

Journal Pre-proof

Time-domain-multiplexed Gaussian Boson sampling for graph problems in finance

Shuo He, Boxuan Ai, Pengfei Gao, Hongbao Liu, Jun-Jie He, Ke-Ming Hu, Yu-Ze Zhu, Jie Wu



PII: S2709-4723(25)00055-3

DOI: <https://doi.org/10.1016/j.chip.2025.100181>

Reference: CHIP 100181

To appear in: *Chip*

Received Date: 11 September 2025

Revised Date: 11 December 2025



Accepted Date: 15 December 2025

Please cite this article as: He S, Ai B, Gao P, Liu H, He JJ, Hu KM, Zhu YZ, Wu J, Time-domain-multiplexed Gaussian Boson sampling for graph problems in finance, *Chip*, <https://doi.org/10.1016/j.chip.2025.100181>.

This is a PDF of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability. This version will undergo additional copyediting, typesetting and review before it is published in its final form. As such, this version is no longer the Accepted Manuscript, but it is not yet the definitive Version of Record; we are providing this early version to give early visibility of the article. Please note that Elsevier's sharing policy for the Published Journal Article applies to this version, see: <https://www.elsevier.com/about/policies-and-standards/sharing#4-published-journal-article>. Please also note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2025 The Author(s). Published by Elsevier B.V. on behalf of Shanghai Jiao Tong University.

Time-domain-multiplexed Gaussian Boson sampling for graph problems in finance

Shuo He^{1,2}, Boxuan Ai² , Pengfei Gao², Hongbao Liu², Jun-Jie He^{3*} , Ke-Ming Hu^{3*}, Yu-Ze Zhu^{3*}, Jie Wu^{1*}

¹Fudan University, Shanghai, China. ²China Unionpay, Shanghai, China.

³TuringQ Co., Ltd., Shanghai, China.

*E-mails: hejunjie@turingq.com (Jun-Jie He), hukeming@turingq.com (Ke-Ming Hu), zhuyuze@turingq.com (Yu-Ze Zhu), jwu@fudan.edu.cn (Jie Wu)

Cite as:

<http://doi.org/10.10xx/x.xxx>

Received: 20 December 2021

Accepted: 31 December 2021

Published online: 4 January 2022

Gaussian Boson sampling (GBS) has emerged as a promising platform for demonstrating quantum computational advantage. Here, we apply GBS to address classically intractable graph problems in financial security. We demonstrate algorithms for: (1) Community structure analysis for market anomaly screening; (2) Synthetic identity fraud detection via hierarchical dense subgraph identification; (3) Collusive stock manipulation detection through clique enumeration; and (4) Anti-money laundering through isomorphic subgraph matching. Notably, we design non-universal time-domain-multiplexed photonic quantum circuits to effectively map these GBS-based algorithms with a quadratic reduction in hardware complexity, achieving over 99% fidelity in fitting the target unitary matrix. This spatial-temporal hybrid architecture and a hardware-algorithm co-design framework establish a promising approach to complex financial graph analytics. Our work bridges the fields of quantum computing and finance, highlighting GBS as a near-term solution to critical industry challenges.

Financial systems globally face escalating threats from money laundering (ML), a sophisticated criminal practice where illicit proceeds are concealed, disguised, and converted to obscure their unlawful origins, thereby integrating them into the formal economic system as seemingly legitimate funds¹. ML operations are typically orchestrated by organized syndicates, characterized by covert and intricate tactics. Perpetrators exploit multiple identities and accounts to execute transactions, with their methods evolving continuously to evade detection. These activities often follow a hierarchical structure corresponding to the three-stage laundering process: placement, layering, and integration. Despite extensive efforts, traditional anti-money laundering (AML) frameworks² struggle to detect such patterns effectively. Rule-based systems, reliant on expert experience, suffer from severe inefficiencies in manual review, leading to frequent issues of missed reports, false positives, and delayed alerts, which hinder the automatic identification of potential ML features in massive transaction datasets. Supervised machine learning models, while promising, depend on high-quality labeled samples, yet the scarcity of blacklist samples (compared to normal samples) in practice results in severe class imbalance, compromising model accuracy³. Community detection algorithms in graph computing⁴, a key method for identifying ML syndicates by rec-

ognizing densely connected subgraphs with sparse external connections, struggle with high time complexity in processing complex, irregular financial transaction networks, failing to meet the accuracy and real-time requirements of AML detection.

The computational intractability of these financial problems is rooted in their NP-hard nature. It is precisely this class of challenges that quantum computing has the potential to address. Quantum computing has evolved from a theoretical concept to an experimental reality, with quantum advantage now demonstrated in specific computational tasks⁵⁻¹¹. Among these, sampling problems such as Boson sampling present formidable challenges to classical simulators¹², yet can be efficiently implemented on photonic processors. As an enhanced variant, Gaussian Boson sampling (GBS) utilizes squeezed states instead of single photons, offering greater scalability¹³. Recent breakthroughs⁹⁻¹¹ have established GBS as a leading platform for quantum computational advantage and confirmed that time-bin encoding with time-delay interferometers provides critical scalability by resource-efficient mode multiplexing, though achieving large-scale universal programmability remains limited under realistic loss constraints. This technological progress has stimulated GBS applications in graph theory^{14,15}, particularly for NP-hard problems like dense subgraph identification¹⁶⁻¹⁸ and maximum clique detection^{19,20}. Meanwhile, more advanced classical algorithms continue to emerge, improving the efficiency of GBS simulations and solving graph problems—some even challenging the evidence of quantum advantage in noisy near-term devices. Specifically, Oh *et al.* introduced a quantum-inspired classical algorithm for graph problems, utilizing a distribution that can be efficiently sampled by two-photon Boson sampling²¹. Furthermore, they employed a tensor-network-based method to effectively simulate lossy GBS, challenging the claims of quantum advantage established in previous experiments²². Recently, Zhang *et al.* proposed a double-loop variant of Glauber dynamics was proposed to sample from GBS distributions on unweighted graphs, proving it mixes in polynomial time on dense graphs²³. As a result, the boundary of quantum advantage is continuously being reshaped by these experimental and theoretical advances. Despite rapid progress in the field, however, the potential of GBS for financial graph analytics—where massive transaction networks demand real-time analysis of combinatorially complex patterns—remains largely unexplored.

Here, we establish a co-design framework connecting GBS algorithms to time-domain-multiplexed (TDM) photonic hardware. While our TDM architectures sacrifice universality for scalability, we demonstrate that financial graph problems could be solved with non-universal, albeit programmable time-delay interferometers. In this work, we: (i) develop novel GBS-based algorithms targeting critical financial fraud scenarios—community structure analysis and synthetic identity fraud via dense subgraph detection, collusion screening via clique enumeration, and AML through isomorphic subgraph matching; (ii) propose an efficient spatial-temporal hybrid architecture, designed for scalability, to balance circuit

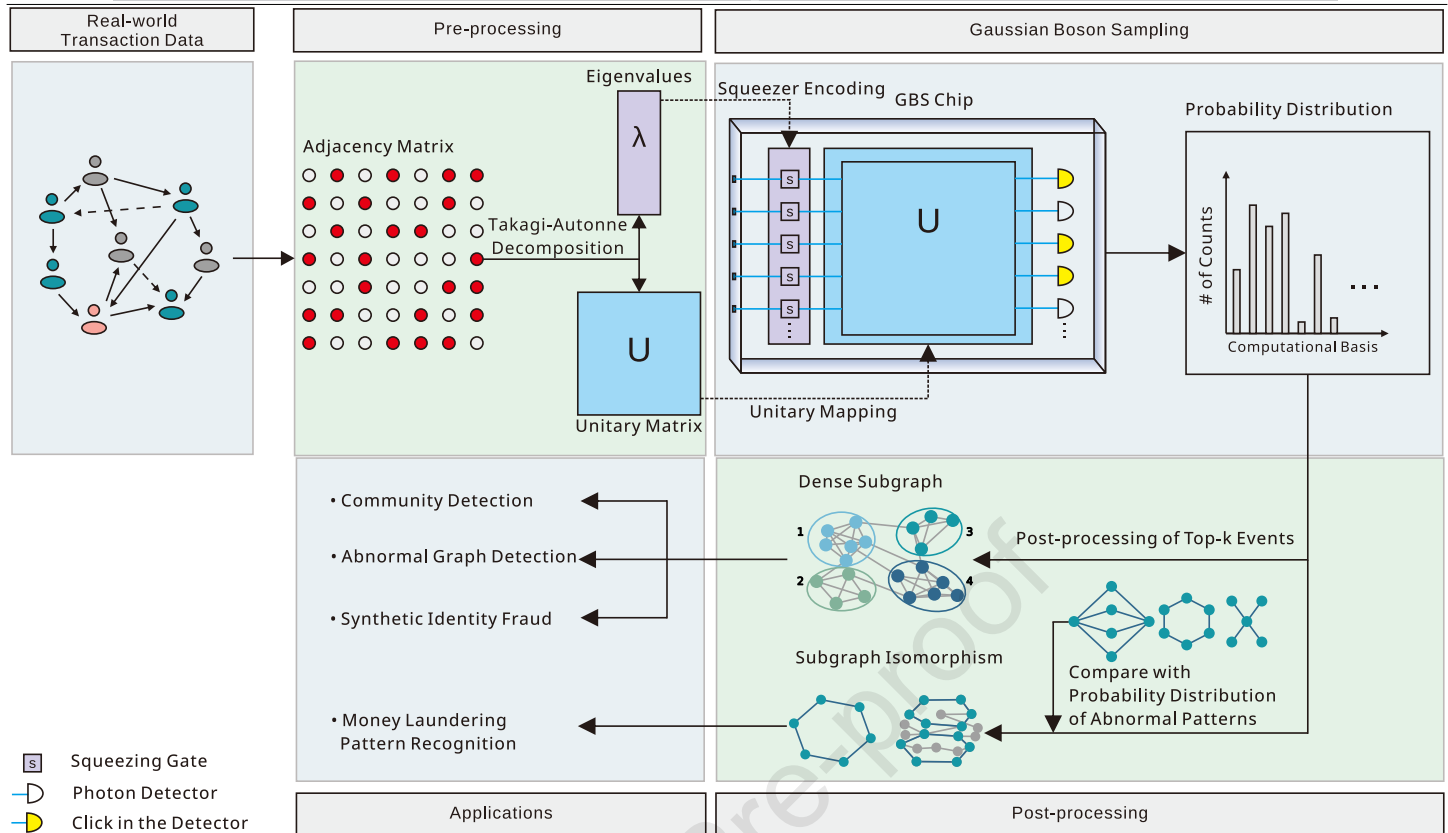


Figure 1 | Workflow of GBS-based algorithms for solving financial graph problems. Our methodology consists of three sequential stages: pre-processing, GBS encoding and sampling, and post-processing. Initially, transaction data is converted into a graph representation, generating an adjacency matrix. This matrix is then encoded onto a GBS device, and repeated sampling of the output state produces a probability distribution. Finally, this distribution undergoes post-processing to identify dense subgraphs whose structures are strongly correlated with fraudulent financial activities.

expressiveness with hardware implementation friendliness; and (iii) validate the effectiveness of this architecture in solving the targeted financial graph problems through numerical simulations leveraging DeepQuantum, a PyTorch-based software platform for quantum machine learning and photonic quantum computing²⁴. This synergy between problem-aware algorithms and hardware-aware mappings positions GBS as a highly promising near-term candidate for tackling computationally intractable problems in financial network analytics.

The workflow of GBS-based algorithms for fraud detection is executed in three distinct stages, as outlined in Fig. 1. The first stage involves pre-processing the financial data, wherein the data is modeled as a graph and its adjacency matrix is computed. In the second stage, the adjacency matrix is encoded to a GBS device by tuning the squeezing and interferometer parameters (see Methods for more details). Sampling the device outputs a probability distribution of potential subgraphs. The final stage consists of post-processing this probability distribution to isolate and identify anomalous subgraphs that are patterns of financial fraud scenarios.

We demonstrate the efficiency of our GBS-based algorithms by applying them to the detection of four distinct types of fraudulent financial activities. Community structure analysis aims to identify groups of nodes that are more densely connected internally than to the rest of the network. Classical approaches include the Girvan-Newman algorithm based on edge betweenness²⁵, as well as the Louvain and Leiden methods that efficiently maximize modularity for large-scale networks²⁶. In the financial area, such structures are often red flags for coordinated illicit activity. For instance, a hidden community might represent a group of colluding traders, or an insurance fraud scheme. Synthetic identity fraud is often perpetrated

by linking new, synthetic accounts to stolen, legitimate Personally Identifiable Information (PII) like addresses or phone numbers. When modeled as a bipartite graph representation connecting accounts to PII, the fraudulent collusion naturally emerges as highly dense subgraphs²⁷. A representative classical method for dense subgraph detection is Charikar's greedy peeling algorithm²⁸, which offers an approximated solution in polynomial time. For exact solutions, Goldberg's max-flow-based formulation provides a theoretically optimal approach²⁹. Collusion screening refers to stock market manipulation by fraudulent trades in a transaction network where traders are nodes. Such collusion often manifests as the formation of cliques. As noted in ref.³⁰, manipulated trading networks tend to exhibit a higher number of cliques. The classical Bron-Kerbosch algorithm uses depth-first expansion of candidate sets to enumerate all maximal cliques³¹. The Carraghan-Pardalos branch-and-bound algorithm is another widely adopted exact method for maximum clique search³². AML focuses on identifying specific, predefined transaction patterns or typologies that indicate illegal money flows in the transactional datasets graph. Unlike the previous scenarios that rely on finding generic dense subgraphs, detecting these anomalies requires isomorphic subgraph matching³³. Ullmann's classic backtracking algorithm with matrix-based pruning laid the foundation for the early research³⁴. Subsequently, the VF2 algorithm introduced more restrictive feasibility rules, achieving substantial practical efficiency³⁵.

Most classical methods suffer from severe search-space explosion on large graph problems. However, the GBS-based algorithm naturally highlights structurally dense and pattern-rich regions of the graph, this bias offers an effective way to pre-select promising candidate subgraphs, thereby

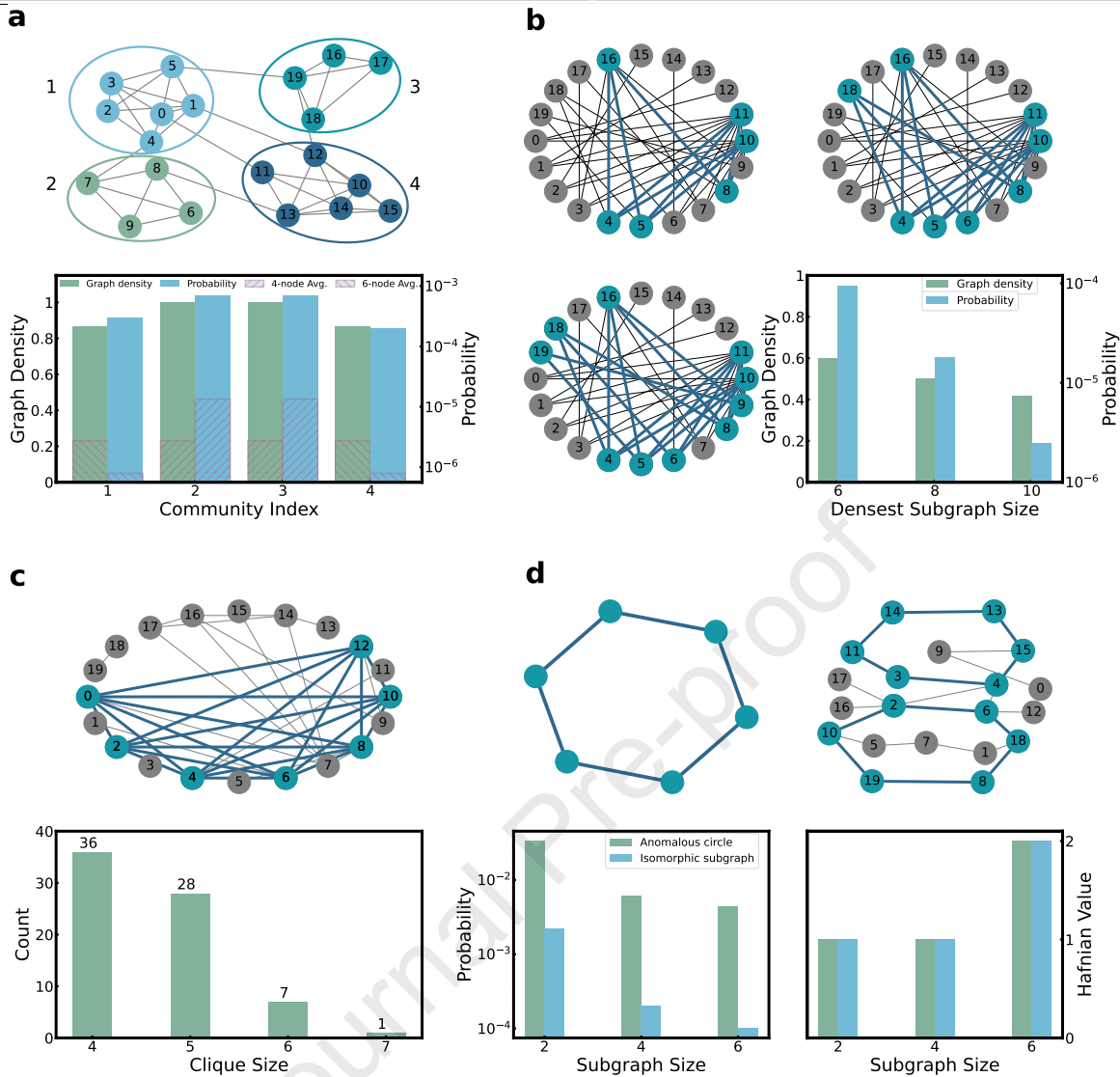


Figure 2 | Detection of four types of fraudulent financial activities with GBS-based algorithms. **a**, Community structure analysis. The upper part visualizes a financial network with four distinct communities detected by our GBS-based algorithm. The bar chart below demonstrates a strong correlation between the intrinsic graph density of each community (green bars) and the GBS detection probability (blue bars). The shaded bars provide a statistical baseline for comparison, showing the average density and probability expected from 10,000 randomly sampled 4-node and 6-node subgraphs, respectively. **b**, Synthetic Identity Fraud Detection. Our GBS-based algorithm identifies the densest subgraphs of sizes 6, 8, and 10 (highlighted in teal) in a bipartite graph. The corresponding bar chart (lower right) confirms that the GBS probability (blue) serves as an effective proxy for graph density (green), enabling the efficient identification of fraudulent collusion. **c**, Clique-based Collusion Detection. The upper part shows the full network graph with the largest detected 7-node clique highlighted in teal, pinpointing a highly suspicious group of colluding traders; the lower bar chart shows the number of cliques of sizes 4–7 found by GBS in the trading network. **d**, AML via Isomorphic Subgraph Matching. The upper left part is a predefined 6-node cycle pattern for AML detection, and the upper right part displays two isomorphic subgraphs identified in the 20-node target graph. A direct comparison of raw GBS probabilities (lower left bar chart) is insufficient for matching. However, comparing the underlying Hafnian values (lower right bar chart) provides a true structural fingerprint.

significantly reducing the search space for the graph problem. The community structure analysis, synthetic identity fraud detection, and collusion screening scenarios are characterized by abnormal dense subgraph detection. Theoretical¹⁶ and experimental³⁶ works demonstrate that the Hafnian of a graph's adjacency matrix, which is equal to the number of perfect matchings in the graph, serves as a quantitative link to graph density. This correlation allows us to leverage statistical outcomes of GBS to identify dense patterns within financial networks, as illustrated in Fig. 2. In Fig. 2a, we present GBS-enhanced community detection results, where the GBS probability for each of the four detected communities clearly correlates with its graph density. Notably, the densities and probabilities of these communities are significantly higher than the statistical baseline (shaded

bars) established from randomly sampled subgraphs. In Fig. 2b, the GBS probability again serves as a reliable proxy for identifying the densest subgraphs with 6, 8, and 10 nodes in the synthetic identity task. As for collusion detection tasks, GBS efficiently identifies 36 four-node cliques, 28 five-node cliques, 7 six-node cliques, and 1 seven-node clique in Fig. 2c, which are strong indicators of coordinated stock market manipulation. For AML detection, our approach employs isomorphic subgraph matching as introduced in ref.³⁷. While two isomorphic graphs produce identical subgraph probabilities under GBS, the observed probability distribution of a subgraph embedded within a larger graph differs from its isolated sampling probability distribution due to a scaling factor. Crucially, however, their Hafnian values remain identical despite this scaling disparity. This

necessitates a more sophisticated matching protocol than direct probabilistic comparison. Our detection workflow comprises: (1) Pre-compute the Hafnian-scaled probabilistic signature of a known anomalous graph (e.g., cycle graph or money-laundering typology); (2) Perform GBS on the target transaction graph; (3) Identify fraudulent subgraphs by matching Hafnian-based signatures against pre-computed patterns, thereby efficiently filtering non-isomorphic candidates. As demonstrated in Fig. 2d, this GBS-based algorithm successfully located two 6-node cycle subgraphs isomorphic to the pattern graph. Notably, all exhibited identical Hafnian values (confirming isomorphism) despite variations in their detection probabilities.

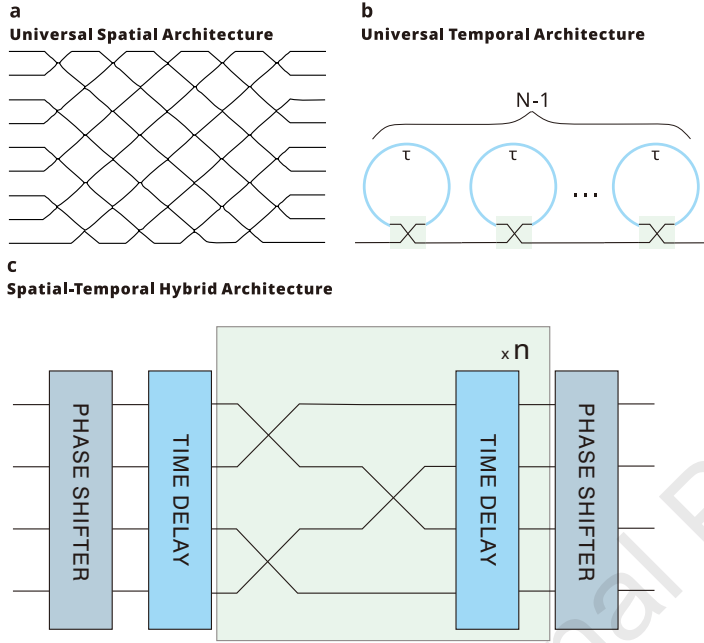


Figure 3 | Different architectures of programmable photonic circuits. a. Spatial Clements architecture for implementing arbitrary unitary transformations. **b.** Temporal chain-loop architecture for implementing arbitrary unitary transformations. **c.** The spatial-temporal hybrid architecture proposed in this work, implementing unitary transformations by integrating spatial and temporal blocks in an iterative pattern.

While the proposed GBS algorithms theoretically provide powerful solutions for tackling the scale and complexity inherent in real-world financial graph problems, their practical implementation on near-term photonic quantum systems faces a significant challenge: scaling the hardware to efficiently map and execute these algorithms on graphs of relevant size. To bridge this gap between algorithmic potential and hardware limitations, we propose an efficient hybrid architecture and a co-design framework. Specifically, scaling GBS hardware to the required number of optical modes (N) for practical financial graphs presents fundamental challenges in mitigating optical loss and achieving full reconfigurability. Conventionally, programmable photonic circuits are realized as spatial multi-port interferometers^{38,39}, constructed from a mesh of beam splitters and phase shifters. For instance, the Clements architecture implements arbitrary unitary transformations on an N -mode input state (Fig. 3a). Despite its powerful expressivity, the Clements architecture exhibits a critical scaling limitation: the depth scales linearly with the number of modes, and the number of programmable elements scales quadratically. This scaling presents a significant bottleneck, rendering large-scale implementations with substantial hardware overhead, even on integrated photonic platforms.

To overcome the scalability limitations inherent in such spatial architectures, alternative designs that exploit temporal multiplexing have been developed. Encoding information in photonic time bins allows for a significant reduction in the physical component count while preserving full programmability. A prominent architecture in this domain utilizes a series of optical delay lines to time-multiplex a small set of optical components^{36,40–42}, as shown in Fig. 3b. The construction of an arbitrary $N \times N$ linear transformation, however, requires the optical signal to perform $O(N)$ passes through the whole circuit.

In this work, we present a novel TDM circuit architecture, as illustrated in Fig. 3c, that is designed to achieve both $O(\sqrt{N})$ width and $O(\sqrt{N})$ depth, targeting a substantial reduction in hardware complexity. This hybrid architecture strategically integrates spatial (e.g., shallow interferometer layers) and temporal blocks (e.g., optical delay loops) in an iterative pattern. The initial and final phase shifter layers enable flexible phase adjustment column-wise and row-wise in the target unitary matrix. By reusing a small spatial circuit over time, it overcomes the non-scalability of pure spatial Clements. Furthermore, by processing multiple spatial modes in parallel at each timestep, the architecture requires fewer passes and less overall computation time than purely temporal schemes. This architecture can be configured to implement unitary transformations through sequentially applying distinct spatial operations at each timestep.

This favorable quadratic reduction is achieved at the cost of universality. However, we demonstrate that the architecture enables efficient encoding of target graphs while yielding output probability distributions with sufficient fidelity for solution extraction. Critically, we implement a co-design paradigm where: (i) Algorithm-hardware mapping employs automatic differentiation with gradient-based optimization to fit target unitaries while exploring optimal time-domain subspaces of TDM circuits; and (ii) Circuit design is tailored to specific problem structures for enhanced fitting performance.

To validate this approach, we benchmark our architecture on a practical task: fitting the 20×20 unitary matrix U derived from the adjacency matrix A of a collusive stock manipulation graph. For this 20-node problem, we implement the specific architecture with 4 spatial blocks (each implemented as two-layer alternating Mach-Zehnder interferometer (MZI) meshes) and 5 temporal blocks (each containing one layer of delay loops with $n_\tau = 1$). We use fidelity to evaluate the fitting performance of the designed TDM circuit with different timesteps (see Methods for more details). For comparison, we also present the results of simplified TDM circuits with 1–3 spatial modes. The best fidelities of each timestep are shown in Fig. 4a. The performance clearly degrades as the number of modes decreases. The 4-mode TDM circuit achieves the best performance, reaching a peak fidelity of over 0.99, showing the strong capability for fitting the target unitary matrix. However, the fidelity reached saturation with the increasing timestep. Since the periodic structure is added with more timesteps, the circuit has reached the optimal performance limit for the large timestep. In Fig. 4b, we show the fitted 20×20 unitary matrix embedded in the global unitary matrix using the 4-mode TDM circuit with a timestep equal to 10, as highlighted by the red box. The evolution of the TDM circuits results in a global circuit with 60 modes, including 40 physical modes and 20 modes in delay loops. The column indices 5, 11, 17, and 23 to 39 indicate the input position of the squeezed light, and the row indices 40 to 59 indicate the detected output position of the global circuit.

To demonstrate a practical application, we then use our best TDM circuit to solve a 6-node clique-finding problem on the given 20-node graph. We need to pick the Fock states containing 6 single photons out of 20 modes (38760 in total), then sample from the GBS device and choose those states with high probabilities. Fig. 4c compares the output probability distribution from the spatial GBS implementation (blue bars) with our TDM GBS (green bars). We show 50 output states with the highest probabilities, sorted in descending order according to the ideal case. For the ideal spatial GBS, it is evident to see that the first 7 states, corresponding to the graph's

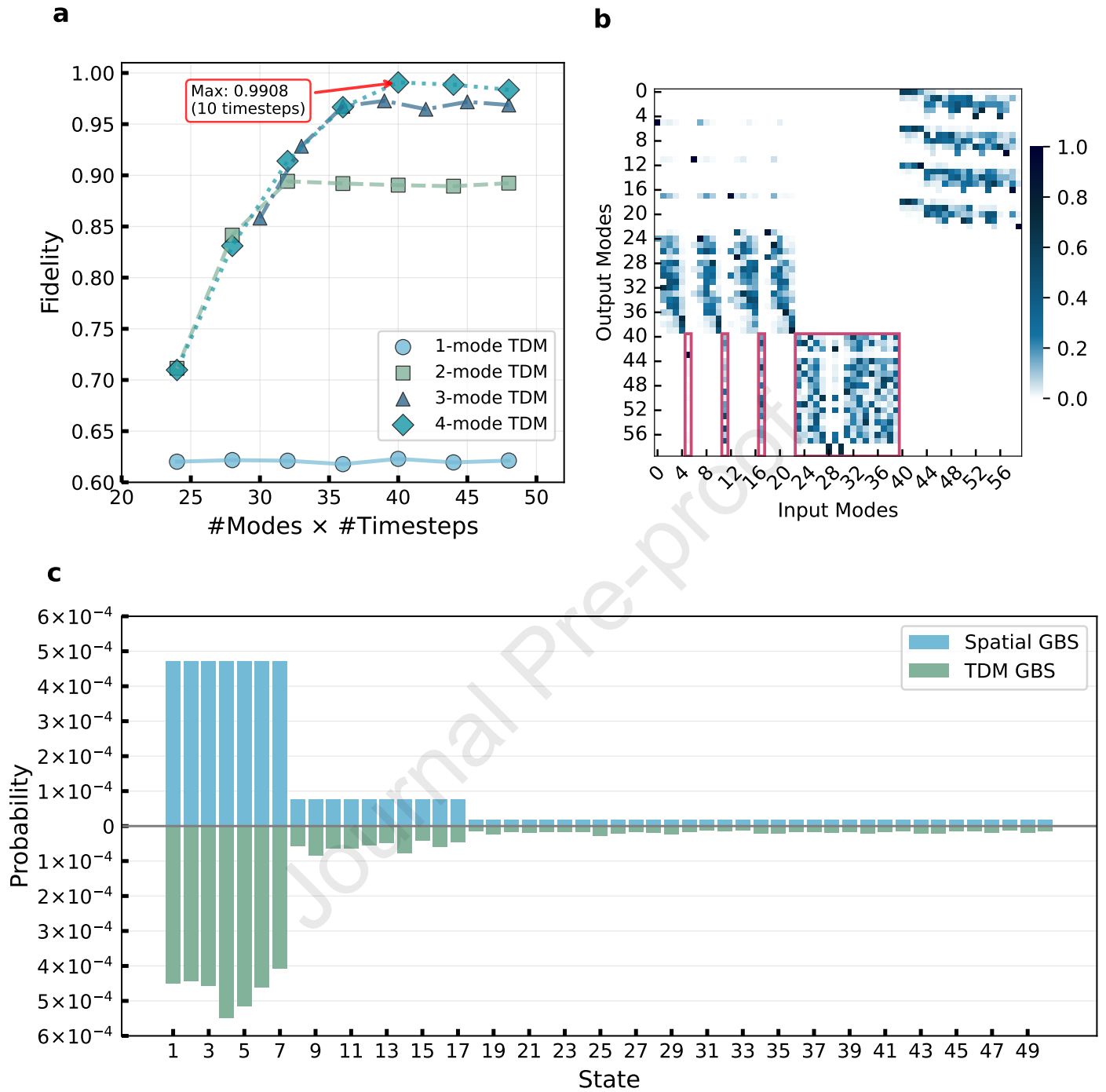


Figure 4 | Fitting the target unitary matrix with TDM circuits and the GBS probability distribution comparison. **a**, Fidelity of unitary matrix fitting. The fidelity for fitting a target 20×20 unitary matrix is plotted as a function of TDM circuit depth (# of modes \times # of timesteps) for four different TDM circuit configurations. The 4-mode circuit demonstrates the capability of fitting 20×20 unitary matrix, reaching a peak fidelity of 0.9908 with 10 timesteps. **b**, Global unitary matrix and the fitted unitary matrix visualization. The absolute values of the global unitary matrix generated by the 4-mode TDM circuit are displayed. The fitted 20×20 unitary transformation is embedded within this larger matrix, as highlighted by the red box, illustrating how the desired input and output operation is realized within the unrolled TDM circuit evolution. **c**, GBS probability distribution for clique finding. This plot compares the output probability distributions for a 6-node clique-finding problem on a 20-node graph. The probabilities for the top-50 output states are shown, sorted according to the ideal case. The blue bars represent the ideal distribution from a fully spatial 20-mode GBS device. The green bars show the distribution from our 4-mode TDM circuit with a timestep equal to 10. Due to the similar probability distribution, the same 7 states corresponding to cliques are still identifiable with our TDM GBS.

seven actual 6-node cliques, share a uniformly high probability of approximately 5×10^{-4} . Despite of imperfect unitary mapping of the TDM

circuit, which causes variations in the output probabilities for these states, the overall probability distribution shows a strong resemblance to the ideal

one. Crucially, the same 7 states corresponding to cliques are still identifiable due to their high relative probabilities. By deliberately trading off universality for scalability and tailoring the circuit design to the problem structure, we show that a compact 4-mode TDM circuit can accurately emulate a full 20-mode unitary transformation—effectively bridging the gap between spatially constrained hardware and real-world problem sizes.

Furthermore, as optical loss is a critical factor limiting fidelity in current imperfect devices, our architecture is expected to maintain higher performance compared to deeper universal designs. **To validate the statement, we evaluated the robustness of these two architectures against optical loss by conducting a noise analysis using three different matrix sizes: 36×36 , 64×64 and 100×100 . For the spatial Clements architecture, we assumed a homogeneous transmittance for each MZI. For our spatial-temporal hybrid architecture, we additionally incorporated the loss from the delay loop, using the same transmittance value. Its lossless fidelity is obtained by a 1000-step optimization. To ensure statistical reliability, we randomly generated 10 different unitary matrices and averaged the performance for each matrix size. The comparative results are presented in Fig.5. The results clearly indicate that our spatial-temporal hybrid architecture maintains significantly higher fidelity with increasing optical loss, demonstrating superior robustness compared to the spatial Clements mesh.**

These results strongly suggest that such a co-design framework and hybrid architecture make it possible to tackle large-scale graph problems on near-term non-universal photonic quantum devices with limited spatial modes. To scale our approach for larger problems, it is essential to ensure sufficient parameterization in the circuit for high-fidelity unitary approximation. The fitting complexity of a target unitary can be diagnosed, for instance, by performing a Clements decomposition and evaluating the number of internal phase shifters with angles close to π —indicating MZIs operating near the bar state. If the initial fitting is inadequate, the architecture's spatial and temporal blocks can be adaptively reconfigured, such as by adding more MZI layers within the spatial blocks.

In summary, this work establishes a co-design framework that connects GBS algorithms with scalable TDM photonic hardware to address computationally intractable graph problems critical to financial security. We demonstrated GBS-based algorithms targeting key financial fraud scenarios: community structure analysis for market anomaly screening, synthetic identity fraud detection via hierarchical dense subgraph identification, collusive stock manipulation detection through clique enumeration, and anti-money laundering via isomorphic subgraph matching. Crucially, we proposed and validated an efficient spatial-temporal hybrid photonic architecture, leveraging programmable time-delay interferometers. This architecture, designed for scalability by sacrificing universality, achieves an effective balance between expressiveness and hardware friendliness. Numerical simulations using the DeepQuantum framework demonstrated the effectiveness of this approach, achieving over 99% fidelity in fitting a target 20×20 unitary matrix. Scaling this approach to larger problems represents a natural and critical future direction, one that will demand more powerful quantum devices and greater classical computational resources. Nonetheless, by bridging quantum algorithm design and hardware mapping, this work positions GBS as a highly promising near-term quantum platform for tackling complex financial network analytics challenges, paving the way for practical quantum advantage in finance.

METHODS

Datasets Given the proprietary and confidential nature of real-world financial data, this study employs synthetic and small-scale graph datasets for method validation and illustration. In Fig. 2a, the corresponding graph is synthetically generated. In this graph, each node can be interpreted as a bank account, and each edge represents a transaction between two accounts. Four distinct communities indicate groups of accounts with frequent internal transactions in a larger financial network. In Fig. 2b, a 20-node subgraph containing hierarchi-

cal dense subgraphs is selected from the financial network dataset presented in ref. ²⁷. The original dataset is constructed as a bipartite graph with account nodes and PII nodes (e.g., email or physical address). An edge indicates that an account is associated with a specific PII. From the constructed graph, a subgraph with 29,851 account nodes and 61,159 PII nodes is used in ref. ²⁷. In Fig. 2c, the corresponding graph is synthetically generated to simulate the manipulated stock market transaction. Each node represents an investor in the stock market, and the edge represents the highly correlated trading behaviors between two investors. The small, fully connected subgraphs are strong topological indicators of coordinated group activity and potential market manipulation. In ref. ³³, the synthetic banking transaction dataset is created using the AMLSim multi-agent simulator⁴³ with nodes $n \in \{8, 16, 32, 64, 128\} \times 10^3$, containing seven known money laundering topologies such as cycle, fan-in, and fan-out. To give a simple demonstration of AML via isomorphic subgraph matching, we extract a 20-node subgraph containing a 6-node cycle from the dataset with 8,000 nodes, as shown in Fig. 2d. In Fig. 4, the encoded 20-node graph example comes from the same graph in Fig. 2c.

Mapping a graph onto GBS device To encode a graph into the GBS device, we map its adjacency matrix A onto the GBS device's physical parameters using the methodology established in ref. ¹⁴. This encoding is achieved via the Takagi-Autonne decomposition, which factorizes any complex symmetric matrix. For a real, symmetric adjacency matrix A , the decomposition is:

$$A = U \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m) U^T. \quad (1)$$

Here U is the target unitary matrix for the GBS device, and λ_i determines the squeezing parameter r_i via the relation $\tanh(r_i) = \lambda_i$. In practice, to keep the total mean photon number \bar{n} within a physically achievable range, we scale the λ s by a factor c such that

$$\bar{n} = \sum_1^M \frac{(c\lambda_i)^2}{1 - (c\lambda_i)^2}. \quad (2)$$

After this encoding, the probability of detecting a specific output Fock state $S = (s_1, s_2, \dots, s_m)$ from the GBS device is given by

$$\text{Pr}(S) = p_0 \cdot c^k \frac{|\text{Haf}(A_s)|^2}{s_1! \dots s_m!}. \quad (3)$$

Here $k = \sum_i s_i$, and p_0 refers to the vacuum probability. A_s is the submatrix of the adjacency matrix A . It is constructed by repeating the i th row and column of A exactly s_i times. When $s_i \in \{0, 1\}$, the submatrix A_s is precisely the adjacency matrix of the subgraph induced by the nodes where photons were detected. This directly links the GBS output probability to the Hafnian of a subgraph's adjacency matrix, forming the foundation for solving graph problems with GBS.

For isomorphic subgraph matching, if the pattern graph has an odd number of nodes or the Hafnian of its adjacency matrix is zero, a common approach is to introduce auxiliary nodes in both the pattern and target graphs. The auxiliary nodes are connected to all existing nodes within their respective graphs, thereby enabling the application of the aforementioned mapping method for sampling. During post-processing, only subgraphs that include all auxiliary nodes are considered as candidates.

Numerical simulation via DeepQuantum All numerical simulations were conducted using DeepQuantum, an open-source software platform built on PyTorch²⁴. The framework's robust support for photonic quantum computing, combined with its native integration with classical deep learning workflows, makes it particularly well-suited for our study. For the specific task of fitting the target unitary matrix, we utilized its `QumodeCircuit` class to construct, simulate the TDM circuit, and optimize its parameters. The optimization algorithm proceeds through the following steps:

1. **Circuit Construction:** A TDM circuit is constructed based on the hybrid architecture shown in Fig. 3c, configured with 1–4 physical modes.
2. **Circuit Unrolling:** For a given number of timesteps, the iterative TDM circuit is computationally "unrolled" into its equivalent global spatial circuit using the `global_circuit` method. The modes in this expanded circuit comprise both the physical modes and the auxiliary modes representing the delay loops.
3. **Submatrix Extraction:** The unitary matrix of the global circuit is then computed. From this, we extract the submatrix U_{sub} that describes the transfor-

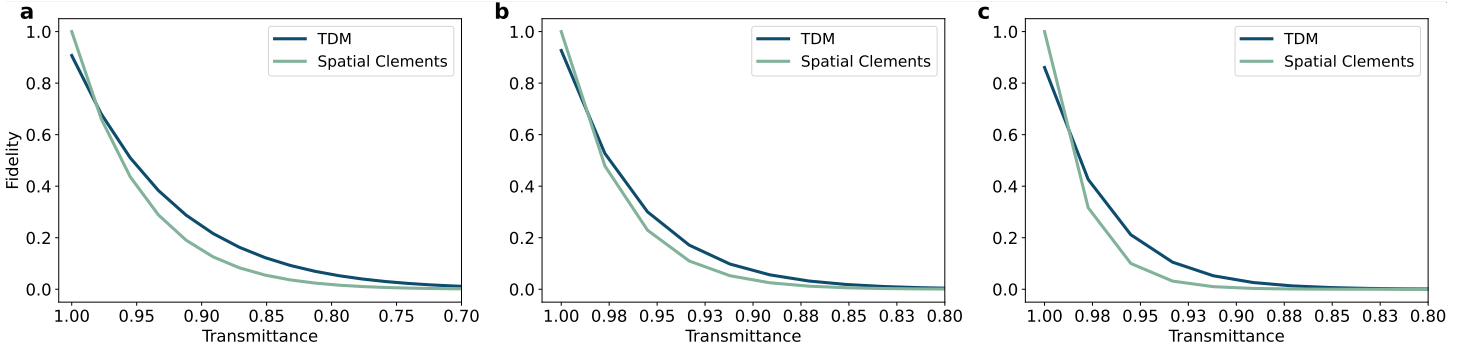


Figure 5 | The robustness of fidelity against optical loss for TDM and spatial Clements architectures. a–c, Fidelity of fitted unitary matrices as a function of transmittance for matrix sizes of 36×36 (a), 64×64 (b), and 100×100 (c). The decrease in transmittance corresponds to increasing optical loss. In all panels, the dark blue curves represent the results of the TDM architecture, simulated with 6 modes in a, 8 modes in b, and 10 modes in c. The green curves represent the results of the spatial Clements architecture, simulated with 36 modes in a, 64 modes in b, and 100 modes in c.

mation exclusively on the physical modes by discarding the rows and columns associated with the delay-loop modes.

4. Optimal Window Identification: To find the optimal embedding of the target $N \times N$ unitary U , we employ a sliding window approach. Considering the time-translation symmetry in TDM circuits, all non-redundant $N \times N$ contiguous sub-blocks within U_{sub} are scanned. Then, the one yielding the highest initial fidelity with U is selected as the candidate matrix U_{fit} for optimization.

5. Optimization and Parameter Search: The circuit parameters are optimized to maximize fidelity. We utilize the Adam optimizer with a learning rate of 0.08. To thoroughly explore the parameter space and avoid local minima, an extensive search is performed for each timestep configuration. We perform 8 rounds of batched optimization with a batch size of 80 over 350 epochs, where each batch concurrently evaluates a distinct set of randomly initialized circuit parameters through parallelized gradient descent. The loss function for each round is defined to maximize the fidelity:

$$\text{Loss} = - \sum_{i=1}^{80} F_i, \quad (4)$$

$$F_i = \frac{|\text{Tr}(U^\dagger \cdot U_{\text{fit}}^i)|}{N}. \quad (5)$$

Here Tr denotes the trace of a matrix. U^\dagger represents the Hermitian conjugate of the target $N \times N$ unitary matrix U , and U_{fit}^i represents the best fitted matrix. The operation $|\cdot|$ takes the magnitude of the complex value.

After completing all 8-round optimizations, the highest fidelity observed across all 640 optimization instances (8 rounds \times 80 configurations per round) is recorded as the final result for that specific timestep.

The GBS simulation algorithm consists of the following steps:

1. Circuit Construction: Construct the GBS circuit using the GBS_Graph class by setting the input adjacency matrix, photon-number cutoff, and total mean photon number.

2. Full Probability Distribution Calculation: Evolve the circuit and obtain the probability distribution over all possible output Fock states $s = (s_1, s_2, \dots, s_m)$ within the given photon-number cutoff.

3. Post-Selection for Fixed-Size Subgraphs: Filter the full probability distribution to isolate outcomes corresponding to subgraphs of the specified size k by selecting all states with $s_i \in \{0, 1\}$ that have a total photon number of exactly k . The result is the desired probability distribution over all k -node subgraphs.

REFERENCES

- Gupta, A., Dwivedi, D. N. & Shah, J. Overview of money laundering. In *Artificial Intelligence Applications in Banking and Financial Services: Anti Money Laundering and Compliance*, 1–11 (Springer, 2023).
- Goecks, L. S., Korzenowski, A. L., Gonçalves Terra Neto, P., de Souza, D. L. & Mareth, T. Anti-money laundering and financial fraud detection: A systematic lit-

- erature review. *Intelligent Systems in Accounting, Finance and Management* **29**, 71–85 (2022).
- Ma, X. *et al.* A comprehensive survey on graph anomaly detection with deep learning. *IEEE transactions on knowledge and data engineering* **35**, 12012–12038 (2021).
- Li, J. *et al.* A comprehensive review of community detection in graphs. *Neurocomputing* **600**, 128169 (2024).
- Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510 (2019).
- Zhong, H.-S. *et al.* Quantum computational advantage using photons. *Science* **370**, 1460–1463 (2020).
- Wu, Y. *et al.* Strong quantum computational advantage using a superconducting quantum processor. *Physical review letters* **127**, 180501 (2021).
- Zhong, H.-S. *et al.* Phase-programmable gaussian boson sampling using stimulated squeezed light. *Physical review letters* **127**, 180502 (2021).
- Madsen, L. S. *et al.* Quantum computational advantage with a programmable photonic processor. *Nature* **606**, 75–81 (2022).
- Deng, Y.-H. *et al.* Gaussian boson sampling with pseudo-photon-number-resolving detectors and quantum computational advantage. *Physical review letters* **131**, 150601 (2023).
- Liu, H.-L. *et al.* Robust quantum computational advantage with programmable 3050-photon gaussian boson sampling. *arXiv preprint arXiv:2508.09092* (2025).
- Aaronson, S. & Arkhipov, A. The computational complexity of linear optics. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, 333–342 (2011).
- Hamilton, C. S. *et al.* Gaussian boson sampling. *Physical review letters* **119**, 170501 (2017).
- Brädler, K., Dallaire-Demers, P.-L., Rebentrost, P., Su, D. & Weedbrook, C. Gaussian boson sampling for perfect matchings of arbitrary graphs. *Phys. Rev. A* **98**, 032310 (2018). URL <https://link.aps.org/doi/10.1103/PhysRevA.98.032310>.
- Bromley, T. R. *et al.* Applications of near-term photonic quantum computers: software and algorithms. *Quantum Science and Technology* **5**, 034010 (2020).
- Arrazola, J. M. & Bromley, T. R. Using gaussian boson sampling to find dense subgraphs. *Phys. Rev. Lett.* **121**, 030503 (2018). URL <https://link.aps.org/doi/10.1103/PhysRevLett.121.030503>.
- Sempere-Llagostera, S., Patel, R., Walmsley, I. & Kolthammer, W. Experimentally finding dense subgraphs using a time-bin encoded gaussian boson sampling device. *Physical Review X* **12**, 031045 (2022).
- Deng, Y.-H. *et al.* Solving graph problems using gaussian boson sampling. *Physical Review Letters* **130**, 190601 (2023).
- Banchi, L., Fingerhuth, M., Babej, T., Ing, C. & Arrazola, J. M. Molecular docking with gaussian boson sampling. *Science advances* **6**, eaax1950 (2020).
- Yu, S. *et al.* A universal programmable gaussian boson sampler for drug discovery. *Nature Computational Science* **3**, 839–848 (2023).
- Oh, C., Fefferman, B., Jiang, L. & Quesada, N. Quantum-inspired classical algorithm for graph problems by gaussian boson sampling. *PRX quantum* **5**, 020341 (2024).
- Oh, C., Liu, M., Alexeev, Y., Fefferman, B. & Jiang, L. Classical algorithm for simulating experimental gaussian boson sampling. *Nature Physics* **20**, 1461–1468 (2024).
- Zhang, Y. *et al.* Efficient classical sampling from gaussian boson sampling distributions on unweighted graphs. *Nature Communications* **16**, 9335 (2025).
- He, J.-J., Hu, K.-M. & Zhu, Y.-Z. Deepquantum: A pytorch-based software platform for quantum machine learning and photonic quantum computing. <https://github.com/TuringQ/deepquantum> (2025).
- Girvan, M. & Newman, M. E. Community structure in social and biological networks. *Proceedings of the national academy of sciences* **99**, 7821–7826 (2002).

26. Traag, V. A., Waltman, L. & Van Eck, N. J. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports* **9**, 1–12 (2019).
27. Zhang, S. *et al.* Hidden: hierarchical dense subgraph detection with application to financial fraud detection. In *Proceedings of the 2017 SIAM international conference on data mining*, 570–578 (SIAM, 2017).
28. Charikar, M. Greedy approximation algorithms for finding dense components in a graph. In *International workshop on approximation algorithms for combinatorial optimization*, 84–95 (Springer, 2000).
29. Goldberg, A. V. Finding a maximum density subgraph (1984).
30. Shi, F.-B., Sun, X.-Q., Shen, H.-W. & Cheng, X.-Q. Detect colluded stock manipulation via clique in trading network. *Physica A: Statistical Mechanics and its Applications* **513**, 565–571 (2019).
31. Bron, C. & Kerbosch, J. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM* **16**, 575–577 (1973).
32. Pardalos, P. M. & Rodgers, G. P. A branch and bound algorithm for the maximum clique problem. *Computers & operations research* **19**, 363–375 (1992).
33. Schmidt, J., Pasadakis, D., Sathe, M. & Schenk, O. Gamnet: a graph based framework for the detection of money laundering. In *2024 11th IEEE Swiss Conference on Data Science (SDS)*, 241–245 (IEEE, 2024).
34. Ullmann, J. R. An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)* **23**, 31–42 (1976).
35. Cordella, L. P., Foggia, P., Sansone, C., Vento, M. *et al.* An improved algorithm for matching large graphs. In *3rd IAPR-TC15 workshop on graph-based representations in pattern recognition*, 149–159 (2001).
36. Sempere-Llagostera, S., Patel, R. B., Walmsley, I. A. & Kolthammer, W. S. Experimentally finding dense subgraphs using a time-bin encoded gaussian boson sampling device. *Phys. Rev. X* **12**, 031045 (2022). URL <https://link.aps.org/doi/10.1103/PhysRevX.12.031045>.
37. Brádler, K., Friedland, S., Izaac, J., Killoran, N. & Su, D. Graph isomorphism and gaussian boson sampling. *Special Matrices* **9**, 166–196 (2021).
38. Reck, M., Zeilinger, A., Bernstein, H. J. & Bertani, P. Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.* **73**, 58–61 (1994). URL <https://link.aps.org/doi/10.1103/PhysRevLett.73.58>.
39. Clements, W. R., Humphreys, P. C., Metcalf, B. J., Kolthammer, W. S. & Walmsley, I. A. Optimal design for universal multiport interferometers. *Optica* **3**, 1460–1465 (2016). URL <https://opg.optica.org/optica/abstract.cfm?URI=optica-3-12-1460>.
40. Motes, K. R., Gilchrist, A., Dowling, J. P. & Rohde, P. P. Scalable boson sampling with time-bin encoding using a loop-based architecture. *Phys. Rev. Lett.* **113**, 120501 (2014). URL <https://link.aps.org/doi/10.1103/PhysRevLett.113.120501>.
41. He, Y. *et al.* Time-bin-encoded boson sampling with a single-photon device. *Phys. Rev. Lett.* **118**, 190501 (2017). URL <https://link.aps.org/doi/10.1103/PhysRevLett.118.190501>.
42. Monika, M. *et al.* Quantum state processing through controllable synthetic temporal photonic lattices. *Nature Photonics* **19**, 95–100 (2025).
43. Suzumura, T. & Kanazashi, H. Anti-money laundering datasets: Inpluslab anti-money laundering datadatasets. *Anti-Money Laundering Datasets: InPlusLab anti-money laundering datadatasets* (2021).

MISCELLANEA

Acknowledgements This work was supported by the Science and Technology Commission of Shanghai Municipality under Grant No. 24LZ1401500.

Declaration of Interest Statement

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for this journal and was not involved in the editorial review or the decision to publish this article.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

--