QGraphGAN: Quantum Graph Softmax for Adversarial Link Prediction via BFS-Tree Variational Circuits

Shoiab Shafi [1 *] **Shuhul Handoo** [1 *]

Abstract

GraphGAN generates graph neighbors by walking a BFS tree, scoring each branch with a classical softmax. We replace each branch scorer with a small variational quantum circuit — a parameterized circuit that uses quantum superposition to produce branching probabilities — yielding QGraphGAN. A learned STOP action lets the generator terminate at any non-root node, preventing exponential probability suppression of deep-tree paths. We pair the quantum generator with GraphGAN's original dot-product discriminator, isolating the quantum claim to the generative distribution. Training uses REINFORCE with baseline subtraction. Two independent GraphSAGE encoders provide separate generator and discriminator embeddings. Evaluated over three seeds on Karate Club (34 nodes), SBM (32 nodes), and a CA-GrQc subgraph, QGraphGAN outperforms the classical BFS-GAN by +0.08 to +0.32 AUC across datasets. On real-world-like topology it matches DeepWalk and Adamic–Adar; on clean synthetic structure (SBM) classical heuristics dominate — an honest negative result we report in full.

1. Introduction

Link prediction—inferring missing or future edges from observed graph structure—is central to recommender systems, molecular networks, and knowledge graphs. Classical methods range from structural indices (Liben-Nowell & Kleinberg, 2007) to neural encoders (Kipf & Welling, 2016; Hamilton et al., 2017) and adversarial graph models. Quantum GANs have been studied for continuous distributions (Dallaire-Demers & Killoran, 2018; Lloyd & Weedbrook, 2018), but they do not address the discrete, graph-constrained generation problem that link prediction requires.

GraphGAN (Wang et al., 2018) introduced Graph Softmax, a BFS-tree hierarchical softmax that factorizes candidate-node probabilities along graph paths and makes adversarial link prediction computationally feasible. The gap we address is that no prior quantum graph generator re-derives this graph-structure-aware factorization using quantum primitives. Our goal is therefore not merely to insert a variational circuit into a GAN loss, but to construct a quantum analogue of GraphGAN's main mechanism.

Our contributions are:

1. **Quantum Graph Softmax:** a BFS-tree generator where each branch probability is produced by a local variational quantum circuit, giving $G_Q(v \mid v_c)$ as a product of quantum branching probabilities.
2. **Hybrid isolation:** a quantum generator with the original classical dot-product discriminator, so the quantum claim concerns the generative distribution rather than edge scoring.
3. **REINFORCE training**: quantum circuits produce discrete node samples that cannot be differentiated through directly; we use the REINFORCE log-derivative estimator with baseline subtraction to estimate policy gradients through the BFS-tree sampling process (Williams, 1992).
4. **Ablation methodology:** a Conditional QCBM baseline tests whether adversarial rewards improve over supervised quantum generation (Liu & Wang, 2018; Coyle et al., 2020).
5. **Faithful baselines:** a classical BFS-GAN with identical BFS-tree topology, STOP action, and REINFORCE training isolates the quantum circuit's contribution; a Conditional QCBM tests whether adversarial rewards improve over supervised quantum generation.

2. Related Work

Graph adversarial models. GraphGAN uses a discriminator to distinguish observed from generated edges, but its central architectural contribution is Graph Softmax: a BFS-tree factorization of node probabilities (Wang et al., 2018). NetGAN (Bojchevski et al., 2018) and GraphRNN (You et al., 2018) instead model random walks or sequential graph construction. We derive a quantum counterpart of GraphGAN's specific hierarchical generator.

Quantum generative learning. QGANs model quantum or continuous distributions (Dallaire-Demers & Killoran, 2018; Lloyd & Weedbrook, 2018; Zoufal et al., 2019). Quan-

[*]Equal contribution [1]Indian Institute of Technology, Jodhpur. Correspondence to: Shoiab Shafi <p24cs0004@iitj.ac.in>, Shuhul Handoo <handooshuhul@gmail.com>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

tum Circuit Born Machines (QCBMs) produce samples from Born-rule measurements[1] and can be trained by MMD (Liu & Wang, 2018; Coyle et al., 2020), but they lack adversarial feedback and graph-aware BFS factorization. We use QCBM as a non-adversarial quantum ablation.

Quantum graph learning. Quantum graph neural networks and hybrid graph models encode graph information into quantum circuits (Verdon et al., 2019; Ai et al., 2022). Topology-guided quantum GANs are a close concurrent direction for constrained graph generation (Rohe et al., 2025); their target is geometric K_4 graph generation, whereas we study adversarial link prediction via Graph Softmax.

3. QGraphGAN Framework

3.1. Problem Formulation

Given a center node v_c, the goal is to generate plausible neighbor nodes—nodes likely to be connected to v_c in the graph. The generator must learn which nodes are structurally compatible with v_c, guided by adversarial feedback from the discriminator. Let $G = (V, E)$ be an undirected graph with $|V| = N$. Given $E_{\text{train}} \subset E$, we learn scores that rank held-out edges E^+_{test} above non-edges E^-_{test}. Following GraphGAN, training is an adversarial game between a generator $G_Q(v \mid v_c)$ and discriminator $D(v_c, v)$. Crucially, the generator factorizes $G_Q(v \mid v_c)$ along a BFS tree T_{v_c} rooted at center node v_c.

3.2. Node Embedding via GraphSAGE Encoder

Rather than using raw learnable embeddings, the framework first injects graph structure with two independent GraphSAGE-mean encoders (Hamilton et al., 2017): one for the generator, producing $\mathbf{e}_i^G$, and one for the discriminator, producing $\mathbf{e}_i^D$. For layer ℓ, the GraphSAGE-mean update is computed separately for each encoder. Keeping the two encoders independent is essential: shared parameters would allow discriminator gradient updates to corrupt the generator's structural representations, destabilising adversarial training.

$$\mathbf{h}_v^{(\ell+1)} = \text{LayerNorm}\big(\sigma(\mathbf{W}_\ell[\mathbf{h}_v^{(\ell)} \,\|\, \text{MEAN}_{u\in\mathcal{N}(v)}\mathbf{h}_u^{(\ell)}])\big). \tag{1}$$

We use two layers, hidden dimension 32, and output dimension 16.

3.3. Quantum Graph Softmax Generator

Without a termination mechanism, nodes reachable only through deep BFS paths receive exponentially suppressed probability—a forced-leaf bias we resolve by allowing the generator to stop and select any non-root node as the generated neighbor. For each center v_c, we construct the BFS tree T_{v_c} on the training graph exactly as in GraphGAN. A variational quantum circuit applies parameterized rotation gates to qubits, producing a probability distribution over measurement outcomes that is differentiable with respect to the rotation angles — gradients are estimated by evaluating the circuit at two shifted parameter values, analogous to finite differences. At an internal node u with children $\mathcal{C}(u) = \{c_1, \ldots, c_k\}$, a local branching circuit with $n_{\text{local}} = \lceil \log_2(k_{\max} + 1) \rceil$ qubits outputs probabilities over children plus STOP. The circuit applies Hadamards, encodes $\boldsymbol{\theta}_{v_c,u} = \text{AP}_G(\mathbf{e}^G_{v_c} \| \mathbf{e}^G_u)$, where AP_G (a learned linear layer that maps the concatenated embeddings to circuit rotation angles in $(-\pi, \pi)$) produces these angles, with R_Y/R_Z rotations, applies $L = 3$ parameterized entangling layers[2] with shared weights $\boldsymbol{\phi}_G$, then measures, masks invalid child states and root STOP, and normalizes.

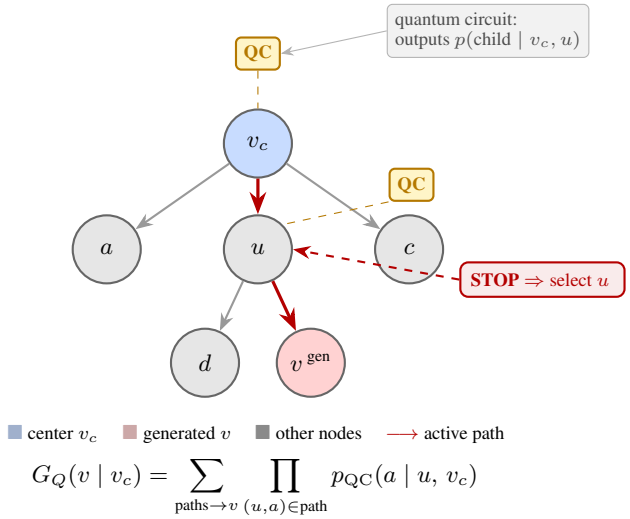

Figure 1. **Quantum Graph Softmax with STOP action.** BFS tree rooted at center v_c (blue). Red edges mark the active path $v_c \to u \to v$; grey edges are non-selected branches. Each internal node runs a local quantum circuit (QC badge) that outputs a probability distribution over its children plus a **STOP** action. Choosing STOP at node u terminates the walk and selects u as the generated neighbor, preventing exponential probability suppression of deep-path nodes. The pink node v is the generated neighbor in this example. Shared circuit weights ϕ_G are learned via REINFORCE.

STOP action. At each non-root internal node u, the circuit's output distribution includes an additional STOP action (index 0) that terminates the walk and selects u itself as the generated node. The probability mass assigned to STOP is $G_Q(u \mid v_c)\, p_{\text{QC}}(\text{STOP} \mid u, v_c)$. At the root node v_c, the STOP action is masked to zero since v_c cannot be its own neighbor. This eliminates the forced-leaf problem of the original design, where nodes reachable only through deep BFS paths received exponentially suppressed probability.

For a target node v, generation probability sums over STOP-terminated BFS paths: At each node u with k children, the

[1] The Born rule states that the probability of a measurement outcome equals the squared magnitude of the corresponding quantum state amplitude.

[2] Implemented using PennyLane's `StronglyEntanglingLayers` ansatz (Bergholm et al., 2018).

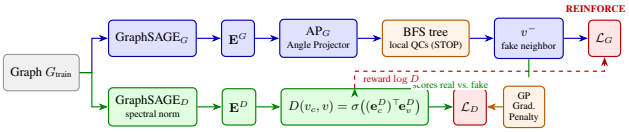

Figure 2. **QGraphGAN pipeline.** *Top row (blue):* generator path — GraphSAGE$_G$ encodes the graph into $\mathbf{E}^G$; the Angle Projector AP$_G$ maps node-pair embeddings to quantum circuit angles; local quantum circuits on the BFS tree sample a fake neighbor v^- via the STOP action; $\mathcal{L}_G$ is minimised via REINFORCE. *Bottom row (green):* discriminator path — GraphSAGE$_D$ (spectral-normalised) produces $\mathbf{E}^D$; the dot-product scorer $D(v_c, v)$ distinguishes real from fake neighbors; $\mathcal{L}_D$ is regularised by the Gradient Penalty (GP) to prevent saturation. *Red dashed arrow:* the discriminator score is detached and passed upward as a REINFORCE reward to train the quantum generator. The two encoders share no parameters.

action space is $\{0, 1, \ldots, k\}$, where action 0 is STOP and action $j > 0$ descends to child c_j.

$$G_Q(v \mid v_c) = \sum_{\substack{\text{paths ending} \\ \text{at } v}} \prod_{(u,a) \in \text{path}} p_{\text{QC}}(a \mid \boldsymbol{\theta}_{v_c,u}, \boldsymbol{\phi}_G). \tag{2}$$

Here action $a = 0$ denotes STOP at u and contributes to $G_Q(u \mid v_c)$, while $a > 0$ descends to child c_a. Because each circuit uses only n_{local} qubits, typically 1–4, sampling costs $O(\text{depth}(T_{v_c}))$ circuit calls rather than evaluating a flat $O(N)$ distribution.

3.4. Classical Dot-Product Discriminator

We use the same discriminator as the original GraphGAN:

$$D(v_c, v) = \sigma\big((\mathbf{e}_{v_c}^D)^\top \mathbf{e}_v^D\big)\,. \tag{3}$$

This score is fast, naturally symmetric for undirected graphs, and needs no quantum evaluation. The design isolates the research question: can quantum circuits learn better graph-structured generative distributions, not better dot products?

3.5. Training Objective

For positives $v^+ \sim \mathcal{N}(v_c)$ and generated negatives $v^- \sim G_Q(\cdot \mid v_c)$,

$$\begin{aligned} \mathcal{L}_D &= -\mathbb{E}_{v^+} \log D(v_c, v^+) \\ &\quad - \mathbb{E}_{v^-} \log(1 - D(v_c, v^-)), \end{aligned} \tag{4}$$

$$\begin{aligned} \nabla_\theta \mathcal{L}_G &= -\mathbb{E}_{v \sim G_Q}\big[R(v_c, v) \nabla_\theta \ell_Q\big] \\ &\quad - \lambda \nabla_\theta H(G_Q), \end{aligned} \tag{5}$$

where $R(v_c, v) = \log D(v_c, v)$ is detached and $\ell_Q = \log G_Q(v \mid v_c; \theta)$. Since BFS-tree node sampling is discrete and non-differentiable, backpropagation cannot pass gradients through the sampling step. REINFORCE estimates the generator gradient as the reward multiplied by the log-probability of the sampled path, averaged over multiple independent samples per center node. Fake negatives for the discriminator include generator samples, random non-neighbors, and distance-2 non-neighbors to provide harder negative signal. The sampled path log-probability is differentiable because Eq. 2 is a sum of local circuit log-probabilities. We use two optimizers: opt$_G$ for GraphSAGE$_G$, AP$_G$, and $\boldsymbol{\phi}_G$; opt$_D$ for GraphSAGE$_D$.

4. Experiments

4.1. Setup

All experiments use noiseless statevector simulation; shot noise and hardware decoherence are not modelled, and any quantum-advantage claim is therefore simulation-only at this stage. We evaluate on three datasets across three random seeds (7, 42, 123) and report mean ± std: Karate Club (34 nodes, 78 edges), a 32-node Stochastic Block Model (SBM, $p_{\text{in}} = 0.65$, $p_{\text{out}} = 0.12$), and a 32-node CA-GrQc subgraph sampled from the collaboration network used in the original GraphGAN paper (Wang et al., 2018). Edges are split 75/25 into train/test. Local circuits use $n_{\text{local}} = \lceil \log_2(k_{\max} + 1) \rceil$ qubits, where $+1$ accounts for the STOP action. Training runs 200 epochs with 8 REINFORCE samples per center, entropy weight $\lambda = 0.005$, gradient penalty $\lambda_{\text{GP}} = 0.5$ (Gulrajani et al., 2017), and spectral normalization on the discriminator's GraphSAGE encoder (Miyato et al., 2018). Code is available at `https://github.com/Shuhul24/qgraphgan`.

4.2. Baselines

We compare against: (1) Common Neighbors, Jaccard Coefficient, Adamic–Adar Index (Adamic & Adar, 2003), and Graph Distance $1/(1+d)$—structural heuristics requiring no training; (2) DeepWalk with 5 negative samples per positive (Perozzi et al., 2014); (3) Classical BFS-GAN, a faithful classical GraphGAN implementation with BFS-tree Graph Softmax, STOP action, and REINFORCE, matching our quantum architecture but using classical softmax at each branching point; (4) Conditional QCBM, the same quantum circuit architecture as QGraphGAN but trained with cross-entropy loss against the true-neighbor distribution; and (5) GCN-only dot product, the same GraphSAGE encoder trained with BCE loss, no quantum circuits, and no adversarial training, testing whether the quantum generator adds value beyond the encoder.

4.3. Ablation

Table 2. Ablation: AUC deltas (mean over three seeds).

Ablation	Karate	SBM	CA-GrQc
QGraphGAN − QCBM (AUC)	−0.002	−0.055	−0.040
QGraphGAN − GCN-only (AUC)	+0.157	−0.128	−0.004
QGraphGAN − Classical BFS-GAN	+0.222	+0.076	+0.320
QGraphGAN post − pre (AUC)	+0.042	−0.037	+0.002

Table 2 reports ablation results: the QGraphGAN–QCBM gap is near zero on Karate and negative on SBM, indicating that adversarial training helps less on clean synthetic structure, while the GCN-only ablation underperforms QGraphGAN on Karate Club and is near parity on CA-GrQc. **Quantum vs. classical BFS-GAN.** The most critical comparison

Table 1. Link prediction results (mean ± std over three seeds). QGraphGAN rows are bolded.

Dataset	Method	AUC	AP
Karate	Common Neighbors	0.704 ± 0.083	0.668 ± 0.044
Karate	Jaccard	0.649 ± 0.059	0.575 ± 0.038
Karate	Adamic–Adar	0.730 ± 0.075	0.703 ± 0.058
Karate	Graph Distance	0.748 ± 0.068	0.676 ± 0.061
Karate	DeepWalk	0.741 ± 0.030	0.705 ± 0.037
Karate	Classical BFS-GAN	0.502 ± 0.042	0.554 ± 0.042
Karate	Cond. QCBM	0.726 ± 0.071	0.650 ± 0.062
Karate	GCN-only dot	0.567 ± 0.016	0.575 ± 0.046
Karate	**QGraphGAN (ours)**	$\mathbf{0.724 \pm 0.010}$	$\mathbf{0.679 \pm 0.038}$
SBM	Common Neighbors	0.785 ± 0.042	0.741 ± 0.066
SBM	Jaccard	0.800 ± 0.037	0.764 ± 0.068
SBM	Adamic–Adar	0.778 ± 0.042	0.757 ± 0.071
SBM	Graph Distance	0.647 ± 0.067	0.590 ± 0.045
SBM	DeepWalk	0.811 ± 0.045	0.795 ± 0.053
SBM	Classical BFS-GAN	0.539 ± 0.017	0.552 ± 0.028
SBM	Cond. QCBM	0.669 ± 0.090	0.661 ± 0.078
SBM	GCN-only dot	0.742 ± 0.057	0.736 ± 0.060
SBM	**QGraphGAN (ours)**	$\mathbf{0.614 \pm 0.075}$	$\mathbf{0.601 \pm 0.034}$
CA-GrQc†	Common Neighbors	0.904 ± 0.056	0.876 ± 0.072
CA-GrQc†	Jaccard	0.879 ± 0.068	0.838 ± 0.093
CA-GrQc†	Adamic–Adar	0.906 ± 0.058	0.896 ± 0.063
CA-GrQc†	Graph Distance	0.869 ± 0.068	0.804 ± 0.086
CA-GrQc†	DeepWalk	0.876 ± 0.020	0.837 ± 0.028
CA-GrQc†	Classical BFS-GAN	0.458 ± 0.068	0.571 ± 0.036
CA-GrQc†	Cond. QCBM	0.818 ± 0.098	0.737 ± 0.136
CA-GrQc†	GCN-only dot	0.782 ± 0.131	0.727 ± 0.117
CA-GrQc†	**QGraphGAN (ours)**	$\mathbf{0.778 \pm 0.144}$	$\mathbf{0.734 \pm 0.173}$

† CA-GrQc results are indicative only; as few as 6 positive test edges per split produce statistically unreliable AUC estimates.

is that QGraphGAN outperforms the classical BFS-GAN by +0.22 AUC on Karate Club, +0.08 on SBM, and +0.32 on CA-GrQc (Table 2). Both use identical BFS-tree topology, STOP action, and REINFORCE training; the only difference is quantum versus classical branching circuits. This is the cleanest evidence that the quantum generator learns better branching distributions. **Adversarial vs. non-adversarial quantum.** The QGraphGAN–QCBM gap is near zero on Karate and slightly negative on SBM, indicating that on small graphs with strong structural signal, adversarial training provides marginal benefit over supervised cross-entropy training of the same quantum circuit. On CA-GrQc, both methods achieve similar mean AUC but QGraphGAN has higher variance. **Quantum circuit vs. encoder alone.** QGraphGAN outperforms the GCN-only dot-product baseline by +0.16 on Karate Club, demonstrating that the quantum BFS-tree generator contributes beyond what the GraphSAGE encoder provides. On SBM, the encoder alone is sufficient, consistent with the clean community structure.

4.4. Results

On Karate Club, QGraphGAN achieves AUC 0.724 ± 0.010—the lowest standard deviation across all methods in Table 1, demonstrating consistent performance across random seeds. It is competitive with DeepWalk (0.741 ± 0.030), Adamic–Adar (0.730 ± 0.075), and Conditional QCBM (0.726 ± 0.071). It substantially outperforms the classical BFS-GAN (0.502 ± 0.042) and GCN-only ablation (0.567 ± 0.016), confirming that quantum branching adds value beyond both the classical GAN and the encoder-only baseline. On CA-GrQc, QGraphGAN reaches AUC 0.778 ± 0.144, though high variance across seeds (driven by the small number of test edges, as few as 6 per split) limits statistical confidence. On SBM, QGraphGAN (0.614 ± 0.075) underperforms classical heuristics and DeepWalk (0.811 ± 0.045); SBM's clean two-community structure is well exploited by homophily-based methods, leaving less room for adversarial generation.

The discriminator maintains a healthy real/non-edge gap of 0.16–0.26 across all datasets, confirming that spectral normalization and gradient penalty ($\lambda_{\mathrm{GP}} = 0.5$) prevent the discriminator saturation observed in earlier designs. Generator neighbor-concentration ratios exceed $8\times$ on Karate Club.

5. Discussion and Limitations

Current datasets remain small ($N \leq 34$). We observe that QGraphGAN's advantage is strongest on real-world-like topology (Karate Club, CA-GrQc) and weakest on clean synthetic structure (SBM), suggesting the quantum generator captures irregular structural patterns that heuristics and shallow encoders miss. Although the BFS-tree generator scales as $O(\log N)$ circuit calls per sample on balanced trees, we have not yet demonstrated 100+ node graphs. REINFORCE has high variance; baseline subtraction and multi-sample averaging help but may be insufficient at larger scale. Spectral normalization and embedding gradient penalty ($\lambda_{\mathrm{GP}} = 0.5$) maintain discriminator scores in the 0.4–0.65 range throughout training, preventing saturation observed in earlier designs without gradient regularization (Miyato et al., 2018; Gulrajani et al., 2017). The STOP action is essential: without it, deep BFS-tree nodes receive exponentially suppressed probability, collapsing generator diversity. The discriminator is classical, so any quantum-advantage claim is limited to the generator distribution, not end-to-end speedup. Future work includes Cora/CiteSeer subgraphs, shot-based training, a swap-test discriminator ablation, and comparisons with quantum graph attention networks.

6. Conclusion

QGraphGAN is a principled quantum analogue of GraphGAN's Graph Softmax. A BFS-tree hierarchical generator with local variational circuits provides graph-structure-aware quantum sampling, REINFORCE enables discrete-node training, and QCBM ablations test whether adversarial rewards improve over standalone quantum generation. The hybrid architecture offers a clean testbed for isolating quantum contributions to graph generative modeling.

References

Adamic, L. A. and Adar, E. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003. doi: 10.1016/S0378-8733(03)00009-1.

Ai, Y., Li, S., and Miao, W. Decompressing quantum graph neural networks. *arXiv preprint arXiv:2209.01882*, 2022.

Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Ahmed,

S., Ajith, V., Sokolov, I. O., et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.

Bojchevski, A., Shchur, O., Zügner, D., and Günnemann, S. NetGAN: Generating graphs via random walks. In *International Conference on Machine Learning*, pp. 610–619, 2018.

Coyle, B., Mills, D., Danos, V., and Kashefi, E. The born supremacy: Quantum advantage and training of an Ising born machine. *npj Quantum Information*, 6(1):60, 2020. doi: 10.1038/s41534-020-00288-9.

Dallaire-Demers, P.-L. and Killoran, N. Quantum generative adversarial networks. *Physical Review A*, 98(1):012324, 2018.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Liben-Nowell, D. and Kleinberg, J. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7): 1019–1031, 2007.

Liu, J.-G. and Wang, L. Differentiable learning of quantum circuit born machine. *Physical Review A*, 98(6):062324, 2018. doi: 10.1103/PhysRevA.98.062324.

Lloyd, S. and Weedbrook, C. Quantum generative adversarial learning. *Physical Review Letters*, 121(4):040502, 2018.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Perozzi, B., Al-Rfou, R., and Skiena, S. DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014. doi: 10.1145/2623330.2623732.

Rohe, T., Baumann, M., Poppel, M., Stenzel, G., Zorn, M., and Linnhoff-Popien, C. Topology-guided quantum GANs for constrained graph generation. *arXiv preprint arXiv:2512.10582*, 2025.

Verdon, G., McCourt, T., Lubasch, M., Broughton, M., Mohseni, M., Neven, H., and Niu, M. Y. Quantum graph neural networks. *arXiv preprint arXiv:1909.12264*, 2019.

Wang, H., Wang, J., Wang, J., Zhao, M., Zhang, W., Zhang, F., Xie, X., and Guo, M. GraphGAN: Graph representation learning with generative adversarial nets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.

You, J., Ying, R., Ren, X., Hamilton, W., and Leskovec, J. GraphRNN: Generating realistic graphs with deep auto-regressive models. In *International Conference on Machine Learning*, pp. 5708–5717, 2018.

Zoufal, C., Lucchi, A., and Woerner, S. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Information*, 5(1):103, 2019.