000	**0.984**±0.000	**0.992**±0.000
EdgeBank	0.665±0.000	0.665±0.000	0.665±0.000	0.665±0.000	0.665±0.000	0.665±0.000	0.665±0.000
Discrete-time GNNs							
GCN	0.595±0.001	0.597±0.000	0.597±0.002	0.614±0.002	0.660±0.005	0.763±0.002	0.824±0.001
GAT	0.647±0.013	0.643±0.004	0.654±0.009	0.654±0.011	0.678±0.006	0.745±0.012	0.840±0.002
GC-LSTM	0.901±0.010	0.907±0.009	0.921±0.014	0.918±0.022	0.900±0.006	0.863±0.006	0.872±0.013
TGCN	*0.936*±0.014	0.898±0.014	0.911±0.009	0.909±0.006	0.919±0.020	0.860±0.024	0.788±0.050
EvolveGCN	0.184±0.067	0.251±0.009	0.325±0.060	0.567±0.174	0.515±0.134	0.422±0.134	0.279±0.110
Continuous-time GNNs							
TGAT	0.928±0.025	0.926±0.017	0.875±0.027	*0.943*±0.021	0.867±0.033	0.866±0.037	*0.938*±0.058
TGN	0.907±0.014	0.899±0.008	0.912±0.025	0.746±0.131	0.709±0.213	0.593±0.132	0.749±0.132
DyGFormer	0.928±0.003	**0.997**±0.004	**0.994**±0.001	0.966±0.007	0.876±0.018	0.906±0.014	0.990±0.002
CTAN	0.797±0.022	0.799±0.027	0.651±0.084	0.670±0.074	0.693±0.070	0.637±0.143	0.835±0.049
*Quantum-walk encodings (*PHOTE*)*							
one-photon	0.934±0.008	0.939±0.005	0.940±0.005	0.922±0.008	0.923±0.014	*0.935*±0.012	0.875±0.029
Q1-role	0.701±0.023	0.717±0.034	0.741±0.003	0.726±0.003	0.727±0.018	0.712±0.021	0.726±0.026
two-photon	0.964±0.026	0.984±0.010	*0.971*±0.001	0.928±0.040	*0.939*±0.020	0.893±0.046	0.883±0.040
Q2-role	0.749±0.003	0.747±0.009	0.740±0.005	0.745±0.003	0.737±0.012	0.706±0.020	0.745±0.018
Classical positional encodings							
LapPE	**0.978**±0.017	*0.974*±0.014	0.973±0.004	**0.972**±0.012	0.953±0.004	0.945±0.024	0.906±0.023
RWPE	0.655±0.008	0.668±0.009	0.669±0.001	0.669±0.003	0.647±0.025	0.641±0.080	0.629±0.032

	change-point F_1						
	n=2	n=4	n=8	n=16	n=32	n=64	n=128
Non-learned baselines							
Persistence	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000
EdgeBank	0.665±0.000	0.665±0.000	0.665±0.000	0.665±0.000	*0.665*±0.000	0.665±0.000	**0.665**±0.000
Discrete-time GNNs							
GCN	0.594±0.000	0.593±0.001	0.591±0.003	0.557±0.029	0.509±0.068	0.407±0.010	0.293±0.002
GAT	0.647±0.012	0.643±0.004	0.651±0.006	0.649±0.009	0.603±0.012	0.494±0.024	0.305±0.003
GC-LSTM	0.904±0.010	0.864±0.005	0.707±0.032	0.658±0.006	0.644±0.011	0.590±0.041	0.577±0.028
TGCN	0.932±0.016	0.778±0.035	0.685±0.012	0.654±0.002	0.614±0.027	0.539±0.100	0.495±0.021
EvolveGCN	0.145±0.042	0.172±0.008	0.183±0.023	0.145±0.049	0.151±0.013	0.152±0.036	0.102±0.031
Continuous-time GNNs							
TGAT	*0.942*±0.031	0.911±0.010	0.860±0.026	0.760±0.049	0.588±0.109	0.598±0.017	0.395±0.044
TGN	0.905±0.018	0.865±0.018	0.758±0.022	0.568±0.076	0.643±0.001	0.347±0.232	0.175±0.068
DyGFormer	0.932±0.006	**0.994**±0.011	**0.981**±0.001	*0.808*±0.006	0.657±0.005	0.530±0.074	0.411±0.049
CTAN	0.783±0.018	0.692±0.053	0.596±0.075	0.597±0.048	0.583±0.027	0.496±0.064	0.434±0.036
*Quantum-walk encodings (*PHOTE*)*							
one-photon	0.936±0.009	0.916±0.015	0.889±0.013	**0.871**±0.014	**0.756**±0.066	**0.688**±0.049	0.632±0.029
Q1-role	0.708±0.033	0.660±0.051	0.648±0.018	0.481±0.026	0.480±0.033	0.462±0.008	0.449±0.008
two-photon	0.964±0.025	0.970±0.023	*0.898*±0.010	0.719±0.075	0.646±0.014	0.644±0.010	0.615±0.032
Q2-role	0.749±0.004	0.734±0.017	0.687±0.017	0.614±0.024	0.509±0.067	0.416±0.045	0.391±0.055
Classical positional encodings							
LapPE	**0.976**±0.015	*0.949*±0.023	0.900±0.011	0.847±0.056	0.716±0.049	*0.652*±0.024	*0.621*±0.033
RWPE	0.655±0.007	0.645±0.023	0.583±0.005	0.506±0.030	0.402±0.042	0.406±0.021	0.358±0.053

the highest change-point score, approximately 0.665. One-photon has the highest mean among the trainable pipelines at this block length, 0.632, followed by LapPE at 0.621 and two-photon at 0.615. The strongest trained GNN, GC-LSTM, scores 0.577. Both identity-sensitive photonic encodings have higher means than both role summaries at every tested n under both metrics.

TGN contains one low-performing run at $\det(2, 32)$ and two at $\det(2, 128)$, with three-seed average F_1 scores $(0.757, 0.894, 0.476)$ and $(0.699, 0.899, 0.649)$, respectively.

In this counting task ($\det(2, n)$), two fixed graphs A and B alternate, with each graph repeated for n consecutive snapshots. Let E_A and E_B denote their edge sets. The graph pair used in our benchmark has no shared edges:

$$E_A \cap E_B = \varnothing, \qquad |E_A| = 50, \qquad |E_B| = 60.$$

For a predicted edge set S and the actual edge set E of the next snapshot, the score is

$$F_1(S, E) = \frac{2|S \cap E|}{|S| + |E|},$$

where $|S \cap E|$ is the number of correctly predicted edges. We average this score over test snapshots. The change-point score averages only the snapshots at which the graph switches from A to B or from B to A.

Persistence predicts that the next snapshot has the same edges as the current one. Its score is 1 when the graph remains unchanged and 0 when it switches, because E_A and E_B have no shared edges. Each complete cycle contains $2n$ prediction steps, with $2(n-1)$ unchanged steps and two switches. Its average score is therefore

$$F_1^{\text{Persistence}} = \frac{2(n-1) \times 1 + 2 \times 0}{2n} = 1 - \frac{1}{n}.$$

Its change-point F_1 is exactly zero. Thus, the average score increases with n even though Persistence never correctly predicts a switch.

EdgeBank predicts all edges observed so far. After both graphs have appeared, it always predicts $E_A \cup E_B$, which contains $50 + 60 = 110$ edges. This prediction includes every actual edge but also includes the edges belonging only to the other graph. The scores for the two possible target snapshots are

$$F_1(E_A \cup E_B, E_A) = \frac{2 \times 50}{110 + 50} = \frac{5}{8},$$
$$F_1(E_A \cup E_B, E_B) = \frac{2 \times 60}{110 + 60} = \frac{12}{17}.$$

Because the two graphs occur equally often, EdgeBank's average score is

$$F_1^{\text{EdgeBank}} = \frac{1}{2}\left(\frac{5}{8} + \frac{12}{17}\right) \approx 0.665.$$

The two switch directions also occur equally often, so the change-point score has the same value. Both averages are independent of n and round to the reported value of 0.665.

Consequently, a high average score does not necessarily indicate that a method predicts the transitions successfully.

Stochastic periodicity. Table 9 shows that two-photon attains or ties the highest mean in all 12 settings at the reported precision. Its mean scores range from 0.937 to 0.938 when $p = 0.9$ and from 0.654 to 0.662 when $p = 0.5$. The corresponding one-photon ranges are 0.933–0.938 and 0.654–0.662. Both encodings remain close to the expected-count F_1 reference values from Proposition 1, 0.9372 and 0.6576, respectively.

At the longest tested cycle, $k = 64$, one-photon and two-photon score 0.933 and 0.937 when $p = 0.9$. Both score 0.657 when $p = 0.5$. GC-LSTM is the strongest trained GNN at this cycle length for both densities, scoring 0.912 and 0.516, respectively. The role summaries have substantially lower means at $k = 64$: Q1-role and Q2-role score 0.420 and 0.417 when $p = 0.9$, and 0.265 and 0.288 when $p = 0.5$. Thus, the strong long-cycle performance of the identity-sensitive encodings is not shared by their role summaries.

For $p = 0.9$, DyGFormer and LapPE each contain one low-performing run at $\mathrm{sto}(8, 1)$, with three-seed scores $(0.895, 0.437, 0.895)$ and $(0.936, 0.928, 0.421)$, respectively. TGAT contains two low-performing runs at $\mathrm{sto}(32, 1)$, with scores $(0.462, 0.462, 0.878)$.

For $p = 0.5$, GAT and Q1-role each contain one low-performing run at $\mathrm{sto}(2, 1)$, with three-seed scores $(0.365, 0.473, 0.614)$ and $(0.335, 0.662, 0.639)$, respectively. CTAN contains two low-performing runs at $\mathrm{sto}(8, 1)$, with scores $(0.299, 0.530, 0.342)$. TGAT contains two low-performing runs at each of $\mathrm{sto}(32, 1)$ and $\mathrm{sto}(64, 1)$, with scores $(0.631, 0.285, 0.289)$ and $(0.495, 0.291, 0.287)$, respectively.

Table 9: Test F_1 for stochastic periodicity $\mathrm{sto}(k, 1)$ at both densities. Each entry is the mean and standard deviation over three seeds, computed over all node pairs in each test snapshot and then averaged over the test split. Proposition 1 gives the optimal expected-count F_1 reference values $F_1^\star = 0.9372$ for p=0.9 and 0.6576 for p=0.5. The projected one-photon and two-photon encodings remain within a few 10^{-3} of these values for every k. The top three means in each column of each block are marked **First**, Second and *Third*; tied entries share a rank, so a column with several tied first places marks no second or third place.

	stochastic p=0.9					
	k=2	k=4	k=8	k=16	k=32	k=64
Non-learned baselines						
Persistence	0.288±0.000	0.294±0.000	0.296±0.000	0.300±0.000	0.299±0.000	0.300±0.000
EdgeBank	0.543±0.000	0.475±0.000	0.463±0.000	0.462±0.000	0.462±0.000	0.462±0.000
Discrete-time GNNs						
GCN	0.426±0.002	0.419±0.012	0.414±0.005	0.430±0.002	0.441±0.005	0.443±0.007
GAT	0.936±0.002	0.642±0.044	0.558±0.013	0.508±0.002	0.484±0.004	0.469±0.003
GC-LSTM	**0.938**±0.000	**0.937**±0.000	**0.938**±0.000	**0.937**±0.000	0.936±0.000	*0.912*±0.021
TGCN	**0.938**±0.000	0.933±0.002	0.927±0.001	0.899±0.004	0.812±0.035	0.634±0.009
EvolveGCN	0.462±0.000	0.461±0.000	0.463±0.000	0.462±0.000	0.462±0.000	0.462±0.000
Continuous-time GNNs						
TGAT	**0.938**±0.000	**0.937**±0.000	0.934±0.002	0.931±0.002	0.600±0.240	0.462±0.000
TGN	0.936±0.003	0.935±0.002	0.921±0.006	0.759±0.100	0.518±0.087	0.441±0.010
DyGFormer	0.921±0.002	0.894±0.015	0.742±0.264	0.527±0.058	0.462±0.000	0.462±0.000
CTAN	0.937±0.001	0.818±0.078	0.584±0.079	0.461±0.002	0.462±0.000	0.462±0.000
*Quantum-walk encodings (*PHoTE*)*						
one-photon	**0.938**±0.000	**0.937**±0.000	**0.938**±0.000	*0.936*±0.001	*0.934*±0.004	0.933±0.003
Q1-role	**0.938**±0.000	0.835±0.062	0.422±0.006	0.437±0.009	0.448±0.011	0.420±0.001
two-photon	**0.938**±0.000	**0.937**±0.000	**0.938**±0.000	**0.937**±0.000	**0.937**±0.000	**0.937**±0.000
Q2-role	0.541±0.136	0.454±0.012	0.463±0.000	0.448±0.007	0.443±0.016	0.417±0.001
Classical positional encodings						
LapPE	0.936±0.001	0.922±0.023	0.762±0.295	0.403±0.014	0.407±0.004	0.405±0.008
RWPE	0.462±0.000	0.437±0.023	0.447±0.007	0.441±0.022	0.452±0.013	0.432±0.010

	stochastic p=0.5					
	k=2	k=4	k=8	k=16	k=32	k=64
Non-learned baselines						
Persistence	0.165±0.000	0.168±0.000	0.165±0.000	0.170±0.000	0.168±0.000	0.169±0.000
EdgeBank	0.349±0.000	0.299±0.000	0.289±0.000	0.291±0.000	0.289±0.000	0.291±0.000
Discrete-time GNNs						
GCN	0.268±0.003	0.266±0.003	0.263±0.002	0.276±0.005	0.273±0.006	0.269±0.005
GAT	0.484±0.125	0.322±0.011	0.290±0.008	0.292±0.005	0.269±0.002	0.266±0.003
GC-LSTM	**0.662**±0.000	**0.657**±0.000	*0.653*±0.001	*0.646*±0.002	*0.616*±0.012	*0.516*±0.049
TGCN	0.661±0.000	0.582±0.008	0.330±0.018	0.288±0.004	0.285±0.007	0.282±0.006
EvolveGCN	0.292±0.000	0.290±0.000	0.288±0.000	0.291±0.000	0.289±0.000	0.291±0.000
Continuous-time GNNs						
TGAT	0.628±0.006	0.651±0.003	0.629±0.009	0.642±0.003	0.401±0.198	0.358±0.119
TGN	0.660±0.001	0.649±0.006	0.588±0.031	0.431±0.019	0.296±0.013	0.288±0.001
DyGFormer	0.625±0.008	0.286±0.003	0.287±0.000	0.291±0.000	0.289±0.000	0.291±0.000
CTAN	**0.662**±0.000	0.503±0.111	0.390±0.123	0.316±0.028	0.289±0.000	0.291±0.000
*Quantum-walk encodings (*PHoTE*)*						
one-photon	**0.662**±0.000	**0.657**±0.000	**0.654**±0.000	0.657±0.000	0.654±0.000	**0.657**±0.000
Q1-role	0.545±0.183	0.277±0.013	0.288±0.000	0.291±0.000	0.268±0.003	0.265±0.001
two-photon	**0.662**±0.000	**0.657**±0.000	**0.654**±0.000	**0.658**±0.000	**0.655**±0.000	**0.657**±0.000
Q2-role	0.639±0.036	0.467±0.035	0.288±0.000	0.286±0.009	0.278±0.001	0.288±0.005
Classical positional encodings						
LapPE	0.657±0.001	0.653±0.001	0.636±0.012	0.266±0.004	0.259±0.002	0.266±0.000
RWPE	0.287±0.010	0.274±0.009	0.266±0.010	0.277±0.009	0.272±0.005	0.274±0.004

Long-range memory. Tables 10 and 11 show that role summaries achieve high scores on many combinations of delay and path length, although several configurations vary substantially across seeds. At the reported precision, Q1-role attains or ties the highest mean in 10 of the 20 long-range settings, and Q2-role does so in 9. Q2-role has a higher mean than one-photon in 19 settings. The exception is $\mathrm{LR}(1, 1)$, where its mean is 0.704, compared with 1.000 for one-photon.

The advantage of role summaries is particularly clear at $\mathrm{LR}(16, 16)$, where Q1-role and Q2-role score 0.940 and 0.947, respectively. The discrete-time GNNs score between 0.057 and 0.059, while TGAT has the highest continuous-time GNN score at 0.441. The ranking is not uniform across all settings, however. At delay $\ell = 32$, TGAT leads or ties for the lead at path lengths $d = 1$, 2, and 4. Q1-role leads at $d = 8$ with 0.894, while Q2-role leads at $d = 16$ with 0.721.

Table 10: Test F_1 for long-range memory $\mathrm{LR}(\ell, d)$ at lags ℓ=1 and 4 on M=100 graphs. The metric is computed on edges incident to the memory node. Entries are means and standard deviations over three seeds. The top three means in each column of each block are marked **First**, Second and *Third*; tied entries share a rank, so a column with several tied first places marks no second or third place.

	long-range $\mathrm{LR}(1, d)$				
	d=1	d=2	d=4	d=8	d=16
Non-learned baselines					
Persistence	0.033±0.000	0.035±0.000	0.035±0.000	0.032±0.000	0.028±0.000
EdgeBank	0.059±0.000	0.059±0.000	0.059±0.000	0.059±0.000	0.059±0.000
Discrete-time GNNs					
GCN	**1.000**±0.000	0.967±0.014	0.630±0.006	0.778±0.039	0.519±0.004
GAT	**1.000**±0.000	0.995±0.002	0.853±0.064	0.759±0.045	0.852±0.133
TGCN	0.999±0.002	**1.000**±0.000	*0.999*±0.001	*0.998*±0.002	0.999±0.001
GC-LSTM	**1.000**±0.000	**1.000**±0.000	0.938±0.017	0.989±0.006	0.979±0.014
EvolveGCN	0.120±0.010	0.119±0.005	0.138±0.018	0.253±0.012	0.059±0.000
Continuous-time GNNs					
TGAT	**1.000**±0.000	**1.000**±0.000	0.990±0.016	0.814±0.160	0.298±0.036
TGN	0.667±0.483	0.995±0.004	0.886±0.081	0.605±0.264	0.135±0.081
DyGFormer	**1.000**±0.000	**1.000**±0.000	0.994±0.004	*0.998*±0.001	0.999±0.001
CTAN	0.017±0.027	0.058±0.001	0.059±0.000	0.059±0.000	0.058±0.001
Quantum-walk encodings (PHoTE)					
one-photon	**1.000**±0.000	0.929±0.012	0.762±0.040	0.759±0.013	0.763±0.026
Q1-role	**1.000**±0.000	0.999±0.000	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000
two-photon	**1.000**±0.000	0.982±0.006	0.862±0.007	0.801±0.192	0.354±0.510
Q2-role	0.704±0.513	**1.000**±0.000	**1.000**±0.000	0.999±0.001	0.997±0.005
Classical positional encodings					
LapPE	0.938±0.009	0.971±0.004	0.959±0.004	0.899±0.005	0.756±0.006
RWPE	0.993±0.004	0.998±0.001	*0.999*±0.000	0.994±0.001	0.515±0.387

	long-range $\mathrm{LR}(4, d)$				
	d=1	d=2	d=4	d=8	d=16
Non-learned baselines					
Persistence	0.033±0.000	0.035±0.000	0.035±0.000	0.032±0.000	0.028±0.000
EdgeBank	0.059±0.000	0.059±0.000	0.059±0.000	0.059±0.000	0.059±0.000
Discrete-time GNNs					
GCN	0.059±0.001	0.059±0.001	0.058±0.000	0.058±0.000	0.058±0.001
GAT	0.059±0.000	0.059±0.001	0.059±0.000	0.059±0.001	0.059±0.001
TGCN	0.577±0.298	0.858±0.030	0.438±0.019	0.177±0.013	0.064±0.002
GC-LSTM	0.983±0.008	0.551±0.089	0.214±0.013	0.102±0.002	0.061±0.001
EvolveGCN	0.059±0.000	0.059±0.000	0.058±0.001	0.058±0.001	0.059±0.000
Continuous-time GNNs					
TGAT	**1.000**±0.000	**1.000**±0.000	0.935±0.006	0.063±0.003	0.127±0.018
TGN	0.896±0.130	0.997±0.004	0.864±0.103	0.808±0.249	0.561±0.434
DyGFormer	0.993±0.003	0.932±0.035	0.863±0.034	0.821±0.126	0.653±0.280
CTAN	0.033±0.030	0.056±0.003	0.059±0.000	0.059±0.000	0.058±0.002
Quantum-walk encodings (PHoTE)					
one-photon	0.998±0.003	0.936±0.001	0.715±0.013	0.715±0.042	0.692±0.025
Q1-role	**1.000**±0.000	**1.000**±0.000	**1.000**±0.001	**1.000**±0.001	**1.000**±0.000
two-photon	**1.000**±0.000	0.988±0.005	0.861±0.013	0.919±0.039	*0.941*±0.034
Q2-role	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000	**1.000**±0.001	0.998±0.004
Classical positional encodings					
LapPE	0.959±0.012	0.978±0.003	0.966±0.001	0.905±0.003	0.676±0.009
RWPE	0.989±0.004	**1.000**±0.000	*0.998*±0.001	*0.948*±0.005	0.556±0.432

The configurations with seed-specific low-performing runs are listed below. At $\mathrm{LR}(1, 1)$, Q2-role scores $(1.000, 1.000, 0.112)$ across the three fixed seeds, and TGN scores $(0.902, 0.111, 0.988)$. At $\mathrm{LR}(1, 16)$, two-photon scores $(0.943, 0.060, 0.059)$, and RWPE scores $(0.072, 0.681, 0.791)$.

At $\mathrm{LR}(4, 16)$, RWPE scores $(0.834, 0.776, 0.059)$, and TGN scores $(0.816, 0.060, 0.806)$.

At $\mathrm{LR}(16, 1)$, one-photon scores $(0.930, 0.957, 0.059)$, LapPE scores $(0.886, 0.059, 0.776)$, and RWPE scores $(0.059, 0.059, 0.763)$. At $\mathrm{LR}(16, 16)$, RWPE scores $(0.171, 0.060, 0.769)$, and DyGFormer scores $(0.337, 0.059, 0.436)$.

At $\mathrm{LR}(32, 1)$, one-photon scores $(0.654, 0.059, 0.249)$. At $\mathrm{LR}(32, 4)$, LapPE scores $(0.352, 0.429, 0.059)$. At $\mathrm{LR}(32, 8)$, Q2-role scores $(0.057, 0.801, 0.059)$. At $\mathrm{LR}(32, 16)$, Q1-role scores $(0.685, 0.059, 0.666)$. These results show that some low means combine one or two low-performing runs with substantially stronger results from the remaining fixed seeds.

Table 11: Test F_1 for long-range memory $\mathrm{LR}(\ell, d)$ at lags ℓ=16 and 32 on M=100 graphs. The metric is computed on edges incident to the memory node. Entries are means and standard deviations over three seeds. The top three means in each column of each block are marked **First**, Second and *Third*; tied entries share a rank.

	long-range $\mathrm{LR}(16, d)$				
	d=1	d=2	d=4	d=8	d=16
Non-learned baselines					
Persistence	0.032±0.000	0.035±0.000	0.034±0.000	0.032±0.000	0.028±0.000
EdgeBank	0.059±0.000	0.059±0.000	0.059±0.000	0.059±0.000	0.059±0.000
Discrete-time GNNs					
GCN	0.058±0.000	0.059±0.000	0.059±0.001	0.058±0.001	0.057±0.001
GAT	0.057±0.000	0.059±0.001	0.058±0.000	0.058±0.000	0.059±0.001
TGCN	0.083±0.007	0.088±0.020	0.066±0.003	0.061±0.001	0.059±0.001
GC-LSTM	0.144±0.075	0.084±0.013	0.059±0.001	0.059±0.000	0.058±0.001
EvolveGCN	0.059±0.000	0.059±0.000	0.058±0.001	0.058±0.001	0.059±0.000
Continuous-time GNNs					
TGAT	**1.000**±0.000	**1.000**±0.000	0.876±0.067	**0.992**±0.002	0.441±0.040
TGN	**1.000**±0.000	0.998±0.001	0.896±0.050	0.871±0.056	0.314±0.136
DyGFormer	0.985±0.004	0.829±0.120	0.648±0.022	0.726±0.149	0.278±0.195
CTAN	0.034±0.030	0.055±0.003	0.060±0.000	0.059±0.000	0.059±0.001
*Quantum-walk encodings (*PHoTE*)*					
one-photon	0.649±0.511	0.868±0.027	0.500±0.084	0.152±0.082	0.060±0.001
Q1-role	0.973±0.002	0.934±0.008	*0.934*±0.018	*0.953*±0.006	0.940±0.017
two-photon	*0.989*±0.011	0.850±0.029	0.728±0.030	0.403±0.011	0.226±0.037
Q2-role	0.988±0.001	*0.959*±0.014	**0.969**±0.004	0.955±0.003	**0.947**±0.021
Classical positional encodings					
LapPE	0.574±0.449	0.888±0.006	0.864±0.023	0.730±0.002	*0.511*±0.023
RWPE	0.293±0.407	0.947±0.013	0.941±0.008	0.894±0.008	0.333±0.382

	long-range $\mathrm{LR}(32, d)$				
	d=1	d=2	d=4	d=8	d=16
Non-learned baselines					
Persistence	0.034±0.000	0.035±0.000	0.034±0.000	0.031±0.000	0.029±0.000
EdgeBank	0.059±0.000	0.059±0.000	0.059±0.000	0.059±0.000	0.059±0.000
Discrete-time GNNs					
GCN	0.058±0.000	0.059±0.000	0.058±0.000	0.057±0.001	0.058±0.001
GAT	0.058±0.000	0.059±0.000	0.059±0.000	0.059±0.001	0.059±0.001
TGCN	0.063±0.002	0.063±0.003	0.059±0.001	0.059±0.000	0.058±0.001
GC-LSTM	0.074±0.004	0.060±0.000	0.059±0.000	0.059±0.001	0.058±0.001
EvolveGCN	0.059±0.000	0.058±0.001	0.057±0.000	0.059±0.000	0.059±0.000
Continuous-time GNNs					
TGAT	**1.000**±0.000	**1.000**±0.000	**0.939**±0.026	*0.323*±0.012	0.059±0.000
TGN	**1.000**±0.000	0.999±0.001	0.773±0.148	0.307±0.046	0.059±0.001
DyGFormer	*0.973*±0.009	0.800±0.041	0.476±0.059	0.115±0.025	0.059±0.000
CTAN	0.022±0.006	0.056±0.001	0.059±0.000	0.059±0.000	0.059±0.001
*Quantum-walk encodings (*PHoTE*)*					
one-photon	0.321±0.304	0.068±0.002	0.059±0.000	0.058±0.001	0.058±0.000
Q1-role	0.839±0.045	0.630±0.164	0.879±0.005	**0.894**±0.021	0.470±0.356
two-photon	0.880±0.017	0.597±0.029	0.072±0.016	0.059±0.001	*0.060*±0.000
Q2-role	0.907±0.016	*0.839*±0.026	*0.857*±0.020	0.306±0.429	**0.721**±0.141
Classical positional encodings					
LapPE	0.060±0.000	0.059±0.000	0.280±0.195	0.155±0.084	0.059±0.000
RWPE	0.532±0.059	0.617±0.043	0.550±0.113	0.595±0.058	0.059±0.000

At the reported precision, Q2-role has a higher mean than Q1-role in 9 of the 20 long-range settings, ties it in 5, and scores lower in 6. Its advantage is therefore task-dependent. For example, at $\mathrm{LR}(32, 2)$, Q2-role scores 0.839, compared with 0.630 for Q1-role. At $\mathrm{LR}(32, 8)$, the ordering reverses: Q1-role scores 0.894, compared with 0.306 for Q2-role. The per-seed results above show that some of these mean differences are strongly affected by individual low-performing runs.

Cause-and-effect. On cause-and-effect, Q2-role has the highest mean among the non-trainable encodings at all six tested lags, including ties (Table 12). It scores 1.000 at lags 1, 2, 4, and 8, 0.996 ± 0.004 at lag 16, and 0.871 ± 0.107 at lag 32. At the longest tested delay, Q1-role is close at 0.863 ± 0.083, while TGN achieves the highest overall mean, 0.993 ± 0.002.

One cause-and-effect configuration contains a seed-specific low-performing run. At $\mathrm{CE}(32)$, LapPE scores $(0.548, 0.264, 0.563)$.

Table 12: Test F_1 for cause-and-effect CE(ℓ) at M=100, computed on edges incident to the memory node. Entries are means and standard deviations over three seeds. The top three means in each column are marked **First**, Second and *Third*; tied entries share a rank, so a column with several tied first places marks no second or third place.

	cause-and-effect CE(ℓ)					
	ℓ=1	ℓ=2	ℓ=4	ℓ=8	ℓ=16	ℓ=32
Non-learned baselines						
Persistence	0.149±0.000	0.149±0.000	0.149±0.000	0.149±0.000	0.149±0.000	0.149±0.000
EdgeBank	0.275±0.000	0.275±0.000	0.275±0.000	0.275±0.000	0.275±0.000	0.275±0.000
Discrete-time GNNs						
GCN	0.990±0.003	0.262±0.000	0.262±0.000	0.262±0.000	0.261±0.000	0.261±0.001
GAT	0.998±0.002	0.264±0.002	0.263±0.000	0.262±0.001	0.262±0.001	0.261±0.002
TGCN	**1.000**±0.000	0.997±0.001	0.874±0.054	0.350±0.022	0.283±0.006	0.269±0.005
GC-LSTM	**1.000**±0.000	0.999±0.001	0.815±0.040	0.342±0.011	0.270±0.003	0.267±0.002
EvolveGCN	0.982±0.006	0.253±0.000	0.253±0.000	0.233±0.017	0.232±0.018	0.243±0.017
Continuous-time GNNs						
TGAT	0.999±0.001	0.999±0.000	0.996±0.007	**1.000**±0.000	0.996±0.007	0.992±0.002
TGN	**1.000**±0.000	0.999±0.001	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000	**0.993**±0.002
DyGFormer	**1.000**±0.000	0.997±0.001	0.996±0.001	0.985±0.004	0.981±0.011	*0.977*±0.002
CTAN	0.256±0.013	0.239±0.011	0.229±0.009	0.236±0.007	0.243±0.007	0.269±0.006
*Quantum-walk encodings (*PHOTE*)*						
one-photon	0.998±0.000	0.995±0.002	0.992±0.001	0.992±0.004	0.957±0.014	0.525±0.044
Q1-role	**1.000**±0.000	**1.000**±0.000	0.999±0.000	**1.000**±0.000	0.993±0.006	0.863±0.083
two-photon	0.999±0.001	**1.000**±0.000	**1.000**±0.000	0.997±0.004	0.976±0.024	0.786±0.008
Q2-role	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000	0.996±0.004	0.871±0.107
Classical positional encodings						
LapPE	0.964±0.002	0.965±0.002	0.954±0.003	0.945±0.001	0.899±0.005	0.458±0.169
RWPE	0.999±0.000	0.999±0.000	0.999±0.000	0.998±0.001	0.978±0.002	0.816±0.034

On cause-and-effect, Q2-role has a higher mean than Q1-role at lags 4, 16, and 32, and ties it at lags 1, 2, and 8. At lag 32, the means are close: 0.871 for Q2-role and 0.863 for Q1-role. Thus, Q2-role matches or exceeds Q1-role across all six tested cause-and-effect settings.

C FIGURES FOR THE FULL SYNTHETIC BENCHMARK

Figures 3–7 plot the results from Tables 7–12 for all 53 synthetic settings of T-GRAB with $M{=}100$ nodes. For trainable methods, curves show the mean test F_1 over three seeds, and shaded bands indicate one standard deviation, clipped to $[0, 1]$.

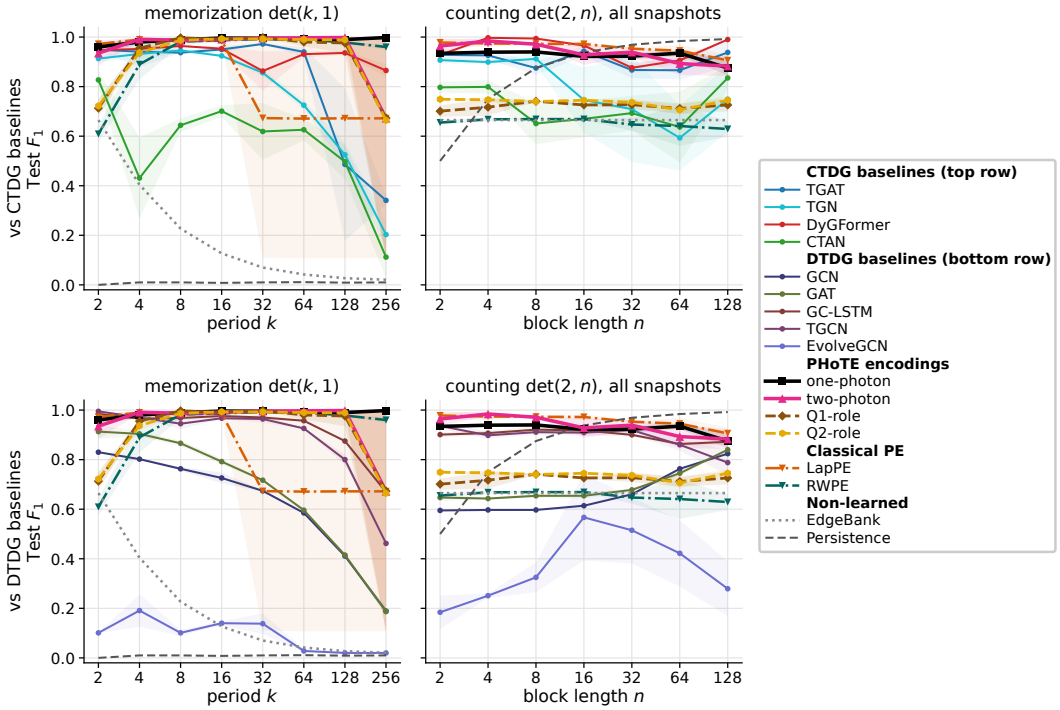


Figure 3: Memorization and Counting: test F_1 averaged over all test snapshots. The left panels show memorization $\det(k, 1)$ and the right panels show counting $\det(2, n)$. Continuous-time GNNs appear in the top row and discrete-time GNNs in the bottom row. Black solid: one-photon; brown dashed: Q1-role; magenta solid: two-photon; gold dashed: Q2-role; gray dashed: Persistence.

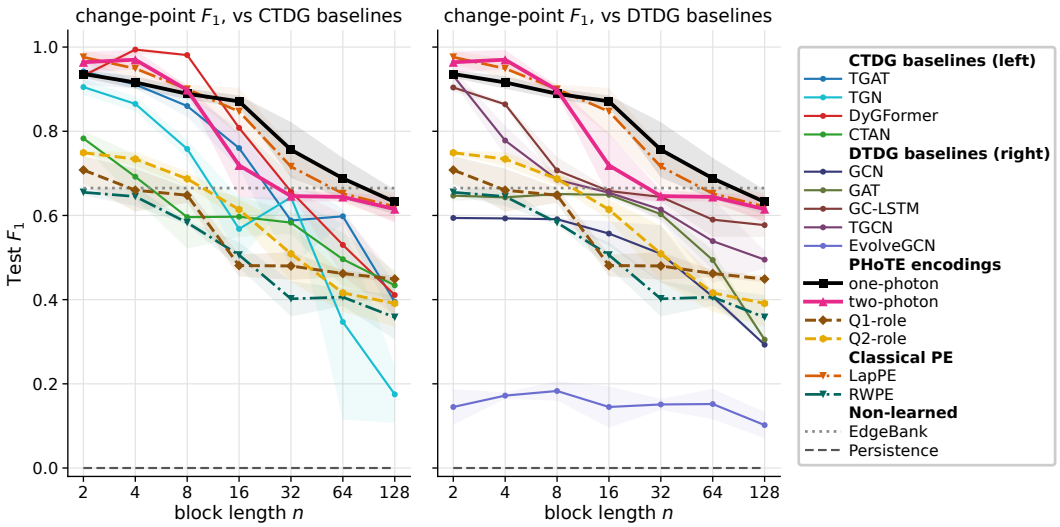


Figure 4: Change-point F_1 for counting $\det(2, n)$, evaluated only on snapshots where the active graph changes. Continuous-time GNNs appear on the left and discrete-time GNNs on the right. The gray dotted line is the constant EdgeBank score of 0.665. The one-photon encoding has the highest mean among trainable methods at $n = 16$, 32, 64, and 128, although its mean of 0.632 at $n = 128$ is below EdgeBank.

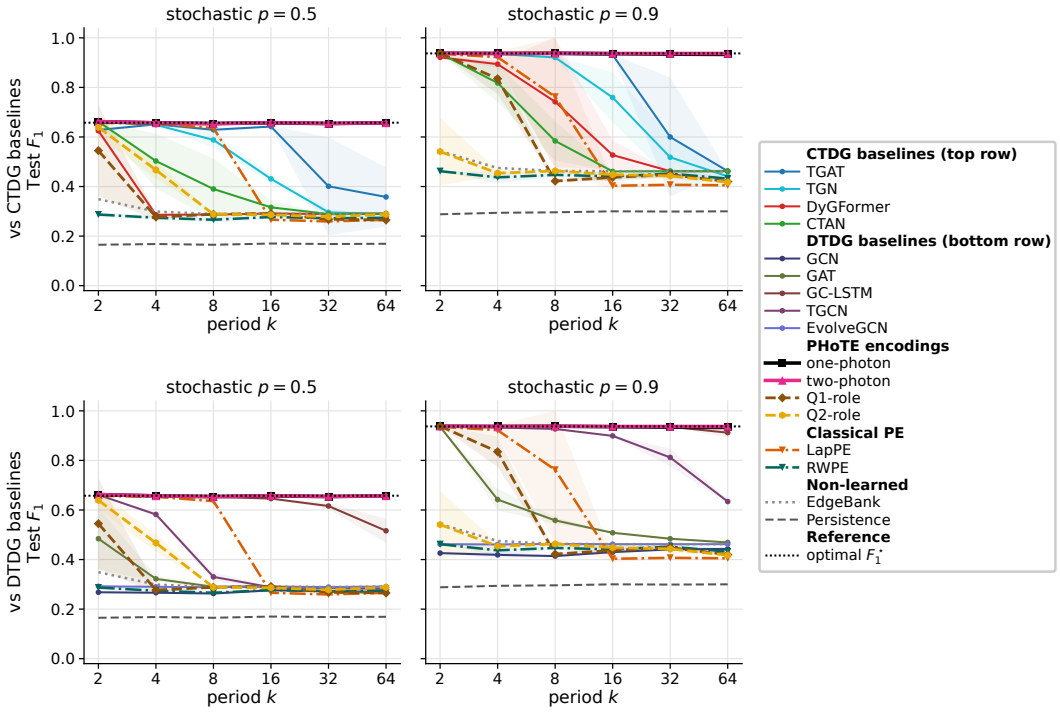


Figure 5: Stochastic periodicity $\mathrm{sto}(k, 1)$ at p=0.5 (left) and p=0.9 (right). Continuous-time GNNs appear in the top row and discrete-time GNNs in the bottom row. The dotted reference line marks the optimal expected-count F_1 from Proposition 1. The projected one-photon and two-photon encodings remain close to this reference at every tested k. In contrast, both role summaries have means at or below EdgeBank at the reported precision for every tested $k \geq 8$.

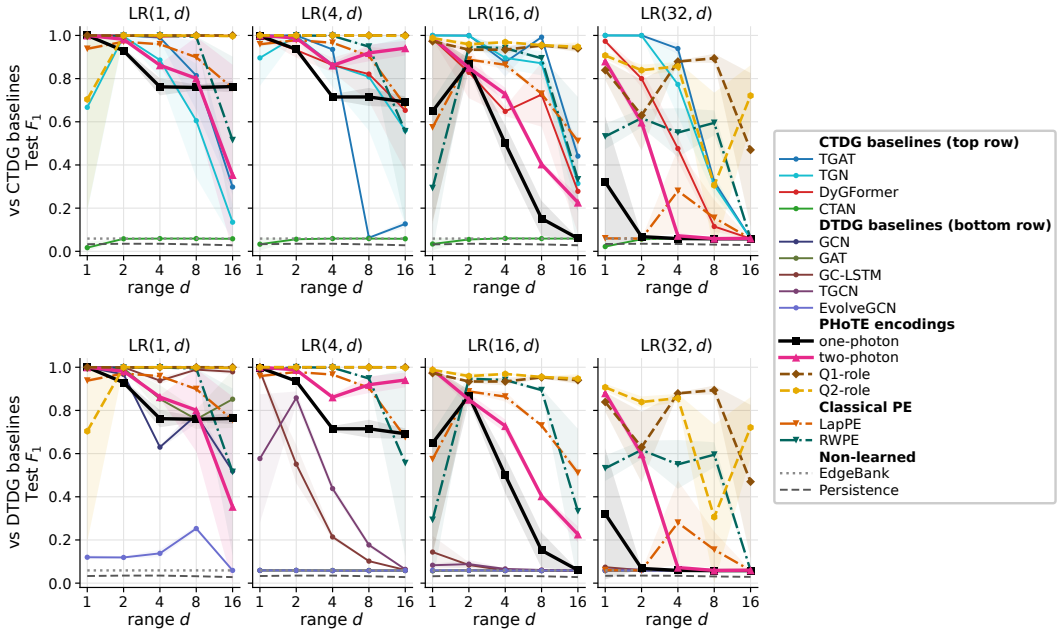


Figure 6: Long-range memory $\mathrm{LR}(\ell, d)$: memory-node F_1 as a function of spatial range d, with one column for each lag ℓ. Continuous-time GNNs appear in the top row and discrete-time GNNs in the bottom row. At $\ell = 16$, TGAT attains or ties the highest mean at $d = 1$, 2, and 8, while Q2-role leads at $d = 4$ and 16. At $\ell = 32$, TGAT leads or ties at $d = 1$, 2, and 4, while Q1-role leads at $d = 8$ and Q2-role at $d = 16$. Some settings exhibit substantial variation across seeds.

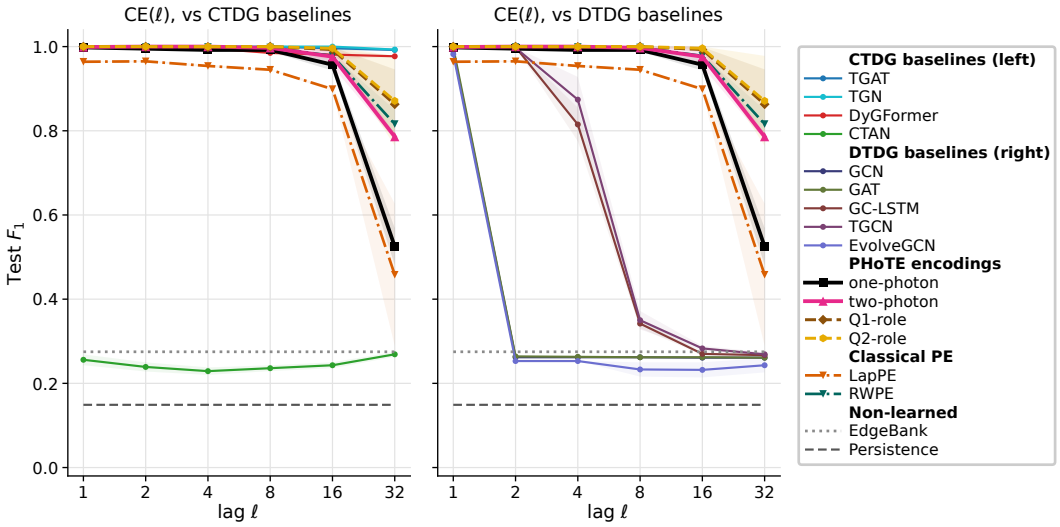


Figure 7: Cause-and-effect $\mathrm{CE}(\ell)$: memory-node F_1 as a function of lag. Continuous-time GNNs appear on the left and discrete-time GNNs on the right. At the reported precision, Q2-role attains or ties the highest mean among the non-trainable encodings at all six tested lags. At $\ell = 32$, its mean is 0.871, compared with 0.993 for TGN, which leads overall.

D QUANDELA EXPERIMENTS

This appendix describes the simulator and hardware experiments on the 13-node baboon contact network. The backend names `sim:belenos` and `qpu:belenos` denote Quandela's Belenos simulator and physical processor, respectively. The simulator evaluates one- and two-photon encodings, whereas the hardware measurements cover the one-photon encoding on the full 15-minute test sequence.

D.1 DATA AND SHARED OPTICAL TRANSFORMATIONS

For the 15-minute sequence, we evaluate analytic, `sim:belenos` and `qpu:belenos` one-photon features on the full test set, and analytic and `sim:belenos` two-photon features. For the 5-minute sequence, we evaluate analytic and `sim:belenos` features on the full test set.

For each indexed test snapshot t, let A_t be its 13×13 binary adjacency matrix after removing self-loops. We retain its unordered edge set E_t as the list of two-photon inputs. At each of the three evolution times, the optical transformation is

$$U_{t,\tau} = \exp(-iA_t\tau), \qquad \tau \in \{0.2, 0.8, 1.8\}.$$

Each transformation acts on 13 graph modes, with each mode corresponding to one node.

D.2 EXECUTION ON `SIM:BELENOS`

One-photon output distributions. For every pair (t, τ), we submit one cloud job containing the corresponding circuit and 13 input iterations. Each iteration injects one photon into a different graph mode. The simulator runs in sampling mode with 10,000 detected events per input; an event is retained only when at least one photon is detected. The returned output frequencies give a 13-component distribution for each input node. The event budget counts accepted detections, not input attempts.

Two-photon output distributions. For each edge $\{a, b\} \in E_t$ and evolution time τ, we inject one photon into mode a and one into mode b, with vacuum in the other modes, so a circuit has $|E_t|$ two-photon input iterations. The simulator runs in sampling mode with 10,000 detected events per input edge. Its output states record the number of photons in each mode, so two photons leaving through the same mode are recorded as a two-photon event. We retain the events in which exactly two photons are detected, in different modes or in the same mode, and normalize them to a distribution over the 91 unordered output pairs; these distributions are summed over the input edges of E_t to construct $E_t^{(2)}(\tau)$.

D.3 FEATURE CONSTRUCTION AND READOUT COMPARISON

Replacing analytic features. For one-photon inputs, the returned distribution for an input node v replaces the corresponding column of the analytic table $Q_t(\tau) = |U_{t,\tau}|^2$ on the test segment. The distributions at the three evolution times are concatenated in the same order as in the analytic encoding. The random projection and temporal readout are unchanged.

For two-photon inputs, let $p_{t,\tau}(i, j \mid a, b)$ denote the returned relative frequency of the unordered output pair $\{i, j\}$, estimated from the retained two-photon events for input edge $\{a, b\}$. The aggregate table is reconstructed as

$$\widehat{E}_t^{(2)}(\tau)[i, j] = \sum_{\{a,b\} \in E_t} p_{t,\tau}(i, j \mid a, b), \qquad i \le j.$$

The diagonal entries are retained. We extract the node columns in the same way as for the analytic two-photon encoding, so the feature definition remains unchanged.

Training and configuration selection. Training and validation use analytic features. In the simulator and hardware comparisons, only test features are replaced by simulator or hardware outputs; the analytic comparison uses analytic features throughout. The one-photon simulator and hardware comparisons use the same readout configuration and seeds.

Table 13: Parameters of the Belenos processor relevant to our measurements, read from Quandela's cloud service during our hardware runs. The values may change after recalibration.

Parameter	`qpu:belenos`
Detectors	threshold (some quasi number-resolving)
HOM visibility	85%
$g^{(2)}$	1.0%
End-to-end transmittance	5.2%

D.4 Execution on `qpu:belenos`

Only one-photon inputs were measured on the physical processor, on all 273 test snapshots of the 15-minute sequence, at all three evolution times and for all 13 input nodes. This gives 819 circuits and 10,647 input settings, with 10,000 detected events per setting.

D.5 The Quandela platform

Platform parameters. Table 13 reports the processor parameters recorded during our hardware runs. We describe their relevance to preparing the photon inputs and estimating the output probabilities used as node features.

Detectors. For the one-photon encoding, we record which output mode detects the photon and use the observed frequencies to estimate the probabilities $Q_t(\tau)[i, v]$. A threshold detector provides this detection signal, called a "click", but cannot distinguish one photon from two arriving together in the same mode. Some channels are quasi number-resolving, meaning that they can identify some, but not all, such two-photon events. This distinction matters when measuring two-photon coincidences, including events with both photons in the same mode. Our physical-processor experiments use only one-photon inputs; the two-photon experiments use simulator outputs that record the photon number in each mode.

HOM visibility. Our analytic two-photon probabilities in Section 2 assume fully indistinguishable photons: the two assignments of input photons to an output pair interfere. Differences between the photons can weaken this interference and change the coincidence probabilities used to construct the two-photon features. Hong–Ou–Mandel (HOM) visibility characterizes this interference in a standard calibration experiment (Hong et al., 1987). The processor reports 85%, compared with 100% in the ideal experiment. Since our hardware evaluation uses only one-photon inputs, we report this value as a source characteristic, not as a measure of the accuracy of our measured one-photon features.

Multi-photon emission ($g^{(2)}$). Our one-photon encoding assumes that exactly one photon is injected into the selected input mode. The source can occasionally emit additional photons, violating this input assumption. The reported $g^{(2)} = 1.0\%$ characterizes such unwanted emission; it is a normalized correlation measure, not the percentage of pulses containing extra photons. This source error is different from our intended two-photon inputs, which place one photon in each of two different modes.

End-to-end transmittance. The reported transmittance of 5.2% means that only about 5.2% of emitted photons are detected. Our output probabilities are estimated from accepted detection events, and each input requires $10{,}000$ such events. Photon loss therefore increases the number of emission attempts and the collection time needed to obtain the features.

These platform parameters alone do not establish how closely the measured features reproduce their analytic counterparts. The output-distribution comparison below evaluates this agreement, including the combined effects of the source, optical circuit, detection, and finite sampling.

Collection time. Table 14 reports the execution time of the processor, as reported by the platform, on the 15-minute aggregation. A test snapshot requires three circuits, one per evolution time, each with 13 one-photon inputs run until $10{,}000$ events are detected. We do not report simulator times: the time from submission to results that we recorded for the simulator is dominated by cloud queueing and does not reflect its computation time.

Table 14: Execution time of `qpu:belenos` per test snapshot of the 15-minute aggregation, in seconds. The time per input is the snapshot time divided by its 39 inputs.

	Mean	Median	Max
One test snapshot	287	290	341
One input	7.4	7.4	8.7

Table 15: Total variation distance between measured and analytic output distributions, over all input settings of the test segment.

Backend	Input	Mean	Median	Max
`qpu:belenos`, 15 min	one photon	0.062	0.035	0.834
`sim:belenos`, 15 min	one photon	0.004	0.002	0.022
`sim:belenos`, 5 min	one photon	0.002	0.000	0.018
`sim:belenos`, 15 min	two photons	0.027	0.030	0.054
`sim:belenos`, 5 min	two photons	0.025	0.027	0.055

Output-distribution agreement. For a one-photon input setting, let p_i be the analytic probability of detection at output mode i and let $\widehat{p}_i$ be its simulator or hardware estimate. We measure their discrepancy using the total variation distance (TVD),

$$\mathrm{TVD}(\widehat{p}, p) = \frac{1}{2} \sum_{i=1}^{13} |\widehat{p}_i - p_i|,$$

which is zero when the two output distributions agree. For a two-photon input setting the same distance is taken over the 91 unordered output pairs, with p the analytic two-photon probability and $\widehat{p}$ the frequency among the sampled events in which exactly two photons are detected, normalized over those events. Table 15 summarizes the TVD over every input setting of the test segment (one setting = one input node or input edge at one evolution time).

On the 15-minute stream, the hardware one-photon distributions have mean TVD 0.062 from the analytic probabilities, against 0.004 for the simulator on the same inputs. The mean hardware TVDs are 0.030, 0.082, and 0.075 at $\tau = 0.2$, 0.8, and 1.8, respectively, with larger discrepancies at the two longer evolution times. The larger two-photon TVD may partly reflect the larger output space being estimated at the same sampling budget.

E PROOFS AND DERIVATIONS

E.1 PROOF OF PROPOSITION 1

We first maximize the F_1 score computed from expected edge counts, then discuss its relation to the score on finite graph samples.

Proof. Condition on the community assignment of the block model generating the next snapshot. Let M_{in} and M_{out} denote the total numbers of unordered node pairs within and between communities, respectively. Each pair independently forms an edge with probability p within a community and q between communities, where $p > q$.

A deterministic predictor selects a set S before observing the next graph sample. Let a and b be the predicted numbers of within-community and between-community pairs in S. Let X be the number of correctly predicted edges and T the total number of edges in the sampled graph. The realized score is

$$F_1 = \frac{2X}{a + b + T}.$$

The expected total edge count is

$$P = \mathbb{E}[T] = pM_{\text{in}} + qM_{\text{out}},$$

and the expected number of correctly predicted edges is $\mathbb{E}[X] = pa + qb$. Thus, the score computed from expected counts is

$$\overline{F}_1(a, b) = \frac{2\mathbb{E}[X]}{a + b + \mathbb{E}[T]} = \frac{2(pa + qb)}{a + b + P}.$$

For any fixed number b of between-community predictions, increasing the number a of within-community predictions increases the expected-count score, since

$$\frac{\partial \overline{F}_1(a, b)}{\partial a} = \frac{2\big[(p - q)b + pP\big]}{(a + b + P)^2} > 0.$$

Here, $p > q \geq 0$, $b \geq 0$, and $P > 0$. Thus, regardless of how many between-community pairs are selected, adding any remaining within-community pair improves the score. Every optimal prediction set must therefore include all M_{in} within-community pairs. We next determine whether it should also include any between-community pairs.

After these pairs have been included, the effect of adding between-community predictions is determined by

$$\frac{\partial \overline{F}_1(M_{\text{in}}, b)}{\partial b} = \frac{2\big[q(M_{\text{in}} + P) - pM_{\text{in}}\big]}{(M_{\text{in}} + b + P)^2}.$$

This derivative is negative when

$$q(M_{\text{in}} + P) < pM_{\text{in}},$$

which holds for both tested densities, $p = 0.9$ and $p = 0.5$, with $q = 0.01$ and community sizes $(34, 33, 33)$.

The optimal prediction set therefore contains every within-community pair and no between-community pair. Its expected-count precision is p, and its expected-count recall is

$$\text{Rec} = \frac{pM_{\text{in}}}{pM_{\text{in}} + qM_{\text{out}}}.$$

Consequently,

$$F_1^\star = \frac{2pM_{\text{in}}}{M_{\text{in}} + P} = \frac{2p\,\text{Rec}}{p + \text{Rec}},$$

as stated in Proposition 1. □

Corollary 1. *For fixed community sizes and edge probabilities, $F_1^\star$ is independent of the cycle length k. Conditional on the block model generating the target snapshot, each prediction concerns one fresh graph sample, regardless of how many block models appear in the cycle.*

We derive the maximum F_1 computed from expected edge counts, assuming that the community assignment generating the next snapshot is known. We use this value as a theoretical reference for the empirical results.

Relation to the empirical results. For the tested community sizes, $M_{\text{in}} = 1617$ and $M_{\text{out}} = 3333$. Substitution gives $F_1^\star = 0.9372$ when $p = 0.9$ and $F_1^\star = 0.6576$ when $p = 0.5$. Table 9 shows that the identity-sensitive one- and two-photon encodings remain within a few 10^{-3} of these values for every tested k.

Other representations show larger losses as k grows. LapPE drops markedly from $k = 16$ onward. Eigenvector sign changes and rotations within an eigenspace can give similar community structures inconsistent coordinates across independently sampled snapshots. RWPE may provide less information for distinguishing communities because nodes in different communities can have similar random-walk return probabilities. Several trained models also decline: GC-LSTM reaches 0.912 at $p = 0.9$, $k = 64$, and TGAT reaches 0.358 at $p = 0.5$, $k = 64$.

For a fixed adjacency matrix and evolution time, the one-photon probability table is uniquely determined and has no eigenvector sign or basis ambiguity. The results close to these reference values suggest that the identity-sensitive encodings retain enough community information for prediction despite edge resampling. In contrast, the role summaries discard the correspondence between non-self feature values and destination nodes. Q1-role is close to the reference value at $p = 0.9$, $k = 2$, but both role summaries fall below it for $k \geq 4$.

E.2 Information at the three evolution times

We consider one snapshot and omit the snapshot index t. Let $G = (V, E)$ have adjacency matrix A, and write $U(\tau) = \exp(-iA\tau)$ for its quantum-walk transformation. The one-photon probability table is $Q(\tau)[i, v] = |U(\tau)[i, v]|^2$.

We evaluate the walk at three fixed times, $\tau_1 = 0.2$, $\tau_2 = 0.8$, and $\tau_3 = 1.8$. To illustrate the information available at these times, let

$$q_v(\tau) = Q(\tau)[v, v] = |U(\tau)[v, v]|^2$$

denote the probability that a photon injected at node v is detected at the same node.

For an undirected, unweighted graph without self-loops, the degree $\deg(v)$ is the number of neighbors directly connected to node v. Since A_{vw} is one for adjacent nodes and zero otherwise,

$$\deg(v) = \sum_{w \in V} A_{vw}, \qquad (A^2)_{vv} = \sum_{w \in V} A_{vw} A_{wv} = \deg(v).$$

The latter identity counts the two-step walks from v to a neighbor and back to v, with one such walk for each neighbor. Expanding $U(\tau) = \exp(-iA\tau)$ and taking the squared modulus of its diagonal entry gives

$$q_v(\tau) = 1 - \deg(v)\tau^2 + O(\tau^4).$$

Thus, at sufficiently short evolution times, the probability of detecting the photon away from its input node is approximately $\deg(v)\tau^2$. The leading departure from the initial self-return probability of one therefore reflects the input node's degree.

Taking an isolated edge as an example, the return probability can be calculated exactly. Let its endpoints be v and w, with neither node connected to any other node. In the node order (v, w), the adjacency matrix of this component is

$$A_{\text{edge}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Let I_2 denote the 2×2 identity matrix. Since $A_{\text{edge}}^2 = I_2$, every even power of A_{edge} equals I_2, and every odd power equals A_{edge}. Substituting these identities into the power-series expansion of the matrix exponential gives

$$\begin{aligned} U_{\text{edge}}(\tau) &= \exp(-iA_{\text{edge}}\tau) \\ &= I_2 - i\tau A_{\text{edge}} - \frac{\tau^2}{2!} I_2 + \frac{i\tau^3}{3!} A_{\text{edge}} \\ &\quad + \frac{\tau^4}{4!} I_2 - \frac{i\tau^5}{5!} A_{\text{edge}} + \cdots . \end{aligned}$$

We now add together the coefficients multiplying I_2 and, separately, those multiplying A_{edge}:

$$U_{\text{edge}}(\tau) = \left(1 - \frac{\tau^2}{2!} + \frac{\tau^4}{4!} - \cdots\right) I_2 - i\left(\tau - \frac{\tau^3}{3!} + \frac{\tau^5}{5!} - \cdots\right) A_{\text{edge}}.$$

The two scalar series are exactly $\cos\tau$ and $\sin\tau$, respectively. Therefore,

$$U_{\text{edge}}(\tau) = \cos\tau\, I_2 - i\sin\tau\, A_{\text{edge}} = \begin{pmatrix} \cos\tau & -i\sin\tau \\ -i\sin\tau & \cos\tau \end{pmatrix}.$$

A photon injected at v therefore has amplitude $\cos\tau$ at v and $-i\sin\tau$ at w. Taking squared moduli yields

$$q_v(\tau) = \cos^2\tau, \qquad Q(\tau)[w, v] = \sin^2\tau.$$

At the shortest evolution time, $\tau_1 = 0.2$, these probabilities are approximately 0.9605 and 0.0395, respectively. The expansion $\cos^2\tau = 1 - \tau^2 + O(\tau^4)$ agrees with the general short-time expression because $\deg(v) = 1$. The exact cosine formula applies to this isolated two-node component, not to every degree-one node in a larger component.

At $\tau_2 = 0.8$, the probability profile can distinguish nodes with the same degree but different surrounding structures. Consider three connected components, each with a degree-one input node v and its only neighbor w. The first component is an isolated edge connecting v and w. The second is a three-node path with edges $\{v, w\}$ and $\{w, u_1\}$. The third is a four-node star: the center w connects to v, u_1, and u_2, with no edges between these three outer nodes. Each outer node is called a leaf because its degree is one. The photon is injected at leaf v, not at the degree-three center w.

For each component, we compute the output probabilities as $Q(\tau_2)[i, v] = |\langle i|U(\tau_2)|v\rangle|^2$. Listing the input node v first, its neighbor w second, and the remaining nodes afterward gives

isolated edge, in order (v, w) :	$(0.485, 0.515)$,
three-node path, in order (v, w, u_1) :	$(0.508, 0.410, 0.083)$,
four-node star, in order (v, w, u_1, u_2) :	$(0.530, 0.322, 0.074, 0.074)$.

All three input nodes have degree one, but their neighbors have degrees one, two, and three, respectively. The resulting probability profiles differ, including their self-return entries. These examples show that the encoding captures information about the structure beyond the input node's immediate connection, rather than only its degree.

At the longest evolution time, $\tau_3 = 1.8$, the probability can either be concentrated away from the input node or return close to it, depending on the graph structure. For a photon injected at endpoint v of an isolated edge $\{v, w\}$, the expression derived above gives

$$q_v(1.8) = \cos^2(1.8) \approx 0.052, \qquad Q(1.8)[w, v] = \sin^2(1.8) \approx 0.948.$$

Thus, the photon is detected at the other endpoint with approximately 94.8% probability.

Now consider the same four-node star as in the preceding example, but inject the photon at its center w, rather than at a leaf. The center connects to three leaves, v, u_1, and u_2, with no edges between them. In the node order (w, v, u_1, u_2), its adjacency matrix is

$$A_{\text{star}} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Every two-step walk from the center goes to one of the three leaves and then returns to the center. For a walk of length $2m$, there are three choices at each of the m departures from the center, giving 3^m closed walks. An odd-length walk cannot return to the center. Therefore, for every integer $m \geq 0$,

$$(A_{\text{star}}^{2m})_{ww} = 3^m, \qquad (A_{\text{star}}^{2m+1})_{ww} = 0.$$

Let $U_{\mathrm{star}}(\tau) = \exp(-iA_{\mathrm{star}}\tau)$. Substituting these diagonal entries into the matrix-exponential series gives the amplitude for returning to the center:

$$\begin{aligned} U_{\mathrm{star}}(\tau)[w,w] &= 1 - \frac{3\tau^2}{2!} + \frac{3^2\tau^4}{4!} - \frac{3^3\tau^6}{6!} + \cdots \\ &= 1 - \frac{(\sqrt{3}\tau)^2}{2!} + \frac{(\sqrt{3}\tau)^4}{4!} - \frac{(\sqrt{3}\tau)^6}{6!} + \cdots \\ &= \cos(\sqrt{3}\tau). \end{aligned}$$

Taking the squared modulus yields

$$q_w(\tau) = \cos^2(\sqrt{3}\tau), \qquad q_w(1.8) = \cos^2(\sqrt{3} \times 1.8) \approx 0.999.$$

The return probability is close to one, which means that probability is almost entirely concentrated at the center again at this particular time, not that it remains there throughout the evolution. The value depends on both the four-node star structure and the chosen evolution time; it does not apply to every degree-three node.

Together, the three evolution-time examples describe different aspects of the walk: the initial spread related to node degree, differences caused by the surrounding structure, and later redistribution or return of probability. Combining measurements at these times therefore records more of this behavior than a short-time degree approximation alone.

E.3 Additional properties of the four encodings

Recurring structure and destination-node information. Role summaries retain selected walk values and statistics without recording which destination node each non-self value refers to. Consider injecting one photon at node 17 separately in each of the two snapshots shown in Figure 8. In the first snapshot, nodes 17 and 42 form an isolated edge. In a later snapshot, nodes 17 and 88 form another isolated edge. In each case, the two endpoints connect only to each other. These are the two events in Table 16, with $v = 17$, $w_1 = 42$, and $w_2 = 88$.

At the same evolution time, the photon has the same probabilities of being detected at node 17 and at its partner in both snapshots. However, the partner's probability occupies a different coordinate in the original one-photon profile. For example, at $\tau_3 = 1.8$, the probability is approximately 0.95 at node 42 in the first snapshot and at node 88 in the later snapshot. Thus, the same structural event produces the same numerical value at different feature positions.

For the two-photon encoding, one photon is injected at each endpoint of the current edge. The coincidence value associated with the partner likewise moves from coordinate 42 to coordinate 88 in the profile assigned to node 17. A temporal readout must therefore learn that values at these different coordinates can describe the same isolated-edge event.

The role summaries give identical vectors for these two events because they retain the self entry, sorted non-self values, and power sums rather than the partner's identity. Figure 8 shows this behavior for the complete Q1-role vector. RWPE is also unchanged in this example, but its return probabilities alternate between zero and one, whereas the role summaries retain the quantum-walk values at the three evolution times. This gives the readout a consistent representation of the isolated-edge event even when the partner changes.

Relation to classical return probabilities. Both Q1-role and RWPE include information about returning to the starting node, but they use different dynamics. Q1-role retains the quantum self-return probability $q_v(\tau) = |U(\tau)[v,v]|^2$ at each evolution time. RWPE instead uses a classical random walk. Let $P[v,w]$ be its probability of moving from node v to node w in one step. Then $(P^k)_{vv}$ is the probability of returning to v after k steps. These classical return probabilities are not the quantum self-return probabilities used by Q1-role.

To illustrate the path structure in $\mathrm{LR}(\cdot, 4)$, consider three paths of length four that share only their source node, which we call the root. Along one path, write the nodes in order as $r - v_1 - v_2 - v_3 - v_4$, where r is the root. The endpoint is v_4, its neighbor is v_3, the middle node is v_2, and the node next to the root is v_1. Other nodes in this illustrative graph have no incident edges. We call these isolated background nodes *filler nodes*.

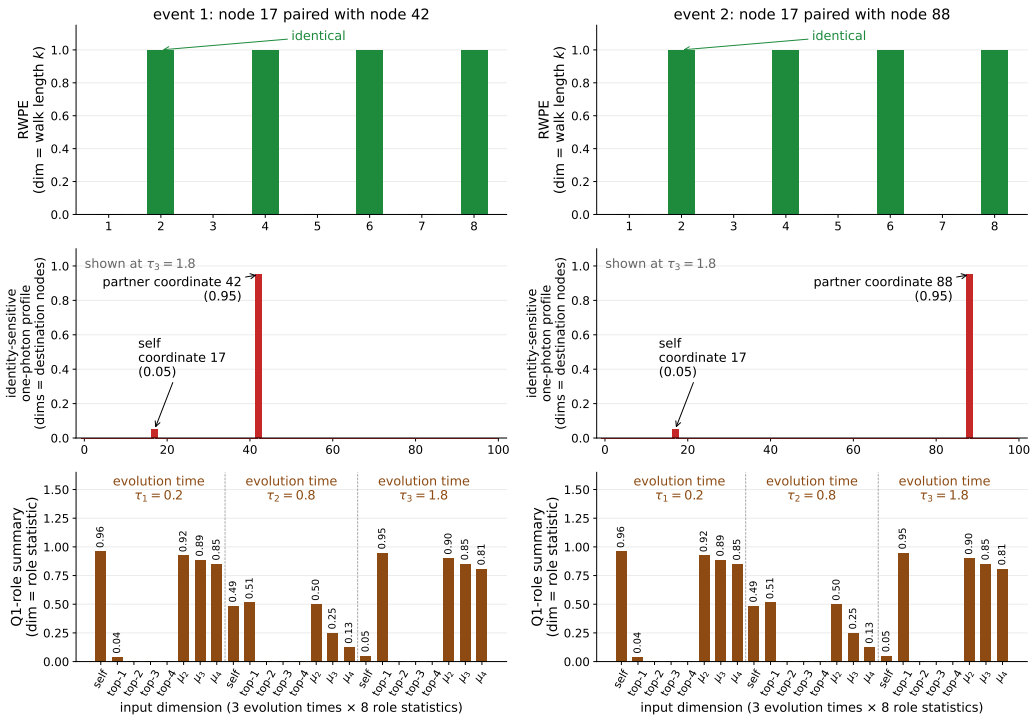


Figure 8: Example underlying Table 16. Node 17 belongs to an isolated edge with node 42 in one snapshot and node 88 in another, at evolution times $\tau = (0.2, 0.8, 1.8)$. Top: RWPE gives the same alternating return probabilities in both snapshots. Middle: at the third evolution time, the original one-photon profile places 0.95 of the probability at the partner's coordinate, which changes between snapshots. Bottom: with $K_{\text{top}} = 4$, the complete 24-dimensional Q1-role vector is identical in the two snapshots while retaining the quantum-walk probabilities and their summaries.

For nodes with at least one neighbor, the classical random walk moves to each neighbor with equal probability. Let P be the matrix used to compute RWPE. For an isolated filler node, we set its entire row in P to zero, so its RWPE vector is also zero. This is an encoding convention for isolated nodes, not the return probability of a walk defined to stay at an isolated node.

The following vectors list the diagonal entries $(P^k)_{vv}$ for $k = 1, \ldots, 6$:

$$\begin{aligned}
&\text{endpoint } v_4: && (0, 0.500, 0, 0.375, 0, 0.312),\\
&\text{endpoint neighbor } v_3: && (0, 0.750, 0, 0.625, 0, 0.542),\\
&\text{middle node } v_2: && (0, 0.500, 0, 0.417, 0, 0.375),\\
&\text{node next to the root } v_1: && (0, 0.417, 0, 0.292, 0, 0.250),\\
&\text{isolated filler node:} && (0, 0, 0, 0, 0, 0).
\end{aligned}$$

For a node on the path, the k-th entry is the probability of being back at that node after exactly k classical steps. For example, the endpoint has return probability 0.500 after two steps and 0.375 after four steps. These entries refer to classical step counts, not to the three quantum evolution times.

Table 16: The same structural event, "node v is in an isolated edge," in two snapshots with different partners $w_1 \neq w_2$, at evolution times $\tau = (0.2, 0.8, 1.8)$. The original one- and two-photon profiles are shown before random projection. Their informative value moves from the coordinate for w_1 to the coordinate for w_2. The role summaries remain unchanged while retaining the numerical values. The two-photon coincidence is near zero at $\tau = 0.8 \approx \pi/4$, where the one-photon probability is split almost equally between the two nodes.

encoding	event 1 features	event 2 features	unchanged
RWPE	$[0, 1, 0, 1, \ldots]$ (return probabilities)	$[0, 1, 0, 1, \ldots]$	yes
one-photon	probability $(0.04, 0.51, 0.95)$ at w_1	same, at w_2	no
Q1-role	self $(0.96, 0.49, 0.05)$, top-1 $(0.04, 0.51, 0.95)$	identical	yes
two-photon	coincidence $(0.85, 0.00, 0.80)$ at w_1	same, at w_2	no
Q2-role	self $(0.08, 0.50, 0.10)$, top-1 $(0.85, 0.00, 0.80)$	identical	yes

The odd-step entries are zero and returning to the starting node requires an even number of steps. The even-step probabilities differ across the listed positions. For example, the endpoint and middle node have the same two-step return probability, 0.500, but different four-step return probabilities, 0.375 and 0.417. Thus, return profiles can distinguish these structural positions without recording the identities of the other nodes visited by the walk. This illustrates the structural information available from classical return probabilities; it does not imply that RWPE and the quantum self-return features of Q1-role are numerically equivalent.

Removing destination-node correspondence can nevertheless discard information needed for prediction. In our results, role summaries show their largest losses on stochastic periodicity and counting, where knowing which specific nodes are involved can help distinguish recurring community assignments or graphs. The trade-off is therefore between recognizing a structural pattern when its participants change and retaining information about those participants. Role summaries support the former, while identity-sensitive encodings retain more information for tracking specific destination nodes across snapshots.

Table 17: Observed mean training time in seconds per epoch on the synthetic tasks. The number of settings in each family is given in parentheses, and settings are weighted equally. Validation, test and input preparation are excluded.

Method	memorization (8)	counting (7)	sto p=0.5 (6)	sto p=0.9 (6)	LR (20)	CE (6)
GC-LSTM	41.394	46.176	24.059	40.175	43.523	38.842
TGCN	35.249	39.093	20.196	38.450	35.365	31.101
GCN	36.007	35.263	19.659	36.426	29.199	26.603
GAT	31.054	34.632	19.443	36.278	31.460	28.392
EvolveGCN	35.210	37.219	20.261	35.539	33.114	25.802
TGAT	42.652	41.452	54.049	95.467	52.161	39.312
TGN	27.460	35.681	24.479	30.569	33.391	34.936
DyGFormer	98.112	239.035	611.863	783.335	129.366	116.047
CTAN	23.328	16.779	15.848	33.880	24.054	22.791
One-photon	4.005	6.827	1.847	2.933	8.976	7.810
Two-photon	7.655	7.678	2.625	3.188	9.449	8.486
Q1-role	7.423	7.515	3.123	3.592	9.573	8.352
Q2-role	6.851	7.017	2.629	3.906	9.089	8.773
LapPE	7.415	6.576	2.108	3.303	9.540	8.394
RWPE	5.505	5.600	2.014	3.394	9.085	8.026

F Implementation and experimental setup

F.1 Environmental setup

Compute environment. Experiments were distributed over several computing clusters, so the software versions and GPU models differ slightly between runs. Each run used CPU cores or a single GPU. Our encoding pipelines ran with PyTorch 2.9 or 2.10, mostly on NVIDIA L40S GPUs and otherwise on NVIDIA RTX 8000 GPUs or CPU cores. The temporal GNN baselines ran with the official T-GRAB code on PyTorch 2.0 or 2.9 and PyTorch Geometric 2.6, mostly on NVIDIA L40S GPUs and otherwise on NVIDIA V100, A100 or H100 GPUs or CPU cores. Simulator and hardware features were collected through Quandela's cloud platform with Perceval 1.1.

Temporal readout and training. The readout is a single-layer LSTM shared across nodes, followed by a two-layer MLP that scores a candidate pair from the LSTM states of its two endpoints. It is trained with binary cross-entropy using AdamW, and the test split is scored once with the weights of the best validation epoch. The default configuration uses an LSTM and MLP width of 512, dropout 0.2, learning rate 2×10^{-3}, weight decay 10^{-4}, 600 epochs and validation every ten epochs.

Simulator and hardware. Simulator and hardware features are obtained from Quandela's Belenos backends (Appendix D). Training and validation always use analytic features; only the test features are replaced.

F.2 Runtime

Tables 17 to 19 report training time per epoch for the 53 synthetic settings and 11 real-data settings, comprising nine interaction networks and two temporal aggregations of the baboon contact network, and Table 20 reports the one-time preparation time of the encoding pipelines. To compare all methods under the same conditions, every method was timed on NVIDIA H100 GPUs with PyTorch 2.9. Each timing run trains for five epochs on the full training data, and we report the mean over the five epochs; validation, testing and input preparation are excluded. Synthetic settings are averaged within each task family. Persistence and EdgeBank have no training epochs and are not listed.

Table 18: Observed mean training time (seconds/epoch) on the real-data settings.

Method	Eu Dept1	Eu Dept2	Eu Dept3	Eu Dept4	MIT Reality	Radoslaw
GC-LSTM	23.895	18.442	12.772	17.909	7.622	9.841
TGCN	22.602	17.537	11.876	13.061	8.118	10.749
GCN	22.867	18.364	9.924	14.085	6.312	8.492
GAT	22.896	15.703	10.621	12.671	6.889	6.479
EvolveGCN	23.108	17.713	9.951	12.377	4.644	7.056
TGAT	5.346	23.379	4.269	19.044	10.900	5.091
TGN	3.679	11.342	5.819	7.802	7.876	9.038
DyGFormer	13.490	158.150	94.576	198.312	108.319	180.944
CTAN	1.969	6.230	5.577	4.919	2.039	3.590
One-photon	0.881	0.190	0.166	0.198	0.493	0.363
Two-photon	0.952	0.372	0.641	0.474	0.303	0.129
Q1-role	0.968	0.351	0.199	0.490	0.200	0.140
Q2-role	0.804	1.077	0.606	0.678	0.274	0.221
LapPE	1.404	1.011	0.657	0.759	0.154	0.271
RWPE	1.083	0.623	0.929	0.640	0.511	0.182

Table 19: Observed mean training time (seconds/epoch) on the real-data settings.

Method	SocialEvo	Baboons 5 min	Baboons 15 min	tgbn-trade	tgbn-genre
GC-LSTM	2.202	88.032	26.462	2.008	303.406
TGCN	1.968	62.869	22.197	1.868	296.026
GCN	1.824	65.888	23.354	1.843	287.884
GAT	1.872	61.016	20.297	1.926	297.760
EvolveGCN	1.909	66.048	21.584	1.975	301.408
TGAT	3.666	95.837	34.221	9.387	258.199
TGN	1.467	57.668	21.866	1.445	26.433
DyGFormer	12.299	165.296	68.636	39.475	494.768
CTAN	0.976	43.312	16.450	1.075	16.776
One-photon	0.163	9.263	3.366	0.284	2.943
Two-photon	0.311	9.439	2.778	0.149	2.282
Q1-role	0.275	9.359	2.853	0.153	2.562
Q2-role	0.300	9.374	2.140	0.118	2.456
LapPE	0.140	9.700	2.920	0.234	2.311
RWPE	0.461	7.227	2.017	0.212	2.372

Table 20: One-time preparation time of the encoding pipelines (seconds), including startup, imports, data loading, feature construction and model initialization. Synthetic entries weight the settings of a family equally.

Task / dataset	One-photon	Two-photon	Q1-role	Q2-role	LapPE	RWPE
memorization	30.274	44.073	36.590	48.701	12.392	12.244
counting	16.140	29.988	21.549	34.051	12.933	12.240
sto p=0.5	26.120	28.164	22.938	23.048	26.807	22.958
sto p=0.9	38.001	45.478	39.507	37.362	33.866	35.795
LR	60.715	78.353	66.999	81.494	43.861	25.872
CE	54.633	72.675	61.691	77.495	42.268	24.805
Eu Dept1	17.511	17.145	15.690	15.034	27.904	7.371
Eu Dept2	13.108	13.727	12.944	8.199	7.357	7.977
Eu Dept3	11.676	11.955	10.923	11.526	9.311	8.206
Eu Dept4	11.930	12.600	12.346	11.256	8.150	7.450
MIT Reality	7.573	10.935	9.838	10.144	8.466	6.590
Radoslaw	6.871	9.357	8.601	8.238	7.700	6.804
SocialEvo	6.943	12.051	11.407	11.451	9.212	7.069
Baboons 5 min	49.552	80.932	65.713	86.005	23.443	24.602
Baboons 15 min	24.276	32.569	23.838	31.539	12.496	13.307
tgbn-trade	7.687	10.830	6.840	9.718	7.813	7.583
tgbn-genre	44.504	628.739	37.633	629.413	29.229	31.760