Learning-Based Hierarchical Scene Graph Matching for Robot Localization Leveraging Prior Maps

Nimrod Millenium Ndulue[1], Jose Andres Millan-Romera[1], Matteo Giorgi[2], Holger Voos[1], Jose Luis Sanchez-Lopez[1]

***Abstract*—Accurate localization is a fundamental requirement for autonomous robots operating in indoor environments. Scene graphs encode the spatial structure of an environment as a hierarchy of semantic entities and their relationships, and can be constructed both online from robot sensor data and offline from architectural priors such as Building Information Models (BIM). Matching these two complementary representations enables drift correction in SLAM by grounding robot observations against a known structural prior. However, establishing reliable node-to-node correspondences between them remains an open challenge: existing combinatorial methods are prohibitively expensive at scale, and prior learned approaches address only flat graph matching, ignoring the multi-level semantic structure present in both representations. Here we present a learned, end-to-end differentiable pipeline that augments both graphs with semantically motivated edge types encoding intra- and inter-level relationships, explicitly exploiting this hierarchy to enable simultaneous matching from high-level room concepts down to low-level wall surfaces. Trained exclusively on floor plans, the proposed method outperforms the combinatorial baseline in F1 on real LiDAR environments while running an order of magnitude faster, demonstrating viable zero-shot generalization for BIM-assisted robot localization.**

I. Introduction

A long-standing goal in robotics is to deploy spatially intelligent robots capable of accurately reconstructing and understanding their environment. Simultaneous Localization and Mapping (SLAM) provides the foundational framework for this, yet sensor noise and drift inevitably lead to inaccurate maps. Augmenting SLAM with a scene graph representation, a hierarchical graph encoding spatial entities such as rooms and wall surfaces (WS) along with their topological relationships, enriches situational awareness beyond metric reconstruction [1], [2]. In indoor environments, Building Information Models (BIM) are available prior maps that encode the geometric and semantic structure of buildings as structured prior knowledge, and have been leveraged to constrain robot pose estimates against stable architectural features, substantially improving localization accuracy [3]. Applying the same hierarchical representation to a BIM yields a structured architectural prior, formalized as an Architectural Graph (A-graph) [4], whose nodes can be put in one-to-one correspondence with those of the robot's online Situational Graph (S-graph) [2] to correct accumulated drift using the building's known structure.

The key challenge is establishing correspondences between robot observations and prior map entities. Prior work has progressed from dense metric registration against BIM geometry [5] to matching semantic scene features [6], with the state of the art addressing this as a metric-semantic hierarchical graph matching problem via combinatorial search [4]. While learned approaches have tackled graph matching on flat, homogeneous graphs [7], none exploit the hierarchical structure shared by A-graphs and S-graphs, and combinatorial methods do not scale, as detailed in Section II.

We address this with a learned, end-to-end differentiable pipeline that exploits the two-level hierarchy shared by both graphs. Our contributions are:

- Situationally aware augmentation of both A-Graphs and S-graphs with two additionally type of edges, enabling information flow across both hierarchy levels within a single unified graph.
- A learned MLP encoder that maps heterogeneous node features into a homogeneous shared embedding space, enabling type-aware and directly comparable node representations.
- A learned end-to-end differentiable pipeline for hierarchical scene graph matching across high-level (room) and low-level (walls) entities, transferring zero-shot to real LiDAR environments under incomplete prior map observations.

II. Related Work

A. Architectural Prior-Based Robot Localization

Early approaches localize robots by registering raw sensor data against metric maps derived from architectural plans [5], [6], [8], but dense geometric alignment scales poorly and is sensitive to plan-reality deviations. None of these methods exploit the topological relationships between architectural entities, limiting their ability to reason about structural organization.

[1]Authors are with the Automation and Robotics Research Group, Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg. Holger Voos is also associated with the Faculty of Science, Technology and Medicine, University of Luxembourg, Luxembourg. {nimrodmillenium.ndulue, jose.millan, holger.voos, joseluis.sanchezlopez}@uni.lu

[2]Author is with the University of Pisa. m.giorgi13@studenti.unipi.it

This work was partially funded by the Fonds National de la Recherche of Luxembourg (FNR) under the projects 17097684/RoboSAUR and C22/IS/17387634/DEUS.

For the purpose of Open Access, and in fulfillment of the obligations arising from the grant agreement, the authors have applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.

B. Graph Matching

Graph matching seeks to establish node correspondences between two graphs. Classical formulations cast this as a quadratic assignment problem [7], which is NP-hard and thus intractable for large graphs. Learned approaches overcome this by combining graph neural network (GNN)-based embeddings with differentiable Sinkhorn assignment [9]–[11]. With SuperGlue [12] and SG-PGM [13] bringing this paradigm to robotics for keypoint matching and 3D scene graph matching respectively. However, these methods operate on flat graphs of homogeneous nodes, not designed to exploit hierarchical structure between node types of fundamentally different geometric nature. iS-Graphs [4] addresses this by performing hierarchical metric-semantic graph matching between a robot-built scene graph and a BIM-derived architectural prior, yet its combinatorial search grows exponentially with the number of rooms and factorially with the number of walls, making it prohibitively expensive at scale.

III. System Overview

The proposed system takes as input an Architectural Graph (A-graph) [4] derived from BIM and a Situational Graph (S-graph) [2] built online from robot observations, and produces a node-to-node correspondence between them. Since the robot may have only partially explored the environment, the S-graph is typically smaller than the A-graph, and the matching is formulated as a rectangular bipartite assignment. As shown in Fig. 1, the pipeline encodes both graphs, scores cross-graph node similarity through an affinity matrix, and produces a soft assignment matrix that is decoded into a hard one-to-one correspondence at inference. The following subsections detail each component.

A. Hierarchical Graph Representations

Both the A-graph [4] and the S-graph [2] share a common two-level hierarchical structure: *rooms*, representing enclosed spaces, and *wall surfaces* (WS), representing the planar surfaces that delimit them.

a) Node features.: Each node is described by a feature vector $x_i \in \mathbb{R}^7$, see Tab. I.

TABLE I
Node feature vector composition for room and WS nodes.

Dims	Feature	Room	WS
1–2	Node type (one-hot)	$[1, 0]$	$[0, 1]$
3–4	2D centroid (x, y)	(x_r, y_r)	(x_w, y_w)
5–6	2D surface normal (n_x, n_y)	$(0, 0)$	(n_x, n_y)
7	Segment length	-1	ℓ

Dimensions (1–2) encode the node type as a one-hot vector ($\texttt{room}= [1, 0]$, $\texttt{ws}= [0, 1]$); (3–4) encode the 2D centroid coordinates; (5–6) encode the 2D outward-facing surface normal; (7) encodes the segment length, set to -1 for room nodes, which carry no meaningful length attribute. Node features are standardized using the mean and standard deviation computed on the training set and applied consistently across all splits.

b) Feature homogenization.: As shown in Tab. I, the proposed graph is heterogeneous, since room and WS nodes are characterized by different meaningful feature dimensions. To make these features comparable across node types, each raw node feature vector x_i is transformed by a two-layer MLP into an initial node embedding of higher dimension, allowing a more expressive representation to be learned before message passing. Each layer applies an affine transformation followed by a ReLU activation and dropout:

$$\phi^{(l)}(z) = \mathrm{Dropout}\Big(\mathrm{ReLU}\Big(\mathbf{W}^{(l)} z + \mathbf{b}^{(l)}\Big)\Big) \tag{1}$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the learnable parameters of layer l. The initial node embedding is then computed as

$$h_i^{(0)} = \phi^{(2)}(\phi^{(1)}(x_i)). \tag{2}$$

c) Edge structure.: Classic formulations of both the A-graph and the S-graph connect each room node only to the walls that belong to it [2], [4]. To enrich the structural information available to the encoder, we augment both graphs with two additional edge types, illustrated in Fig. 1.A in black. The full edge set therefore comprises three directed relation types. *Room-to-WS edges* capture spatial containment, connecting each room to its constituent wall segments, the only edge type present in the original formulations. *Room-to-room edges* encode adjacency between spaces that share a physical wall. *WS-to-WS edges* connect wall segments belonging to the same room in an angularly-sorted ring around the room center. Together, these augmented edge types encode spatial context across both abstraction levels within a single unified graph. This richer connectivity allows the GNN to propagate context both within and across hierarchy levels during message passing, which would not be possible with the original single-edge-type graphs.

B. Context-Aware Node Embeddings via GATv2

Raw node features are enriched through GATv2 [14], whose pairwise attention mechanism computes scores over the joint representation of both edge endpoints, allowing the model to learn distinct notions of relevance for neighbors of fundamentally different geometric nature — such as a WS node attending to its parent room versus its angular siblings. The attention score for each directed edge $(u \to v)$ is:

$$e_{uv} = \mathbf{a}^\top \,\mathrm{LeakyReLU}(\mathbf{W}_a \cdot [\mathbf{h}_u \| \mathbf{h}_v]) \tag{3}$$

where $\mathbf{W}_a$ and $\mathbf{a}$ are learnable parameters. The resulting attention weights α_{uv} update each node representation as:

$$\mathbf{h}_v^{(l+1)} = \sum_{u \in \mathcal{N}(v)} \alpha_{uv} \cdot \mathbf{W}_h \mathbf{h}_u^{(l)} \tag{4}$$

The encoder consists of L=2 GATv2 layers with 4 attention heads, hidden dimension 64, and output embedding dimension 32. ReLU and node dropout (p=0.15) are applied after the first layer only; attention dropout (p=0.12) is applied at both. The encoder is shared across both graphs, projecting rooms and

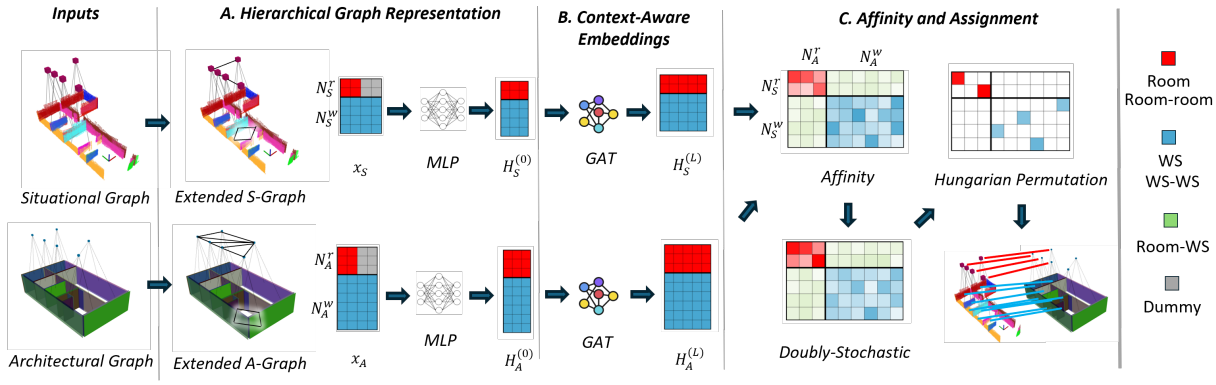

Fig. 1. **Overview of the proposed pipeline.** A shared MLP improves the initial node features, after which a shared GATv2 encoder produces structure-aware embeddings for both the A-graph (derived from BIM) and the S-graph (built online from LiDAR SLAM). A dot-product affinity matrix is computed, normalized via Sinkhorn with dummy-column padding to handle partial observations, and decoded into a hard one-to-one correspondence by the Hungarian algorithm.

wall segments into a common embedding space where cross-graph similarity is meaningful.

C. Cross-Graph Affinity and Differentiable Assignment

Building on the affinity-Sinkhorn pipeline introduced in PCA-GM [9] and adapted to scene graph matching in SG-PGM [13], node-to-node correspondences are established by computing a pairwise affinity matrix and solving the resulting combinatorial assignment through a continuous relaxation over soft assignment matrices.

Let $\mathbf{H}_1^{(L)} \in \mathbb{R}^{N_1 \times d}$ and $\mathbf{H}_2^{(L)} \in \mathbb{R}^{N_2 \times d}$ denote the node embeddings of the A-graph (N_1 nodes) and S-graph (N_2 nodes), with d the embedding dimension. The affinity matrix is computed as:

$$\mathbf{A} = \mathbf{H}_1 \mathbf{H}_2^\top \in \mathbb{R}^{N_1 \times N_2} \tag{5}$$

where A_{ij} measures the similarity between node i in the A-graph and node j in the S-graph. A plain dot product is used in place of the learnable bilinear form of PCA-GM [9], which was found to cause gradient explosions in preliminary experiments [15]. Instance Normalization is then applied to $\mathbf{A}$, standardizing each sample to zero mean and unit variance to ensure well-conditioned input to the Sinkhorn layer across graphs of variable size.

Because the S-graph is in general smaller than the A-graph ($N_2 \leq N_1$), $\mathbf{A}$ is rectangular and Sinkhorn cannot be applied directly. To recover a square matrix, $N_1 - N_2$ dummy columns are appended to $\mathbf{A}$, yielding $\hat{\mathbf{A}} \in \mathbb{R}^{N_1 \times N_1}$. The dummy columns absorb the probability mass of A-graph nodes that have no observed counterpart in the S-graph, providing a principled mechanism to handle graph matching when the affinity matrix is rectangular, without requiring explicit outlier rejection. The Sinkhorn algorithm [16] is then applied to $\hat{\mathbf{A}}$ to approximate the one-to-one correspondence constraint: by iteratively normalizing rows and columns to sum to one, each node is forced to distribute its total probability mass across candidates in the other graph, concentrating it on the most compatible matches, producing a doubly-stochastic matrix $\mathbf{S} \in [0,1]^{N_1 \times N_1}$. At inference, the $N_1 \times N_2$ submatrix of $\mathbf{S}$ corresponding to real S-graph nodes is passed to the Hungarian algorithm [17], extracting a globally optimal hard one-to-one correspondence in $\mathcal{O}(n^3)$.

IV. Experimental Evaluation

A. Setup

Datasets. We evaluate the proposed method on both synthetic and real data. For Synthetic evaluation, the MSD dataset [18] provides *real* architectural floor plans represented as A-graphs. S-graphs are simulated by applying perturbations to these A-graphs, mimicking the partial and noisy observations a robot builds online. We consider the *ws-room noise* condition, in which wall surface and room nodes are randomly dropped and geometric noise is applied. For Real evaluation, an S-graph was acquired through a Velodyne VLP-16 LiDAR sensor in a real indoor environment (RE), see Fig. 2. The corresponding A-graph was constructed from architectural blueprints. No retraining or fine-tuning was performed, making this a zero-shot generalisation test.

Baselines. We compare against iS-Graphs [4], the only prior work performing hierarchical semantic graph matching between S-graphs and A-graphs. To the best of our knowledge, no learned method addresses this problem, making iS-Graphs the most relevant baseline.

Evaluation metrics. Matching is evaluated as binary classification over all $N_2 \times N_1$ candidate node pairs, where N_2 is the number of S-graph nodes and N_1 is the number of A-graph nodes, as defined in Section III-C. We report **Precision**, **Recall**, **F1**, **Accuracy**, and **Matching time**. F1 is the primary metric of interest, as true negative dominance inflates accuracy and makes it a misleading measure of matching quality.

B. Training

The model is trained on the MSD dataset (Section IV-A), split into training (70 %), validation (15 %), and test (15 %) subsets using a stratified split that preserves the distribution of graph sizes across all sets.

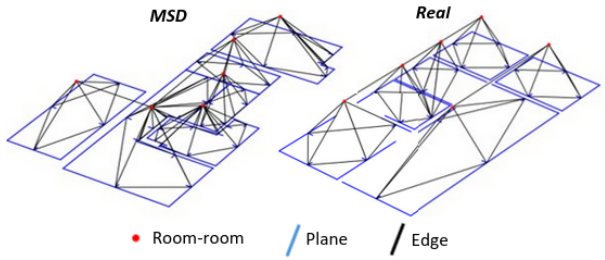

Fig. 2. Example graph used for evaluation: a floor plan from the MSD synthetic dataset (left) and the real indoor environment RE with its corresponding A-graph constructed from architectural blueprints (right).

The loss function is the Permutation Loss [9], a Binary Cross-Entropy applied element-wise between the predicted soft assignment matrix $\mathbf{S}$ and the ground-truth assignment matrix $\mathbf{P}_{\mathrm{gt}}$. BCE penalizes both false positives — node pairs incorrectly predicted as matches — and false negatives — true correspondences missed by the model — independently across all entries, better suiting the sparse bijective structure of the assignment problem than row-wise Cross-Entropy. Optimization is performed with AdamW (learning rate 1×10^{-3}, weight decay 5×10^{-5}, batch size 16). Early stopping on validation loss is applied to prevent overfitting. Hyperparameters were tuned using Optuna [19] with the Tree-structured Parzen Estimator (TPE) sampler.

C. Results and Discussion

MSD dataset. Table II reports matching performance on the MSD test split, comparing the proposed method against iS-Graphs [4]. Due to the factorial and exponential complexity of iS-Graphs, a timeout threshold was applied: samples exceeding 60 seconds are discarded, and the percentage completed within the timeout is reported. While iS-Graphs achieves higher precision and F1 on completed samples, only 86% of samples finish within the timeout, confirming that its combinatorial cost does not scale to real-time deployment. On the other side, the proposed method processes all samples deterministically, with competitive matching quality. In terms of computation time, the proposed method achieves an $82\times$ speedup over iS-Graphs (0.093 s vs. 7.66 s per sample), consistent with the 14% of samples that exceed the timeout threshold and are discarded from iS-Graphs' evaluation.

TABLE II
MATCHING PERFORMANCE ON THE MSD SYNTHETIC TEST SET.
†PERCENTAGE OF SAMPLES COMPLETED WITHIN THE TIMEOUT.

Method	Prec.%	Rec.%	F1%	Time(s)	Completed†
iS-Graphs [4]	99	92	95	7.66	86%
Ours	85	85	85	0.093	100%

Real dataset. Table III reports performance on the real environment (RE). Despite being trained exclusively on synthetic data, the proposed method outperforms iS-Graphs in F1. iS-Graphs achieves higher precision but at the cost of recall: its conservative combinatorial search only commits to matches satisfying strict geometric constraints, leaving many true correspondences unmatched under real sensor noise. The GNN-based matcher, trained to penalise both false positives and false negatives, learns a more flexible assignment that better tolerates domain shift, yielding higher recall and overall F1. Notably, iS-Graphs completes within the timeout here due to the relatively low number of nodes in RE. Nevertheless, the proposed method remains $9\times$ faster (0.021 s vs. 0.191 s per sample); on larger environments, where node count increases, its combinatorial cost would render it impractical.

TABLE III
MATCHING PERFORMANCE AND INFERENCE TIME ON REAL LIDAR ENVIRONMENT RE. BEST RESULTS IN **BOLD**.

Method	Prec.%	Rec.%	F1%	Time [s]	Completed
iS-Graphs [4]	**88**	53	67	0.191	100%
Ours	80	**88**	**84**	**0.021**	100%

V. CONCLUSION

We presented a learned, end-to-end differentiable pipeline for hierarchical scene graph matching between BIM-derived Architectural Graphs and robot-built Situational Graphs, achieving competitive matching quality while running orders of magnitude faster than the combinatorial baseline and transferring zero-shot to real LiDAR environments. A current limitation is that the model does not account for structural symmetry, which may lead to ambiguous correspondences in symmetric floor plans. As future work, we plan to first improve precision and recall of the learned matcher, and subsequently address symmetry detection in environments with structurally similar rooms and repeated wall configurations.

REFERENCES

[1] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3D scene graph construction and optimization," in *Robotics: Science and Systems (RSS)*, 2022.

[2] H. Bavle, J. L. Sanchez-Lopez, M. Shaheer, J. Civera, and H. Voos, "Situational graphs for robot navigation in structured indoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9107–9114, 2022.

[3] R. W. Hendrikx, E. de Gelder, D. Habets, P. Pauwels, E. Torta, and J. P. van den Heuvel, "Connecting semantic building information models and robotics: An application to 2D LiDAR-based localization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 654–11 660.

[4] M. Shaheer, J. A. Millan-Romera, H. Bavle, J. L. Sanchez-Lopez, J. Civera, and H. Voos, "Graph-based global robot localization informing situational graphs with architectural graphs," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 9155–9162.

[5] F. Boniardi, T. Caselitz, R. Kümmerle, and W. Burgard, "A pose graph-based localization system for long-term navigation in CAD floor plans," *Robotics and Autonomous Systems*, vol. 112, pp. 84–97, 2019.

[6] H. Yin, J. M. Liew, W. L. Lee, M. H. Ang, K.-W. Yeoh, and J. Tian, "Semantic localization on BIM-generated maps using a 3D LiDAR sensor," *Automation in Construction*, vol. 146, p. 104759, 2022.

[7] R. Wang, J. Yan, and X. Yang, "Neural graph matching network: Learning Lawler's quadratic assignment problem with extension to hypergraph and multi-graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5261–5279, 2022.

[8] J. Stührenberg and K. Smarsly, "LIO-BIM – coupling lidar inertial odometry with building information modeling for robot localization and mapping," *Advanced Engineering Informatics*, vol. 66, p. 103477, 2025.

[9] R. Wang, J. Yan, and X. Yang, "Learning combinatorial embedding networks for deep graph matching," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3056–3065.

[10] T. Yu, R. Wang, J. Yan, and B. Li, "Learning deep graph matching with channel-independent embedding and hungarian attention," in *International Conference on Learning Representations (ICLR)*, 2020.

[11] Y. Lin, M. Guo, P. Hu, C. Wang, and J. Lv, "Graph matching with bi-level noisy correspondence," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[12] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4938–4947.

[13] Y. Xie, A. Pagani, and D. Stricker, "SG-PGM: Partial graph matching network with semantic geometric fusion for 3D scene graph alignment and its downstream tasks," in *arXiv preprint arXiv:2403.19474*, 2024.

[14] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" in *International Conference on Learning Representations (ICLR)*, 2022.

[15] M. Giorgi, "Graph neural network-based scene graph matching for robot localization," Master's thesis, University of Pisa, 2024.

[16] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343–348, 1967.

[17] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1–2, pp. 83–97, 1955.

[18] S. van Engelenburg, T. Lucassen, F. I. Karahanoglu, and M. A. Westenberg, "Modified swiss dwellings: A 2D floor plan dataset for semantic scene understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023.

[19] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 2623–2631.