Integrating Quantum Feature Maps into Graph Neural Networks for Robust Visual Localization

Emma Bennett
Department of Electrical Engineering and Computer Science
University of Missouri
Columbia, MO, USA
enb7x6@umsystem.edu

Ricky D. Massaro
Geospatial Research Laboratory
US Army ERDC
Alexandria, VA, USA

Filiz Bunyak*
Department of Electrical Engineering and Computer Science
University of Missouri
Columbia, MO, USA
bunyak@umsystem.edu

***Abstract*—Robust aerial localization in GNSS-denied environments is a critical challenge for autonomous navigation. This paper introduces a hybrid Quantum-Classical Graph Neural Network (QGNN) framework for visual localization, treating the task as an approximate subgraph matching problem robust to structural perturbations. We propose and evaluate two architectures for integrating quantum feature maps into a classical graph convolution backbone: a hardware-efficient global projection, where a single quantum circuit processes the pooled graph embedding, and a structurally expressive local projection, where quantum circuits are applied to individual node embeddings prior to aggregation. The models are trained via triplet loss on geospatial graphs constructed from the Massachusetts Buildings dataset. We demonstrate that while global methods are computationally tractable on current IBM Quantum hardware, the local quantum-classical hybrid significantly outperforms classical baselines in high-noise regimes. Specifically, the local quantum architecture improves Recall@1 by up to 20 percentage points under severe structural perturbations (e.g., node deletions and additions), suggesting that node-level quantum feature mapping provides improved regularization against topological noise.**

I. Introduction

Reliable spatial localization is a fundamental task for autonomous navigation systems. Although Global Navigation Satellite Systems (GNSS) are widely used, their susceptibility to signal obstruction (e.g., urban canyons) and adversarial interference (e.g., jamming and spoofing) motivates the development of alternative vision-based approaches [1]. Visual localization traditionally relies on pixel-level descriptors or deep-learning features to enable operation in GNSS-denied settings [2], [3], [4]. However, these methods often fail to bridge the modality gap created by seasonal shifts or sensor variances, and struggle in dense urban environments that lack distinctive visual landmarks. To mitigate these challenges, structural abstractions represent scenes geometrically, transforming localization into a topology-matching problem beyond image-level distractors [5], [6], [7], [8], [9]. The trade-off is that upstream detection errors (e.g., missing or merged buildings) directly perturb the resulting graph structure, placing greater pressure on the expressivity of the node and graph features used for retrieval.

*Corresponding author

Because classical heuristics and standard Graph Neural Networks (GNNs) often lack the high-dimensional expressivity required to handle this structural noise, we propose a hybrid Quantum Graph Neural Network (QGNN) framework for aerial localization. Quantum feature maps encode classical graph data into a high-dimensional Hilbert space, which has been theoretically shown capable of capturing complex correlations that are computationally difficult for classical kernels [10], [11]. By mapping structural graph data into this quantum-enhanced feature space, we hypothesize an improvement in subgraph matching robustness against significant structural noise.

We investigate two architectural paradigms for integrating quantum circuits into a GNN pipeline:

1) *Global Projection:* A hardware-efficient approach where the entire graph is pooled classically before being processed by a single quantum circuit.
2) *Local Projection:* A structurally expressive approach where quantum kernels are applied to individual node embeddings prior to pooling.

This paper introduces a novel end-to-end quantum-enhanced processing workflow evaluated on real-world satellite imagery, offering a systematic study of the trade-offs between the computational efficiency of global quantum projections and the increased robustness provided by local quantum feature maps.

II. Related Work

A. Visual and Structural-Based Localization

Traditional UAV localization relies on template matching or deep learning networks (e.g., CNNs, VAEs) to extract features prior to matching [2], [3], [4]. To overcome performance degradation caused by sensor modality gaps, environmental changes, and structural homogeneity in urban areas, recent research has shifted toward structural abstractions [5], [6], [7]. By extracting buildings from images [12] and representing scenes as spatial graphs of semantic objects, localization is framed as an approximate subgraph matching problem, offering inherent robustness to non-structural variations [8], [9].

B. Graph Matching and Learned Representations

Framing aerial localization as subgraph-to-graph matching introduces an NP-hard computational challenge. Classical graph alignment-based methods offer exact structural guarantees but scale exponentially (Ullmann[13], VF2[14], VF3[15]), while heuristic approaches improve speed at the cost of match quality guarantees (e.g., SAGA[16], G-Finder[17]). Feature-space similarity methods embed graphs into fixed-dimensional spaces using kernel-based strategies (shortest-path[18], graphlet[19], WL[20]), improving efficiency but remaining costly over large geospatial search spaces.

More recently, Graph Neural Networks (GNNs) have emerged as a popular tool for learning graph representations and similarity measures. Architectures such as Graph Convolutional Networks[21] (GCNs) and GraphSAGE[22], a convolution-esque GNN variant, iteratively propagate and aggregate neighbor features to produce node and graph embeddings that can be used for matching and retrieval. Architectures like NeuroMatch[23] and SimGNN[24] learn similarities between query and candidate graphs, but lack interpretability and theoretical guarantees of match quality. Other methods such as GOTSIM[25] provide improved interpretability via node alignment, but do not inherently prioritize the spatial and geometric information essential for UAV localization.

C. Quantum Methods

While learned GNN embeddings improve over heuristic methods, they can still degrade under structural noise from scene graph construction, motivating more expressive similarity mechanisms. A common quantum machine learning approach embeds classical inputs into an exponentially large Hilbert space (size 2^n for n qubits), producing quantum-enhanced features theorized to capture complex correlations [10], [11].

In the noisy intermediate scale quantum (NISQ) era, these features are typically generated with parameterized quantum circuits (PQCs), which learn feature mappings through updates analogous to classical neural networks [26]. Gradient estimation, however, differs on quantum computers. On real-hardware it is computed via parameter-shift[27], while statevector simulators can use adjoint differentiation or backpropagation through the simulator's computational graph to compute expectations exactly [28].

Quantum methods have also been explored for learning with graph-structured data. Early work used quantum graph kernels for graph similarity and classification [29], [30], while other approaches used QAOA for tasks such as max-cut and graph coloring [31], [32], [33].

Quantum neural network architectures have also been adapted to graphs. One foundational QGNN approach uses PQCs to represent graph topology by mapping nodes to qubits while edges constrain entangling connectivity[34]. Although this enables global graph classification, it remains limited to small-scale problems on current NISQ hardware. To improve scalability, another approach uses QCNNs, with a fixed-size, weight-sharing circuit architecture trainable on small systems and reusable across larger graph sizes[35]. More recent hybrid frameworks decouple graph size from qubit count by acting locally on nodes. These models use the quantum circuit as a local feature extractor, where each node is associated with a quantum state (often prepared from a classical node embedding) to predict unlabeled node features or refine node-level representations [36].

III. Methodology

Figure 1 summarizes the proposed hybrid classical–quantum subgraph matching workflow for aerial navigation based on spatial neighborhood graphs. Starting from onboard and reference imagery, building centroids are extracted and organized into spatial graphs via Delaunay triangulation with pruning and attribute encoding. These graphs are used to construct anchor, positive, and negative triplets for metric learning. In the *global method*, a classical graph neural network (GNN) first encodes relational structure and produces a graph-level embedding, which is then mapped into a quantum circuit to enhance representation capacity beyond purely classical pooling. In the *local method*, lightweight quantum circuits are applied at the node level prior to pooling, enabling quantum feature transformations to directly influence local structural aggregation. At inference time, embeddings are generated from sliding reference windows and compared against a query embedding using cosine similarity. Localization performance is reported in terms of Recall@1 and Recall@5 measures.

A. Graph Construction

Each satellite image tile is represented as a geospatial graph $G = (V, E)$ built from the building–segmentation masks. Building centroids $\{v_i = (x_i, y_i)\}_{i=1}^{n}$ serve as nodes V. Edges are formed by Delaunay triangulation T and then pruned to retain only local spatial relationships. Nodes are then assigned spatial attributes based on their local, scaled cartesian coordinates (x and y), and an 8-bit directional encoding that discretizes the relative spatial position and orientation of each edge-connected neighbor. Similarly, edges are assigned features based on their normalized relative displacement (Δx, Δy). These continuous values encode the precise spatial offset and orientation between connected neighbors [8]. This process is illustrated in Figure 2.

B. Classical Graph Neural Network Backbone

a) Convolution: We use two Gaussian Mixture Model Convolution (GMMConv) [37] layers with normalization and LeakyReLU activation. Node attributes together with edge gradients $(\Delta x, \Delta y)$ are processed, producing per-node hidden embeddings of size 128 ($[N, 128]$). GMMConv modulates neighboring node contributions with learnable Gaussians dependent on edge attributes. This directional / length-aware weighting is well suited for geospatial building graphs, where spatial layout encoded in edges is informative for localization.

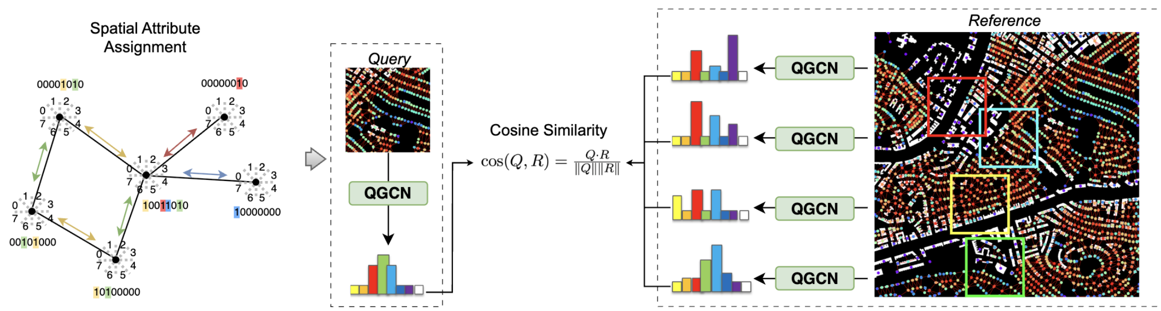

Fig. 1: **Graph-based Aerial Visual Localization Process. (1) Graph processing:** anchor/positive/negative generation, centroid extraction, Delaunay + pruning, and spatial attribute assignment. **(2) Query + reference graph embeddings:** Query graph embeddings and a graph embedding from reference subgraphs are assigned using the trained hybrid QGCN. **(3) Cosine similarity comparison:** Quantum graph embeddings are compared via cosine similarity and the top ranked match returned as the nearest approximate subgraph-graph match.

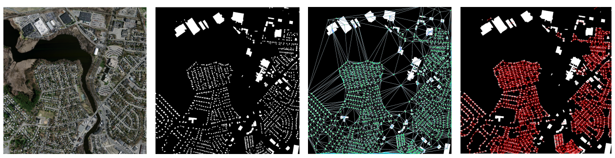

Fig. 2: Preliminary graph construction. From left to right graph construction procedure begins with a satellite or UAV image (a), building masks are then extracted agnostic of this pipeline (b), using building centroids a preliminary graph is constructed via Delaunay triangulation (c), edges are pruned based on the spatial resolution of the image (d).

b) Pooling: To retain multi-scale context, features from multiple convolutional depths are concatenated per node using Jumping Knowledge (JK) [38], yielding a multi-scale representation of shape $[N, 256]$. A learned attention aggregation layer [39] then summarizes all nodes into a fixed-size graph vector$[1, 128]$. Concatenation preserves complementary information from shallow and deeper hops, while attention focuses the readout on the most informative structures, producing a stable, fixed-length graph embedding.

C. Quantum Circuit

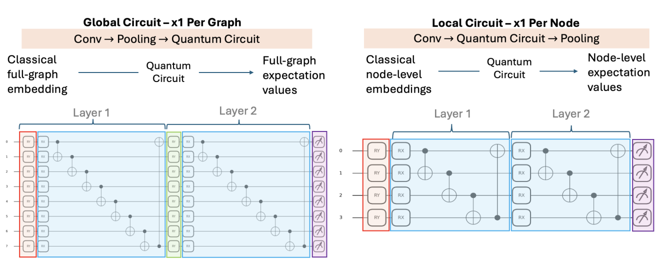

Fig. 3: **Global vs. Local quantum projection.** *Left (Global, 1× per graph):* the classical GNN produces a pooled full-graph embedding which is mapped to a two-layer quantum circuit. Each layer consists of angular $R_y(\theta)$ data embedding, a basic entangling block with trainable $R_x(\phi)$ rotations followed by a CNOT ring, and data re-uploading of the same angles. Z-expectations are measured to yield a quantum vector *Right (Local, 1× per node):* node-level embeddings are each mapped to a smaller 4-qubit circuit of the same structure, node-wise expectations are then pooled by the classical readout. The global design is hardware-efficient (one circuit per graph, shallow depth), while the local design is more expressive but computationally intractably on real hardware.

We investigate two styles of quantum projections. In the *global method*, a single graph-level embedding produced by the classical GNN (after convolution and pooling) is mapped once to an 8-qubit circuit. In the *local method*, a smaller node embedding is mapped per node to a 4-qubit circuit prior to classical pooling. The resulting node-wise quantum vectors are then aggregated to a single graph representation. Both methodologies share the same state-preparation, entangling, and measurement pattern described below. Training is completed on simulators utilizing PennyLane's "backprop" method [28]. These circuits are utilized as a learnable nonlinear embedding that allows high-order interactions between the feature dimensions. In testing, applying the quantum layer at the node level after convolution and before pooling improved robustness to structural perturbations (e.g., missing/merged buildings), likely shaping node features prior to aggregation in a way that regularizes edge-wise variability introduced by GMMConv.

a) Angular Embedding: Given a classical feature vector, we first normalize and scale it to angles $\theta \in [-\pi, \pi]$ using a trainable scale parameter $s \in (0, 1]$, giving us a vector $\theta = [\theta_0, \theta_1, ..., \theta_n]$. Starting from $|0\rangle$, we embed these angles with single-qubit R_y rotations [40].

$$R_y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \tag{1}$$

b) Parameterized Entangling: Each "hidden layer" of the circuit consists of learned single-qubit R_x rotations followed by a ring of CNOTs (derived from the PennyLane BasicEntangler pattern [41]).

$$R_x(\phi) = \begin{pmatrix} \cos\frac{\phi}{2} & -i\sin\frac{\phi}{2} \\ -i\sin\frac{\phi}{2} & \cos\frac{\phi}{2} \end{pmatrix} \tag{2}$$

The CNOT ring couples each qubit to its neighbor, enabling learned, direction-sensitive interactions among features. Additionally, in our global method, we increase expressivity without excessive depth, we re-upload the initial input angles between the entangling layers.

c) Measurement: We read out Pauli-Z expectations on each qubit to obtain a quantum feature vector

$$v_{\text{out}} = [z_0, z_1, \ldots, z_{n_q-1}]^\top, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3}$$

The quantum vector is mapped back to the model dimension and fused with the classical pathway using a learned residual.

D. Training

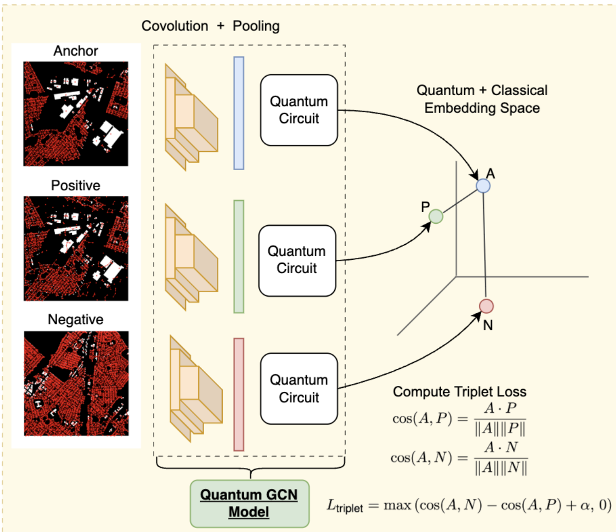

Fig. 4: **Triplet Style Training.** The hybrid QGCN network is trained over anchor, positive (perturbed anchor), and negative (random assigned) graph triplets to encourage embedding clusters to form over spatially similar graphs via Triplet Loss.

Because there are no ground truth labels for the similarity of two graphs in an embedding space, we adopted a self-supervised training approach using a Siamese triplet network [42]. Each triplet (A, P, N) consists of an anchor graph (A), a positive sample (P) generated as a perturbed version of (A), and a negative sample (N) corresponding to an unrelated, randomly selected tile. The loss function utilized is

$$\mathcal{L}_{\text{triplet}} = \max\Big(cos(A, N) - cos(A, P) + \alpha, 0\Big), \tag{4}$$

where the margin α balances separation strength and training stability [43]. This objective directly encodes the retrieval goal that A and P be closer than A and N. Additionally, for each quantum + classical variant (global/local), we train a parameter- and layer-matched classical-only baseline.

To ensure that each batch contains informative positive and negative pairs, we use a class-balanced sampler that first draws P distinct classes (anchor groups) and then samples K instances from each selected class, where each discrete class label denotes an unperturbed graph tile and all of its perturbations derived from the same-tile. This strategy guarantees, for every anchor in the batch, the presence of multiple positives in-class and many negatives outside class, which is critical for stable learning [44].

Batch-hard triplet mining [44] was utilized to further improve the learned expressiveness of the network. Within each batch embeddings are computed and the hardest positive and hardest negative by the lowest similarity among same-class and highest similarity among different classes respectively. This tightens the constraint that the anchor be more similar to every positive than to any negative by optimizing the most difficult (in-batch) cases.

E. Testing

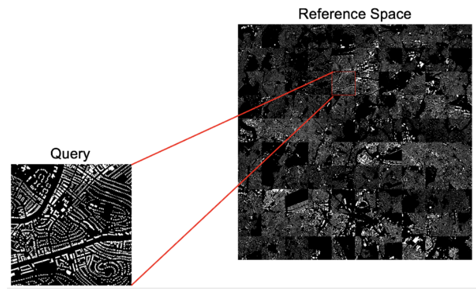

Fig. 5: Experimental testbed.

The experimental testbed was created to simulate real-life UAV localization. A query graph (from our test set of held-out tiles) and its perturbations were compared against a large reference space composed of individual tiles (shown in Figure 5). Candidate locations are generated by sliding a window over the reference space. Extended-maxima peak suppression removes near-duplicate windows. Performance is reported as Recall@1 and Recall@5, the fraction of queries for which the correct location ranks first or appears in the top five, respectively. These metrics are more appropriate than classification accuracy because the task is approximate matching in a large candidate set, not assignment to a fixed label. Each perturbation condition was tested using the held-out queries. These structural perturbations serve as proxies for field effects (occlusion/merging, seasonal change, weather) [8].

F. Implementation Details

Circuit construction and gradient-based training was done in PennyLane using `default.qubit` and `backprop` differentiation[28], which enables end-to-end autograd through the quantum layer. PennyLane's `AngleEmbedding` and `BasicEntanglerLayers` blocks were used to build the quantum circuit[40], [41]. For noisy simulation and exploratory on-device characterization, circuits were mirrored in Qiskit and executed via IBM Quantum Runtime (`Estimator`) on `ibm_brisbane`[45]. Parameters learned in PennyLane were exported to Qiskit to compare embedding behavior under hardware constraints using the same pre/post processing for comparison.

All models were trained for 50 epochs with Adam (weight decay 1×10^{-5}) using a class-balanced $P \times K$ sampler with P=16 classes per batch and K=2 samples (32 graphs/step). Batch-hard triplet mining was used with a triplet margin of α=0.8. The classical-only baseline used a learning rate of 1×10^{-4}. For both hybrid models (global and local), two optimizer parameter groups were used: 1×10^{-4} for non-quantum parameters and 5×10^{-4} for the quantum-layer parameters (i.e., circuit weights).

G. Hardware Analysis (Global, 8-qubit)

Only the *global* method was run on real hardware due to runtime constraints / the computational intractability of the local method. We assessed by comparing generated embeddings across four conditions: Classical (no quantum layer), Quantum–Ideal Simulator, Quantum–Noisy Simulator, and Quantum–Real Hardware. Note: all Recall@K retrieval results in this paper are reported from simulator runs. Hardware experiments were conducted as executability characterization of the global circuit.

All on-device tests were run on `ibm_brisbane`, a 127-qubit Eagle–family processor that uses IBM's heavy-hex layout [46], [47]. The native basis includes single-qubit SX, RZ, and X gates, and the two-qubit ECR gate. Our circuits were transpiled to the basis set which is decomposed into native hardware operations by the compiler [45]. Qiskit's transpiler converts the abstract circuit into the device's native gates and maps our logical qubits onto connected physical qubits. The resulting circuit is shown in Figure 6. Classical processing (graph construction, GNN backbone) was performed on a workstation equipped with an Intel Core i9-10900X processor and 128GB RAM.

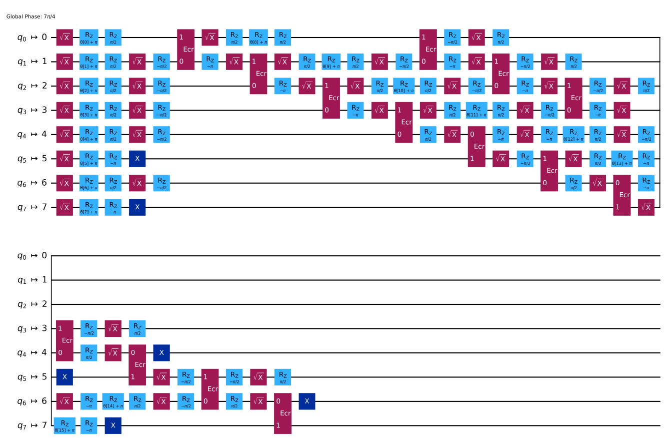

Fig. 6: Transpiled global circuit on `ibm_brisbane`. Colors indicate gate type: R_z (light blue), SX and ECR (magenta), and X (dark blue). R_y-based embedding/data reuploading compiles into R_z + SX sequences, and the CNOT ring compiles into ECR gates.

a) Error mitigation: We used only lightweight options available in Qiskit Runtime: basic measurement-error mitigation (calibrating readout confusion and correcting expectations) [45]. We did not apply heavier techniques such as zero-noise extrapolation. To sanity-check results, each hardware run was compared with an ideal state-vector simulation and a noisy Aer simulation parameterized by device calibrations. We qualitatively compared embedding behavior across conditions (ideal, noisy, hardware) and observed broadly consistent structure with increased variance on-device.

b) Shot counts and jobs: Each expectation value $\langle Z \rangle$ was estimated from S samples ("shots"), so statistical error shrinks as $1/\sqrt{S}$; we therefore used $S{=}1024$ shots for the global method to keep variances small.

IV. Experimental Results

A. Datasets and Testbed

a) Dataset: Experiments were conducted using the Massachusetts Buildings Dataset [48], a standard benchmark for aerial image analysis. The dataset consists of 151 aerial images (1500×1500 pixels) of the Boston area with a spatial resolution of 1 meter, covering a total area of approximately 340 km^2. Ground truth is derived from OpenStreetMap data. This dataset was selected for its high building density and regular urban structures, which provide a challenging environment for graph-based structural matching. A data split of 80% training (used to compose triplet sets) and 20% testing (used to produce the query and reference testbed described below) was used for the following experiments.

TABLE I: Structural perturbations applied to query graphs to simulate segmentation errors and environmental changes.

Condition	Del.	Add.	Shift	Impact
Q_{Base} (Ideal)	-	-	-	Reference
Q_{D10} / Q_{D20}	10%/20%	-	-	Missing bldgs.
Q_{A10} / Q_{A20}	-	10%/20%	-	False positives
Q_{AD10} / Q_{AD20}	10%/20%	10%/20%	-	Combined noise
Q_{SH} (Noise)	-	-	$\mathcal{N}(0, \sigma)$	Boundary jitter

b) Experimental Testbed: To evaluate the robustness of the hybrid QGNN pipeline, we constructed a retrieval testbed designed to simulate GNSS-denied localization under imperfect perception conditions (see Figure 5):

1) *Reference Search Space:* We generated a large-scale reference map composed of a mosaic of 10×10 individual image tiles from the dataset. This creates a contiguous search area where the system must localize the query.
2) *Query Graphs:* We reserved 20% of the dataset (30 tiles) as distinct query tiles for our testing set. For each query, we generated seven perturbed variations (see Table I) to mimic specific failure modes of upstream segmentation algorithms:
   - **Node Deletion (Q_D):** Simulates occlusion (clouds, vegetation) or under-segmentation errors.
   - **Node Addition (Q_A):** Simulates spurious detections (e.g., containers mistaken for buildings) or new construction.
   - **Coordinate Shift (Q_{SH}):** Simulates noisy centroids due to projection errors or irregular segmentation boundaries.

These perturbations are applied to the node set V prior to edge construction (Delaunay triangulation), meaning they propagate changes to the graph topology (edge set E) and attributes, testing the model's topological invariance.

We evaluate on two testbeds: a *full-graph* setting using entire satellite tiles (1500×1500 px; $\sim$1400 nodes/graph; stride 150 px) and a *small-graph* setting using clipped tiles

TABLE II: **Recall@1/5 under controlled perturbations.** Three cases were examined in terms of Classical vs Classical + Quantum (C vs C+Q). Evaluation uses n=30 held-out queries per perturbation. Results indicate (i) Local hybrid is most robust to moderate–heavy perturbations, improving R@1 in 6/8 and R@5 in 7/8 cases on 70-node graphs (typical +10–20 points); (ii) Global hybrid matches classical on light perturbations but degrades under add+del/noise; (iii) For large graphs Local outperforms Global results. Best values are shown in green.

Global ($\sim$1,400 **buildings**)

Perturbations	**R@1**		**R@5**	
	C	**Q**	**C**	**Q**
None	100	100	100	100
10% Del	100	95	100	100
20% Del	100	70	100	95
10% Add	100	95	100	100
20% Add	100	95	100	95
10% Add+Del	95	95	100	100
20% Add+Del	95	50	100	85
Noise Shift	95	60	100	80

Local

Perturbations	$\sim$70 **buildings**				$\sim$1,400 **buildings**			
	R@1		**R@5**		**R@1**		**R@5**	
	C	**Q**	**C**	**Q**	**C**	**Q**	**C**	**Q**
None	90	90	95	95	100	100	100	100
10% Del	80	90	85	100	100	100	100	100
20% Del	55	75	70	75	100	100	100	100
10% Add	80	90	80	90	100	100	100	100
20% Add	75	85	80	90	100	100	100	100
10% Add+Del	75	70	80	85	100	100	100	100
20% Add+Del	55	60	70	70	100	95	100	100
Noise Shift	75	85	85	90	100	100	100	100

(500×500 px; $\sim$70 nodes/graph; stride 50 px). Given the window sizes and strides, each testbed yields $\sim$8,649 windows. Both *global* and *local* hybrids and equivalent classical-only counterparts were first benchmarked on the full-graph setting to establish a common baseline. The small-graph analysis focuses on the *local* hybrid. See Table II for results of all experiments.

c) Findings at satellite scale (1400 nodes): The local hybrid remained at (or extremely near) ceiling across all perturbations (R@1/R@5 $= 100/100$ in the table), while the global hybrid was competitive on mild changes but degraded under the heaviest stressors, 20% add+del and noise shift, with R@1 dropping to 60 and R@5 to 80–85. This confirms that local $>$ global in robustness on large graphs.

d) Findings at onboard-view scale (70 nodes): When the graphs are smaller ($\approx$70 nodes), the local quantum variant outperforms its classical-only counterpart on most perturbations: it improves R@1 in 6/8 settings (and ties once) and improves R@5 in 7/8 settings (ties once). Typical gains are $+10$ to $+20$ points, for example, under 20% deletion R@1 improves $55 \rightarrow 75$ and R@5 $70 \rightarrow 75$; under 10% addition R@1 improves $80 \rightarrow 90$ and R@5 $80 \rightarrow 90$.

V. Conclusion

This paper presented a hybrid quantum–classical graph neural network framework for robust visual localization in aerial imagery by formulating localization as an approximate subgraph matching problem under structural perturbations. By integrating quantum feature maps into a classical graph neural network, we investigated two complementary hybrid architectures: a hardware-efficient global quantum projection, and a structurally expressive local projection.

Experimental results on geospatial building graphs demonstrate promising results for both global and local embeddings. While the global quantum-classical architecture provides computational efficiency and deployability on current hardware, the local quantum–classical architecture provides improved robustness under severe graph perturbations, leading up to a 20 percentage point improvement in Recall@1 compared to classical baselines, indicating enhanced regularization against local topological variations in the subgraph matching task.

These findings suggest that quantum-enhanced graph neural networks offer a promising direction for learning robust graph representations and approximate subgraph matching. The proposed quantum-enhanced workflow is broadly applicable to subgraph matching beyond aerial imaging from molecular discovery or biomedical graphs, to cybersecurity/fraud networks where perturbed graphs are the norm.

Future work will explore scalable circuit designs, adaptive quantum feature maps, and deployment on larger real-world localization scenarios, as well as deeper investigation into the role of quantum embeddings in graph matching and representation learning.

VI. Acknowledgments

The computation for this work was performed on the University of Missouri's Quantum Innovation Center, in partnership with IBM Quantum and facilitated by Research Support Services at the University of Missouri, Columbia MO. DOI: 10.32469/10355/107781. This material is based upon work supported in part by the U.S. Army Corps of Engineers, Engineering Research and Development Center—Information Technology Laboratory (ERDC-ITL) under Contract W912HZ23C0041. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the U.S. Government or agency thereof.

This paper includes text edited for grammar and readability using OpenAI's ChatGPT with all content reviewed for correctness by authors.

References

[1] J. Kinnari, F. Verdoja, and V. Kyrki, "GNSS-denied geolocalization of UAVs by visual matching of onboard camera images with orthophotos," in *IEEE International Conference on Advanced Robotics (ICAR)*, Ljubljana, Slovenia, Dec. 2021, pp. 555–562.

[2] A. Couturier and M. A. Akhloufi, "A review on absolute visual localization for UAV," *Robotics and Autonomous Systems*, vol. 135, p. 103666, Jan. 2021.

[3] ——, "Convolutional neural networks and particle filter for UAV localization," in *SPIE Unmanned Systems Technology XXIII*, vol. 11758, 2021, pp. 108–120.

[4] M. Bianchi and T. D. Barfoot, "UAV localization using autoencoded satellite images," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1761–1768, 2021.

[5] J. Choi and H. Myung, "BRM localization: UAV localization in GNSS-denied environments based on matching of numerical map and UAV images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4537–4544.

[6] M. Schleiss, "Translating aerial images into street-map-like representations for visual self-localization of UAVs," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, pp. 575–580, 2019.

[7] T. Senlet, T. El-Gaaly, and A. Elgammal, "Hierarchical Semantic Hashing: Visual Localization from Buildings on Maps," in *International Conference on Pattern Recognition*, Aug. 2014, pp. 2990–2995.

[8] E. Bennett, J. R. Arredondo, E. Sava, R. D. Massaro, K. Palaniappan, and F. Bunyak, "Bag-of-graph-attributes and sparse graph structures for efficient UAV localization in complex environments," in *Geospatial Informatics XV*, vol. 13461. SPIE, 2025, pp. 107–123.

[9] T. Winterton IV, E. Bennett, B. C. Atmaca, S. Shojaei, K. Palaniappan, and F. Bunyak, "Aerial visual localization through novel applications of Weisfeiler-Lehman graph embeddings," in *Geospatial Informatics XV*, vol. 13461. SPIE, 2025, pp. 96–106.

[10] M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," *Physical Review Letters*, vol. 122, no. 4, p. 040504, 2019. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.122.040504

[11] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.

[12] S. Shojaei, J. R. Arredondo, E. Sava, R. D. Massaro, K. Palaniappan, and F. Bunyak, "Split: A separation-preserving loss for topology-aware building segmentation in remote sensing images," in *Applied Imagery Pattern Recognition Workshop (AIPR)*, LNCS. Springer Nature Switzerland, 2026, pp. 326–344.

[13] J. R. Ullmann, "An Algorithm for Subgraph Isomorphism," *Journal of the ACM*, vol. 23, no. 1, pp. 31–42, Jan. 1976.

[14] L. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub)graph isomorphism algorithm for matching large graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1367–1372, Oct. 2004.

[15] V. Carletti, P. Foggia, A. Greco, M. Vento, and V. Vigilante, "VF3-Light: A lightweight subgraph isomorphism algorithm and its experimental evaluation," *Pattern Recognition Letters*, vol. 125, pp. 591–596, Jul. 2019.

[16] Y. Tian, R. C. McEachin, C. Santos, D. J. States, and J. M. Patel, "SAGA: A subgraph matching tool for biological graphs," *Bioinformatics*, vol. 23, no. 2, pp. 232–239, Jan. 2007.

[17] L. Liu, B. Du, J. Xu, and H. Tong, "G-Finder: Approximate Attributed Subgraph Matching," in *IEEE International Conference on Big Data*, 2019, pp. 513–522.

[18] K. M. Borgwardt and H.-P. Kriegel, "Shortest-path kernels on graphs," in *5th IEEE Int. Conference on Data Mining (ICDM'05)*, 2005, pp. 8–pp.

[19] N. Pržulj, "Biological network comparison using graphlet degree distribution," *Bioinformatics*, vol. 23, no. 2, pp. e177–e183, 2007.

[20] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-Lehman graph kernels." *Journal of Machine Learning Research*, vol. 12, no. 9, 2011.

[21] T. Kipf, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[22] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[23] Z. Lou, J. You, C. Wen, A. Canedo, J. Leskovec *et al.*, "Neural subgraph matching," *arXiv preprint arXiv:2007.03092*, 2020.

[24] Y. Bai, H. Ding, S. Bian, T. Chen, Y. Sun, and W. Wang, "Simgnn: A neural network approach to fast graph similarity computation," in *Proceedings of the 12th ACM international conference on web search and data mining*, 2019, pp. 384–392.

[25] K. D. Doan, S. Manchanda, S. Mahapatra, and C. K. Reddy, "Interpretable graph similarity computation via differentiable optimal alignment of node embeddings," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 665–674.

[26] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum science and technology*, vol. 4, no. 4, p. 043001, 2019.

[27] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Physical Review A*, vol. 99, no. 3, p. 032331, 2019.

[28] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi *et al.*, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv preprint arXiv:1811.04968*, 2018.

[29] M. Schuld, K. Brádler, R. Israel, D. Su, and B. Gupt, "Measuring the similarity of graphs with a Gaussian boson sampler," *Physical Review A*, vol. 101, no. 3, p. 032314, 2020.

[30] L.-P. Henry, S. Thabet, C. Dalyac, and L. Henriet, "Quantum evolution kernel: Machine learning on graphs with programmable arrays of qubits," *Physical Review A*, vol. 104, no. 3, p. 032416, 2021.

[31] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[32] A. Angone, X. Liu, R. Shaydulin, and I. Safro, "Hybrid quantum-classical multilevel approach for maximum cuts on graphs," in *IEEE High Performance Extreme Computing Conference (HPEC)*, 2023, pp. 1–7.

[33] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, "Hybrid quantum-classical algorithms for approximate graph coloring," *Quantum*, vol. 6, p. 678, 2022.

[34] G. Verdon, T. McCourt, E. Luzhnica, V. Singh, S. Leichenauer, and J. Hidary, "Quantum graph neural networks," *arXiv preprint arXiv:1909.12264*, 2019.

[35] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, 2019.

[36] K. Beer, M. Khosla, J. Köhler, T. J. Osborne, and T. Zhao, "Quantum machine learning of graph-structured data," *Physical Review A*, vol. 108, no. 1, p. 012410, 2023.

[37] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," *arXiv preprint arXiv:1611.08402*, 2016. [Online]. Available: http://arxiv.org/abs/1611.08402

[38] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," *arXiv preprint arXiv:1806.03536*, 2018. [Online]. Available: http://arxiv.org/abs/1806.03536

[39] Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli, "Graph matching networks for learning the similarity of graph structured objects," *arXiv preprint arXiv:1904.12787*, 2019. [Online]. Available: http://arxiv.org/abs/1904.12787

[40] R. LaRose and B. Coyle, "Robust data encodings for quantum classifiers," *Physical Review A*, vol. 102, no. 3, p. 032420, 2020.

[41] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum–classical algorithms," *Advanced Quantum Technologies*, vol. 2, no. 12, p. 1900070, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/qute.201900070

[42] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot learning," in *ICML Deep Learning Workshop*, 2015.

[43] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," *arXiv preprint arXiv:1412.6622*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6622

[44] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017. [Online]. Available: http://arxiv.org/abs/1703.07737

[45] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross *et al.*, "Quantum computing with Qiskit," *arXiv preprint arXiv:2405.08810*, 2024.

[46] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, "Topological and subsystem codes on low-degree graphs with flag qubits," *Physical Review X*, vol. 10, no. 1, p. 011022, 2020.

[47] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. Van Den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme *et al.*, "Evidence for the utility of quantum computing before fault tolerance," *Nature*, vol. 618, no. 7965, pp. 500–505, 2023.

[48] V. Mnih, *Machine learning for aerial image labeling*. University of Toronto (Canada), 2013.