Integrating Prior Observations for Incremental 3D Scene Graph Prediction

Marian Renz[1], Felix Igelbrink[1], Martin Atzmueller[1,2]

***Abstract*—3D semantic scene graphs (3DSSG) provide compact structured representations of environments by explicitly modeling objects, attributes, and relationships. While 3DSSGs have shown promise in robotics and embodied AI, many existing methods rely mainly on sensor data, not integrating further information from semantically rich environments. Additionally, most methods assume access to complete scene reconstructions, limiting their applicability in real-world, incremental settings. This paper introduces a novel heterogeneous graph model for incremental 3DSSG prediction that integrates additional, multi-modal information, such as prior observations, directly into the message-passing process. Utilizing multiple layers, the model flexibly incorporates global and local scene representations without requiring specialized modules or full scene reconstructions. We evaluate our approach on the 3DSSG dataset, showing that GNNs enriched with multi-modal information such as semantic embeddings (e.g., CLIP) and prior observations offer a scalable and generalizable solution for complex, real-world environments. The full source code of the presented architecture will be made available at https://github.com/m4renz/incremental-scene-graph-prediction.**

***Index Terms*—3D Semantic Scene Graphs, Graph Neural Network, Heterogeneous Graph Learning, RGB-D Sequence**

I. Introduction

Semantic scene graphs (SSGs) offer a structured and compact representation of visual environments by explicitly modeling objects, their attributes, and inter-object relationships in a semantically rich way. Initially developed for 2D image understanding, the extension of SSGs into the 3D domain [1] has gained significant traction, particularly in robotics, where spatial reasoning and situational awareness are critical. 3D semantic scene graphs (3DSSGs) enable environmental modeling by incorporating geometric and topological information. In consequence, this allows for a more accurate interpretation of complex scenes.

Furthermore, 3DSSGs serve as a powerful bridge between raw sensory input and high-level semantic understanding by facilitating the integration of multi-modal information, such as additional sensory data and even common-sense knowledge. As a result, they are increasingly adopted in robotics research as a foundational representation for embodied AI systems that require both perceptual grounding and semantic reasoning. The construction or generation of 3DSSG from sensor data has therefore become a prominent topic in machine learning and robotics [2]–[4].

With progress in 3DSSG inference using Graph Neural Networks (GNNs), a variety of approaches have emerged that integrate 3DSSG generation with additional information sources. These methods leverage supplementary data to refine object and relationship predictions, enhance generalization across environments, and support downstream tasks such as navigation [5], exploration [6], or task planning [7]. However, while all integration approaches independently show promising results, they lack a generalized mechanism that is agnostic to the modality of the utilized information, making them highly dependent on the utilized training data.

Moreover, most existing approaches focus on inferring 3DSSGs from fully reconstructed scenes, where complete geometric information is available at inference time [8]. This makes these approaches impractical for many real-world tasks, where a scene is typically captured incrementally from a stream of sensor data. Incremental SSG generation requires models to utilize information acquired from prior observations to predict and interpret new sensor inputs.

In this work, we present a method for incremental 3DSSG generation by integrating the sub-tasks required for the SSG construction into a multi-layered architecture. This design allows for flexible incorporation of multi-modal information into the model architecture without the need for specialized modules. This is not only limited to new features, it also extends to topologically different graphs.

Central to our approach is a heterogeneous scene graph design that fuses sensor data and observations from previous time steps across *global* and *local layers*. Global layers provide spatial, geometric, and semantic context for the entire scene, while local layers integrate current sensor data. The proposed model efficiently stores and integrates spatial, geometric, and semantic features by embedding them directly into the message-passing process, eliminating the need to store numerous point-cloud segments or time-series data.

The main contributions of our work are summarized as follows:

- We propose a novel heterogeneous graph model for 3D scene graph generation that integrates multi-modal information for incremental prediction.
- We evaluate the proposed model on the 3DSSG dataset for per-frame incremental scene graph prediction.
- We demonstrate the robustness of our model against erroneous predictions in prior observations.

The DFKI Niedersachsen (DFKI NI) is sponsored by the Ministry of Science and Culture of Lower Saxony and the VolkswagenStiftung. This work is supported by the Federal Ministry of Research, Technology and Space (BMFTR) within the ExPrIS project (grant number: 01IW23001)

[1] Cooperative and Autonomous Systems, DFKI Niedersachsen, German Research Center for Artificial Intelligence, Osnabrück, Germany
{firstname}.{lastname}@dfki.de

[2] Semantic Information Systems, Osnabrück University, Osnabrück, Germany

II. Related Work

SceneGraphFusion [9] was the first approach to generate scene graphs incrementally. The authors infer local 3D scene graphs from partial point cloud segments derived from individual RGB-D frames in the 3DSSG dataset [8]. These local scene graphs are subsequently fused into a global graph. A variation of this method has also been applied using only RGB image sequences [10]. However, the proposed model utilizes only the updated geometrical information when predicting novel frames. The existing global scene graph, generated from previous frames, remains invisible to the model. As a result, the model does not benefit from prior knowledge of the scene structure and instead predicts each frame independently. In our approach, we directly integrate prior predictions by linking instances from frames to previously predicted nodes, thus enabling our model to benefit from earlier observations without the need to store the fully segmented point cloud.

Most similar to our idea, Feng et al. [11] incorporate historical predictions for incremental scene graph generation, using a recurrent mechanism to integrate the last *m* processed graphs and embedding a global graph representation as a one-hot encoded matrix into the prediction process. In contrast, we do not encode global information explicitly; instead, we integrate it directly into the message passing by linking past predictions and matching node instances. This approach allows newly integrated information to directly enhance downstream SSG prediction. Furthermore, we explore the use of heterogeneous GNNs to improve the integration of semantically relevant information.

In the context of multi-modal integration, several recent approaches have investigated heterogeneous graph structures and external knowledge sources. Ma et al. [12] infer relationship types based on three top-level categories from the 3DSSG dataset and apply heterogeneous message passing on the learned graph structure. Directed Spatial Commonsense Graphs (D-SCG) [13] incorporate heterogeneous information from ConceptNet [14] with 3DSSGs to localize objects in partial 3D scenes. Knowledge-Scene Graph Networks [15] integrate external knowledge curated from multiple sources using GB-Net [16], embedding this knowledge directly into the message passing process. While these approaches successfully integrate multi-modal information for their respective tasks, none have been applied to the problem of incremental 3DSSG generation.

III. Method

A. Dataset and Preprocessing

To train and evaluate our proposed method, we utilize the 3DSSG dataset, which extends the 3RScan dataset [17] with scene graph annotations for over one thousand indoor 3D scenes created using RGB-D reconstruction.

We use the RIO27 label set, which features a total of 27 object categories and 16 relationship categories derived from [18]. After filtering out invalid scenes, we obtained a total of 1,320 usable 3D scenes from the dataset. Each scene consists of an annotated reconstruction of the geometry, available as a 3D mesh, the complete scene graph, as well as the raw RGB-D frames and their poses used to reconstruct the scene.

Since, for incremental scene graph prediction, neither the full scene geometry nor the complete graph is available to the model at inference time, we extracted for every RGB-D frame F_t the visible geometry as a point cloud with instance annotations, along with the currently visible portion of the ground-truth scene graph. Additionally, for each frame F_t, we included the partial scene graph constructed from the preceding frames $\{F_0, \ldots, F_{t-1}\}$ (see Section III-B).

B. Graph Model

The core intuition behind our heterogeneous modeling approach is to connect previously observed objects in the sensor data stream to the same objects in newly recorded frames. This enables the model to leverage information from earlier observations during prediction.

When represented as a scene graph, this results in a two-layer architecture: a global scene graph that accumulates observations, objects, and relationships from previous frames, and a local scene graph constructed from the sensor data of a single frame. The target task is to predict the classes and relationships in the local scene graph by utilizing both the sensor data and the global scene graph.

Assuming a partial global scene graph with already classified objects and predicates, as provided by the dataset preprocessing (see Section III-A), the first step is to perform object segmentation on the current frame to identify visible object instances for the local graph. For this work, we use the ground-truth segments from the 3DSSG dataset. To construct the local graph, we convert the depth frame into a point cloud and add bidirectional edges between objects that are less than 0.5 meters apart, following the approach in [9]. During training, we also use ground-truth information to match local nodes to previously seen nodes in the global graph.

Nodes and edges in the global and local scene graphs are modeled as distinct node and edge types within a single heterogeneous graph. Additional edges connect global and local nodes for all matched node pairs (see Fig. 1), allowing information to flow from the global to the local graph.

The node features are similar to those used in [9]. For each object, 256 points are sampled from the point cloud. Additionally, a hand-crafted descriptor is computed, consisting of the center c and standard deviation std of the sampled points, bounding box side lengths l, w, and h, the maximum bounding box length L, and the bounding box volume V. Global nodes also include information from previous predictions, either as a class label or as a CLIP [19] embedding of the predicted label.

To merge the local scene graph into the global one, the points of matched nodes are downsampled again to 256 points, and the descriptor is recalculated. Ground-truth labels and instance identifiers remain unchanged. New nodes and edges are added directly to the global graph.

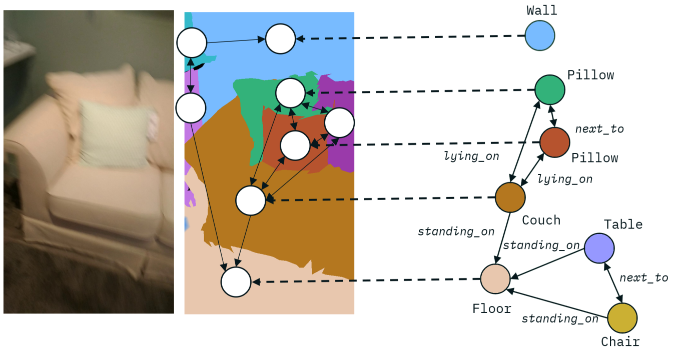

Fig. 1. Example of the heterogeneous scene graph. The left image shows the RGB-D frame from [8], the middle shows the segmented frame with the unpredicted local scene graph, and the right shows the already predicted global scene graph. Dashed lines represent the edges between matched nodes.

C. Graph Neural Network Architecture

Each node's feature representation is constructed by embedding 256 sampled points using a PointNet encoder [20], and concatenating the resulting point feature with a geometric descriptor vector. For global nodes, an additional feature vector is included based on the ground truth label, either as a one-hot encoded class index or a CLIP embedding of the text label. Edge features are computed by subtracting centroid and standard deviation offsets, and applying logarithmic ratios of bounding box dimensions, following the approach of [9]. Specifically, the edge feature vector for an edge e_{ij} between nodes i and j is defined as:

$$[c_j - c_i,\ \mathrm{std}_j - \mathrm{std}_i,\ \log(\tfrac{l_j}{l_i}, \tfrac{w_j}{w_i}, \tfrac{h_j}{h_i}),\ \log(\tfrac{L_j}{L_i}),\ \log(\tfrac{V_j}{V_i})],$$

and is passed through a two-layer MLP. Message passing is performed using a two-layer GNN, either a heterogeneous GraphSAGE [21], [22] or HGT [23]. Node classification is conducted via a two-layer MLP applied to the updated node features. For edge classification, the updated source and target node features are concatenated with the edge feature vector. Note that edge features are not used during message passing due to limitations of the employed GNN layers. All layers use ReLU activation, layer normalization, and a dropout rate of 0.5, except for the final layers of each sub-network.

As baselines, we include homogeneous GraphSAGE and SGFN [9] applied only to the local frame graph. Additionally, we evaluate a homogeneous version of the global-local heterogeneous graph, where missing label features in local nodes are replaced with -1, and missing CLIP embeddings with zero vectors to ensure consistent feature dimensions. Edges between global and local nodes in this setting do not carry ground truth labels. We also test a variant of the heterogeneous architecture without ground truth features in the global layer. To assess robustness, we additionally train the homogeneous SAGE and HGT/heterogeneous SAGE with 20 % and 50 % falsified global labels, resulting in incorrect CLIP embeddings.

D. Training Details

The model is trained using a composite loss function that combines a weighted node classification loss and a weighted binary edge classification loss, scaled by a factor $\alpha = 40$ for positive edge classes. The total loss is defined as

$$\mathcal{L} = w_n \mathcal{L}_n + \alpha w_e \mathcal{L}_e, \tag{1}$$

where the weights are computed based on the inverse log-frequency of class occurrences, i. e.,

$$w_n = \frac{10}{\log(n_n)}, \quad w_e = \frac{10}{\log(n_e)} + 1. \tag{2}$$

Only predictions for local nodes and edges are considered during loss and gradient computation across all models. Note that node classification is treated as a multi-class problem, with only one correct ground truth class per object, whereas edge prediction is a multi-label task, allowing multiple valid ground truth labels per edge.

Training is conducted for up to 100 epochs, with early stopping triggered if the validation loss does not improve for 5 consecutive epochs. For SAGE models, a learning rate of 0.0001498 is used with a step scheduler (decay factor $\gamma = 0.05$, step size = 30), while HGT models are trained with a learning rate of 0.0001 and a step scheduler with $\gamma = 0.05$ and a step size of 20. The SGFN model follows the training procedure of the GraphSAGE model, using the hyperparameters and loss function described in [9].

IV. Evaluation

We train all models on the preprocessed dataset described in Chapter III-A, following a 0.8/0.1/0.1 split for training, validation, and test data. For all models, we predict and evaluate only nodes and edges within the local frame, and not in the global graph. Additionally, we evaluate models trained on data with 20 % and 50 % falsified ground truth labels in the global layer to assess the robustness of the approach.

A. Metrics

We evaluate four aspects of incremental scene graph prediction: (1) Node classification, measured by mean accuracy across all local nodes; (2) Edge classification, evaluated using mean recall; (3) Relationship prediction, measured by the number of correctly predicted ground truth triples using the ng-Recall@k metric [24], which determines the fraction of detected ground truth triples among the top-k predicted triples of the local scene graph; and (4) Node classification for previously unseen nodes, i.e., nodes appearing for the first time in the sequence and not yet present in the global graph, also evaluated using mean accuracy.

B. Scene Graph Prediction

Our results (see Table I) highlight several key findings. Models that operate solely on individual frames, such as SGFN and GraphSAGE, achieve the highest accuracy on previously unseen nodes, indicating that the introduction of

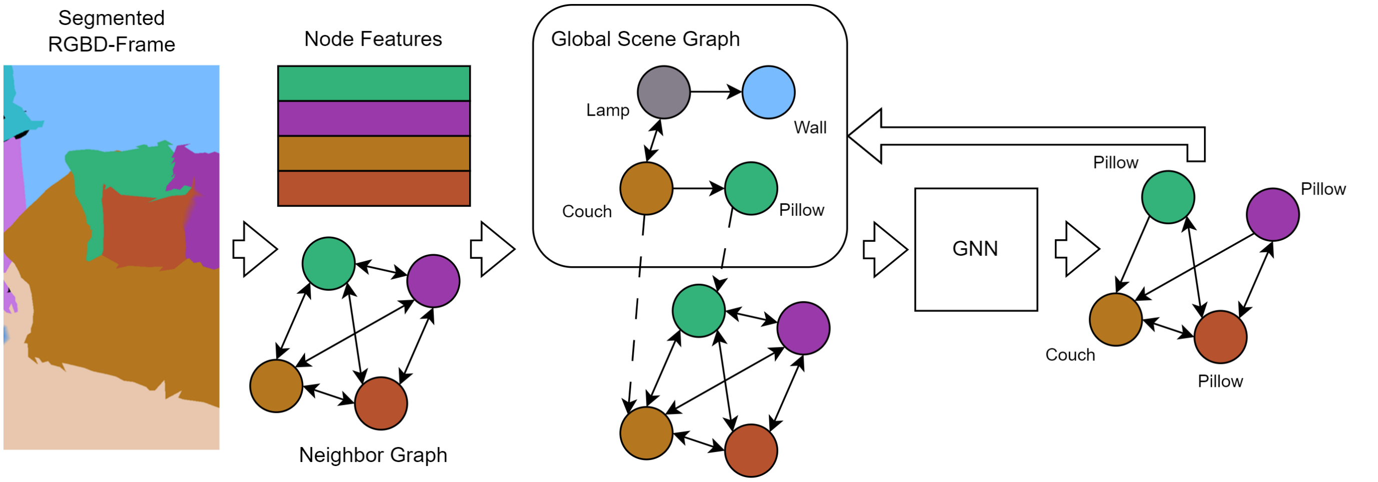

Fig. 2. Pipeline of the proposed approach. At timestep t, a local frame graph is constructed from a segmented RGB-D frame based on a neighborhood graph and segment-specific node features. The local graph is then connected to a globally constructed scene graph from frame $t-1$. After message passing, node and edge classes are predicted, and the local graph is merged into the global graph.

TABLE I
RESULTS FOR THE INCREMENTAL 3D SCENE GRAPH GENERATION. THE FIRST SECTION SHOWS RESULTS FOR HOMOGENEOUS GNNS, THE SECOND SECTION FOR HETEROGENEOUS GNNS AND THE THIRD SECTION FOR HETEROGENEOUS GNNS WITH ADDITIONAL EDGES.

	Acc@1	Acc@5	Rec	ng-R@50	ng-R@100	U-Acc@1	U-Acc@5
SGFN	0.48	0.80	0.31	0.00	0.00	**0.40**	**0.83**
SAGE	0.28	0.65	0.64	0.00	0.00	0.37	0.80
SAGE+label	0.92	0.98	**0.81**	0.02	0.01	0.30	0.70
SAGE+clip	**0.98**	**0.99**	0.80	0.14	0.18	0.36	0.79
HetSage-plain	0.34	0.76	0.6	0.07	0.09	0.31	0.71
HGT-plain	0.42	0.81	0.63	0.16	0.20	0.21	0.58
HetSAGE+label	0.73	0.98	0.74	0.53	0.61	0.31	0.70
HGT+label	0.95	0.95	0.69	0.71	0.78	0.24	0.62
HetSAGE+clip	**0.98**	**0.99**	0.75	0.58	0.69	0.33	0.75
HGT+clip	**0.98**	**0.99**	0.77	**0.80**	**0.84**	0.29	0.73
HetSAGE+clip+add	**0.98**	**0.99**	0.76	0.68	0.76	0.32	0.76
HetSAGE+clip+add-only	**0.98**	**0.99**	0.73	0.51	0.61	0.34	0.78

prior classifications can impair generalization in node classification. However, these models perform poorly in predicting the overall scene graph structure. Incorporating a heterogeneous scene graph without additional semantic features (HetSAGE-plain and HGT-plain) yields only minor improvement on the relationship prediction, suggesting that structural information alone is insufficient. In contrast, enriching node features with simple labels or CLIP embeddings leads to consistently better performance, where clip embeddings lead to better results in all cases. Homogeneous models perform equally well or better than heterogeneous models on node and edge prediction metrics, while heterogeneous models excel in relationship prediction. Among these, the HGT+label/CLIP models achieve the highest relationship prediction performance, with HGT+CLIP also showing competitive results across other metrics. These results suggest that heterogeneous models are well-suited for capturing the rich semantic structures present in 3DSSGs.

C. *Robustness Against False Labels*

When introducing falsified labels into the global scene graph, we observe a general decline in performance across all metrics, except for mean edge recall, which slightly improves under $20\ \%$ falsified labels for all models except HGT. The most pronounced drop occurs in relationship prediction, with reductions ranging from 0.06 to 0.34 for $20\ \%$ falsified labels and from 0.11 to 0.46 for $50\ \%$ falsified labels at $k=50$ and 0.07 to 0.37 for $20\ \%$ falsified labels and from 0.13 to 0.52 for $50\ \%$ falsified labels at $k=100$. Interestingly, the accuracy on previously unseen nodes is less affected, showing only a modest decline between 0.02 and 0.07, suggesting that the models still effectively learn to generalize prior classification features. Similar to the evaluation with correct labels, the homogeneous GNN with integrated prior observations outperforms its heterogeneous counterparts on node and edge prediction metrics. In contrast, heterogeneous models achieve better performance in relationship prediction. However, they also show the largest performance drop in this

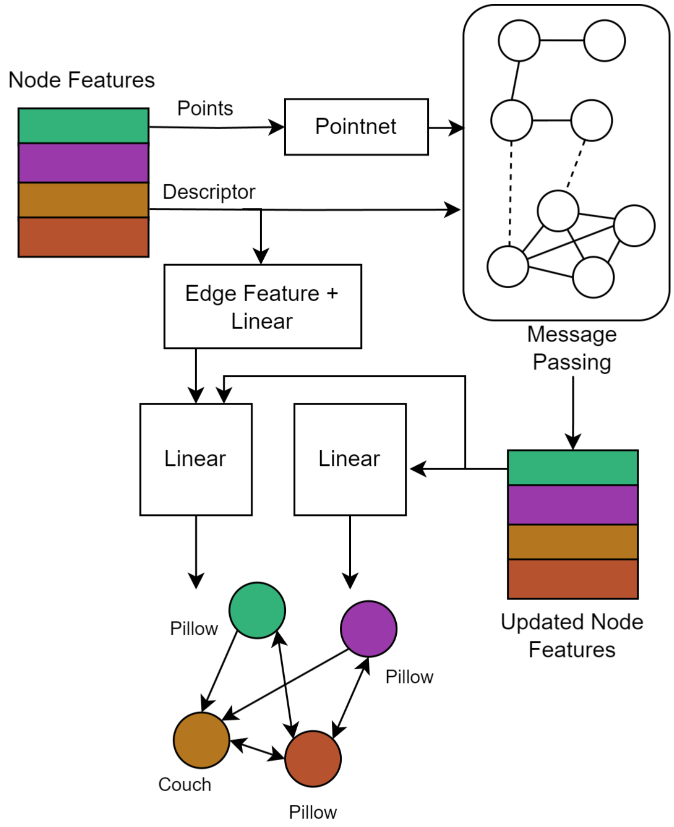

Fig. 3. GNN architecture for 3D scene graph generation. For node features, the sampled points are passed through a PointNet encoder and concatenated with a geometric descriptor. For edge features, the descriptor for an edge e_{ij} connecting nodes i and j is formed by concatenating descriptors d_i and d_j, which are then passed through two linear layers. After message passing, node classes are predicted using two linear layers applied to the updated node features. For edge classification, the edge feature of e_{ij} is concatenated with the updated node features of nodes i and j, and passed through two linear layers as well.

category when label noise is introduced. Despite 20 % falsified labels representing a substantial level of noise, the relatively minor performance degradation in node and edge classification tasks across all models indicates a degree of robustness. However, the drop for relationship prediction suggests a strong reliance on prior observations, which has to be mitigated for real application scenarios. For most metrics (excluding unseen node accuracy), models trained with 20 % falsified labels consistently outperform those trained with 50 %, reinforcing the notion that the learned prior features remain informative even under moderate label corruption.

D. Integration of additional layers

To evaluate the flexibility of our heterogeneous structure, we introduce an additional edge type between global nodes, providing information about the centrality of objects within a scene, derived from geometric data. This layer is computed by performing geometric collision checks between objects in a scene using the FCL library [25]. Based on the amount of overlap between pairs of colliding objects, a hierarchy is derived using the harmonic centrality metric [26]. This hierarchy is then integrated into the heterogeneous model as a new edge type between global nodes, providing a topologically different subgraph.

Since GraphSAGE and HGT do not natively support edge features, we implement a specialized edge layer for this edge type, which updates the target node using only the edge features:

$$x_i' = \gamma(x_i) + \sum_{j \in \mathcal{N}(i)} \phi(x_{ji}), \qquad (3)$$

where $\mathcal{N}(i)$ denotes the neighbors of node i, x_i' is the updated node feature, x_i is the original target node feature, x_{ji} is the edge feature, and γ and ϕ are learnable linear transformations. The edge feature x_{ji} consists of the amount of geometric overlap between nodes, their respective bounding boxes, and the difference in harmonic centrality as described above. For all other edge types, GraphSAGE message passing without edge features is used.

We evaluate this new layer both as an additional edge type between global nodes and as a substitute for the original edges resulting from the integration of local SSGs into the global SSG.

Results (see tab. I and II) show that adding this additional information between global nodes yields performance comparable to HetSAGE+CLIP, with a slight improvement on previously unseen nodes and relationship prediction when using the additional edge type together with the integrated global edges. The same tendency is seen in the evaluation with falsified labels.

Although the improvements in the reported metrics are relatively modest, the results demonstrate that the proposed model can seamlessly integrate multi-modal information into the message passing process without requiring external modules.

V. Conclusion

We present a heterogeneous graph model that enables the integration of multi-modal information for incremental 3D scene graph generation. Specifically, we connect information from sensor data frames to a concise global scene graph model built from previous observations. We show that the integration of these prior observations benefits the overall prediction performance on both homogeneous and heterogeneous GNN architectures.

The proposed model demonstrates strong predictive performance for heterogeneous GNN architectures. The integration of additional information sources yields comparable results, indicating that the model effectively incorporates multi-modal data. While homogeneous GNNs achieve high performance on straightforward classification tasks, heterogeneous GNNs are more suited to capture the heterogeneity of multi-modal, semantic information. Furthermore, the heterogeneous graph learning framework offers flexibility for incorporating task-specific information or external knowledge graphs without altering the core architecture.

TABLE II
RESULTS FOR THE 3D SCENE GRAPH PREDICTION WITH FALSIFIED FEATURE CLASSES IN THE GLOBAL LAYER. THE FIRST SECTION SHOWS MODELS TRAINED WITH 20 % RANDOMLY FALSIFIED LABELS, THE SECOND SECTION SHOWS MODELS TRAINED WITH 50 % RANDOMLY FALSIFIED LABELS.

	Acc@1	Acc@5	Rec	ng-R@50	ng-R@100	U-Acc@1	U-Acc@5
SAGE+clip+0.2	**0.94**	**0.99**	**0.82**	0.08	0.11	**0.34**	**0.78**
HetSAGE+clip+0.2	0.91	0.98	0.76	0.33	0.38	0.29	0.72
HGT+clip+0.2	0.89	0.98	0.74	**0.5**	**0.59**	0.24	0.65
HetSAGE+clip+add+0.2	0.88	0.98	0.75	0.34	0.4	0.28	0.72
HetSAGE+clip+add-only+0.2	0.89	0.98	0.73	0.24	0.27	0.30	0.73
SAGE+clip+0.5	**0.86**	**0.98**	**0.81**	0.03	0.05	**0.33**	**0.76**
HetSAGE+clip+0.5	0.8	0.96	0.73	0.19	0.23	0.27	0.70
HGT+clip+0.5	0.81	0.96	0.73	**0.37**	**0.44**	0.23	0.64
HetSAGE+clip+add+0.5	0.79	0.96	0.72	0.22	0.25	0.29	0.70
HetSAGE+clip+add-only+0.5	0.76	0.95	0.64	0.18	0.21	0.30	0.71

Future work will explore applying this architecture to full-scale 3D semantic mapping for real-world robotics tasks, integrating additional prior knowledge sources to enhance inference and support explainability.

REFERENCES

[1] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, S. Savarese, 3D scene graph: A structure for unified semantics, 3D space, and camera, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 5664–5673.

[2] N. Hughes, Y. Chang, L. Carlone, Hydra: A real-time spatial perception system for 3D scene graph construction and optimization, in: Robotics: Science and Systems XVIII, Robotics: Science and Systems Foundation, 2022. doi:10.15607/rss.2022.xviii.050.

[3] H. Bavle, J. L. Sanchez-Lopez, M. Shaheer, J. Civera, H. Voos, *S-graphs+:* real-time localization and mapping leveraging hierarchical representations, IEEE Robot. Autom. Lett. 8 (8) (2023) 4927–4934. doi:10.1109/lra.2023.3290512.

[4] S. Looper, J. Rodriguez-Puigvert, R. Siegwart, C. Cadena, L. Schmid, 3D VSG: Long-term semantic scene change prediction through 3D variable scene graphs, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 8179–8186. doi:10.1109/ICRA48891.2023.10161212.

[5] Z. Ravichandran, L. Peng, N. Hughes, J. D. Griffith, L. Carlone, Hierarchical representations and explicit memory: Learning effective navigation policies on 3D scene graphs using graph neural networks, Proceedings - IEEE International Conference on Robotics and Automation (2022) 9272–9279doi:10.1109/ICRA46639.2022.9812179.

[6] X. Li, D. Guo, H. Liu, F. Sun, Embodied semantic scene graph generation, in: A. Faust, D. Hsu, G. Neumann (Eds.), Proceedings of the 5th Conference on Robot Learning, Vol. 164 of Proceedings of Machine Learning Research, PMLR, 2022, pp. 1585–1594.

[7] C. Agia, K. M. Jatavallabhula, M. Khodeir, O. Miksik, V. Vineet, M. Mukadam, L. Paull, F. Shkurti, Taskography: Evaluating robot task planning over large 3D scene graphs, in: A. Faust, D. Hsu, G. Neumann (Eds.), Proceedings of the 5th Conference on Robot Learning, Vol. 164 of Proceedings of Machine Learning Research, PMLR, 2022, pp. 46–58.

[8] J. Wald, H. Dhamo, N. Navab, F. Tombari, Learning 3D semantic scene graphs from 3D indoor reconstructions, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020. doi:10.1109/cvpr42600.2020.00402.

[9] S.-C. Wu, J. Wald, K. Tateno, N. Navab, F. Tombari, SceneGraphFusion: Incremental 3D scene graph prediction from RGB-D sequences, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2021. doi:10.1109/cvpr46437.2021.00743.

[10] S.-C. Wu, K. Tateno, N. Navab, F. Tombari, Incremental 3D semantic scene graph prediction from RGB sequences, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (2023) 5064–5074doi:10.1109/CVPR52729.2023.00490.

[11] M. Feng, C. Yan, Z. Wu, W. Dong, Y. Wang, A. Mian, History-enhanced 3D scene graph reasoning from RGB-D sequences, IEEE Transactions on Circuits and Systems for Video Technology (2025) 1–1doi:10.1109/TCSVT.2025.3548308.

[12] Y. Ma, H. Liu, Y. Pei, Y. Guo, Heterogeneous graph learning for scene graph prediction in 3D point clouds, ECCV (2024) 274–291doi:10.1007/978-3-031-73347-5_16.

[13] F. Giuliari, G. Skenderi, M. Cristani, A. D. Bue, Y. Wang, Leveraging commonsense for object localisation in partial scenes, IEEE Trans. Pattern Anal. Mach. Intell. 45 (10) (2023) 12038–12049. doi:10.1109/TPAMI.2023.3272523.

[14] R. Speer, J. Chin, C. Havasi, ConceptNet 5.5: An open multilingual graph of general knowledge, Proc. Conf. AAAI Artif. Intell. 31 (1) (12 Feb. 2017). doi:10.1609/AAAI.V31I1.11164.

[15] Y. Qiu, H. I. Christensen, 3D scene graph prediction on point clouds using knowledge graphs, IEEE International Conference on Automation Science and Engineering 2023-August (2023). doi:10.1109/CASE56687.2023.10260650.

[16] A. Zareian, S. Karaman, S.-F. Chang, Bridging knowledge graphs to generate scene graphs, in: Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII, Springer-Verlag, Berlin, Heidelberg, 2020, pp. 606–623. doi:10.1007/978-3-030-58592-1_36.

[17] J. Wald, A. Avetisyan, N. Navab, F. Tombari, M. Niessner, RIO: 3D object instance re-localization in changing indoor environments, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, 2019. doi:10.1109/iccv.2019.00775.

[18] C. Zhang, J. Yu, Y. Song, W. Cai, Exploiting edge-oriented reasoning for 3D point-based scene graph analysis, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2021, pp. 9700–9710. doi:10.1109/CVPR46437.2021.00958.

[19] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International conference on machine learning, PmLR, 2021, pp. 8748–8763.

[20] C. R. Qi, H. Su, K. Mo, L. J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[21] W. L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, Adv. Neural Inf. Process. Syst. 2017-December (2017) 1025–1035.

[22] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: Proc. European Semantic Web Conference (ESWC), 2018, pp. 593–607.

[23] Z. Hu, Y. Dong, K. Wang, Y. Sun, Heterogeneous graph transformer, in: Proceedings of The Web Conference 2020, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 2704–2710. doi:10.1145/3366423.3380027.

[24] N. Gkanatsios, V. Pitsikalis, P. Koutras, P. Maragos, Attention-translation-relation network for scalable scene graph generation, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), IEEE, 2019. doi:10.1109/iccvw.2019.00218.

[25] J. Pan, S. Chitta, D. Manocha, Fcl: A general purpose library for collision and proximity queries, in: 2012 IEEE international conference on robotics and automation, IEEE, 2012, pp. 3859–3866.

[26] P. Boldi, S. Vigna, Axioms for centrality, Internet Mathematics 10 (3-4) (2014) 222–262.