Using 2D scene graphs as an enabler for DT topology

Nayun Kim*, Atacan Kural Avgoren*, Mohammad Adnan Mohammad Alrabab'h*, Fiona Collins and Changyu Du
*equally contributed to this work
The Chair of Computational Modeling and Simulation, Technical University of Munich, Arcisstraße 21, 80333 Munich, Germany
E-mail(s): ny.kim@tum.de, atacan.avgoren@tum.de, m.alrababh@tum.de, fiona.collins@tum.de, changyu.du@tum.de

Abstract: Scan-to-BIM research has gained significant attention, yet many reconstruction methodologies overlook the crucial topological relationships in Building Information Modeling (BIM). To address this challenge, we propose using Scene Graphs to capture contextual relationships in images, focusing on hallways and offices at the Technical University of Munich (TUM). Our approach involves fine-tuning an existing Scene Graph Generation (SGG) model and proposing an in-house model both aiming to predict scene graphs, by integrating object detection and predicate detection methods. The fine-tuned SGG model with the benchmark archived the recall@50 of 95.57 in predicate classification mode. Comparatively, our in-house model attained a recall of 65.12 for the overall scene graph generation. Despite these promising results, some limitations remain, such as low object detection accuracy and the exclusion of non-relationships, as well as the evaluation being limited to qualitative comparisons and was done independently for each method. These limitations highlight areas for future work. Nevertheless, this study offers a proof of concept for integrating scene graph predictions into Scan-to-BIM workflows while identifying areas for further improvement.

Keywords: Closed-Vocabulary 2D Scene-Graph, Computer Vision, Rule-Based ML, Transfer Learning, Digital Twin

Erschienen in Tagungsband 35. Forum Bauinformatik 2024, Hamburg, Deutschland, DOI: 10.15480/882.13529

1 Introduction

In this work, we explore the use of scene graph generation to enhance the Digital Twinning method and Scan-to-BIM process. Current Scan-to-BIM methodologies often overlook crucial topological relationships, leading to incomplete representations of buildings and spaces. This limitation is particularly evident in complex environments like spaces with open areas or void spaces, where spatial definitions rely on more than just physical proximity. Scene graphs can capture intricate semantics by explicitly modeling objects (e.g., doors, windows, walls) and their relationships (e.g., "hang-on",

"above"). By integrating 3D scene graphs into the Scan-to-BIM process, we aim to improve the accuracy of classification and clustering, especially in these sophisticated spatial arrangements. Our main goal is to investigate methods for automatically enriching Scan-to-BIM images with object and relationship semantics. This approach has the potential to significantly enhance the handling of complex spaces and improve the overall accuracy of digital twin creation in the built environment.

2 Literature Review

Several studies have explored methods for generating scene graphs from images. Zellers et al. [1] introduced the Stacked Motif Network (MOTIFNET) for scene graph parsing, which captures higher-order dependencies in scene graphs. Tang et al. [2] presented VCTREE, a model that dynamically constructs tree structures to capture visual context in images. Both models were evaluated on general datasets like Visual Genome and showed improvements in scene graph generation. Some research shows the application of relationship-based structures similar to scene graphs in construction and building information modelling. Bueno and Bosché [3] emphasize the importance of understanding relationships between elements in construction projects. They demonstrate how these relationships provide vital semantic and topological context, enabling precise validation and quality control in specific domains like railways. While these studies have advanced scene graph generation, they lack specific focus on building components. Common datasets like COCO and Visual Genome omit crucial building elements such as walls, roofs, and floors, vital objects in building context, leaving a gap in relationship definitions for these objects. Our research addresses this limitation by investigating how existing scene graph models perform in a building context. We aim to evaluate their applicability and performance when dealing with building-specific components and relationships, an area not thoroughly explored in previous studies. This work seeks to contribute to scene graph generation techniques tailored for the built environment, potentially enhancing Scan-to-BIM processes and digital twinning in the building domain.

3 Methodology

We employ two approaches for building-related scene graph generation. The first approach involves fine-tuning the Scene Graph Generation with the Visual Genome Benchmark (SGGVG) Model developed by Kaihua Tang et al., which is a PyTorch implementation of their paper [4]. This model was initially trained on a widely adopted Visual Genome (VG) split [5], which contains 108k images with the most frequent 150 object categories and 50 predicate categories. This model builds on a pre-trained, but relatively old versioned, Faster R-CNN model for object detection [6]. The second approach utilizes an in-house model based on a Multi-Layer Perceptron (MLP) and leverages the recent pre-trained weights of the Faster R-CNN model.

3.1 Data Preparation

The dataset used in this study is derived from the research by Pan et al. [7], consisting of images from buildings at the Technical University of Munich (TUM). It includes 377 images with 23 object classes, ranging from predominant elements like walls, roofs, and ceiling lights to smaller objects such as light switches and manual call points, all annotated with segmented masks. To provide a

consistent dataset for both methodologies, we converted the segmented masks into bounding boxes (Bbox). We then used a rule-based Python script to automate the annotation of obvious predicates and performed semi-automated manual annotation for the scene graph, including 13 predicate classes. No annotations were made for non-relationships. For the first approach, it was necessary to convert the common dataset into the Visual Genome format. We used the Iterative Message Passing (IMP) [8] toolkit to compress the data into the H5 format, making it compatible with the pre-trained models.

3.2 SGG with Fine-tuned Benchmark Model

As illustrated in Figure 1, The SGGVG model builds upon the Mask R-CNN framework [9] by adding a relation head for scene graph generation. By toggling specific components, the model supports three modes of inference: Predicate Classification (PredCls), using only the Relation Head for predicate prediction, Scene Graph Classification (SGCls), using the Box Head and Relation Head for predicates and bounding box prediction, and Scene Graph Detection (SGDet), using all components for all prediction including scene graph.

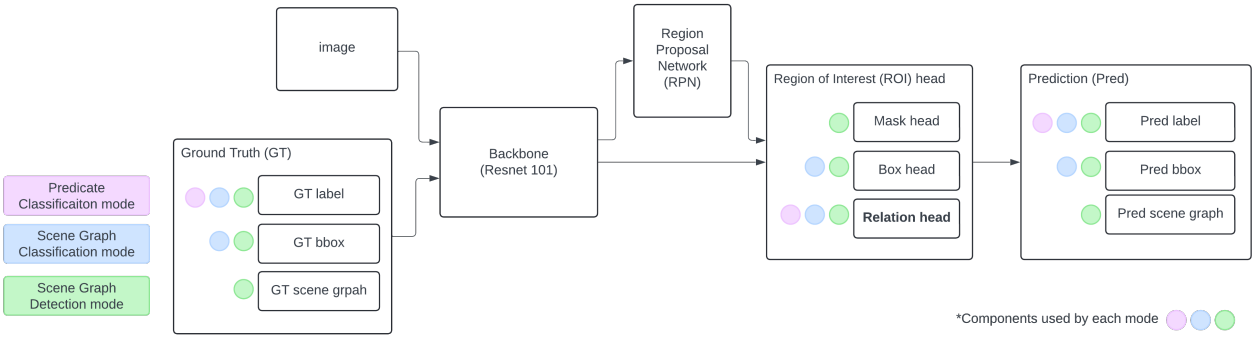

Figure 1: Flow diagram of SGGVG Model

SGGVG model can perform various object detection and predicate detection inferences. For object detection, it can utilize pre-trained weights from Faster R-CNN [6] and Mask R-CNN [9]. For predicate detection, it supports various methods for prediction during inference, including MOTIFS [1], Iterative Message Passing (IMP) [8], VCTree [1], and the Unbiased-Causal-TDE [4] model. The repository [10] offers pre-trained weight of the SGGVG model, trained and evaluated on the Visual Genome benchmark [5], which contains 108k images with the most frequent 150 object categories and 50 predicate categories [4]. The pre-trained object detector utilizes a Faster R-CNN with a ResNeXt-101-FPN backbone, achieving mAP scores of 26.35 and 28.14 on the VG validation and test sets, respectively [10]. The pre-trained SGGVG model is trained in SGDet mode, utilizing all components for scene graph generation. In Predicate Classification (PredCls) mode, which only evaluates the relation head, the model achieves Recall@20, Recall@50, and Recall@100 scores of 59.64, 66.11, and 67.96, on VG dataset respectively. Before fine-tuning, qualitative testing with sample images revealed that the pre-trained model failed to accurately detect primary objects in the building context, leading to irrelevant scene graphs in SGDet mode [7]. Consequently, we decided to fine-tune both the relation head and the Faster R-CNN model using our custom VG dataset.

3.3 SGG with In-house Model

In the In-house Model, we followed two main steps: Object Detection and Predicate Detection. In the first step, our model predicts objects in the images along with their labels and bounding box coordinates. In the second step, the predicate detection model predicts the predicates between these detected objects. Performance metrics for each step of this model are discussed in section 4.2.

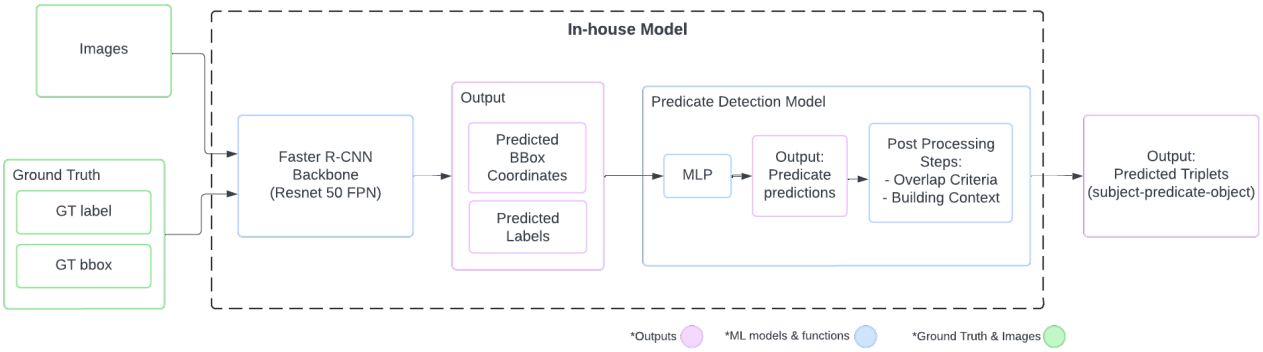

Figure 2: Flow diagram of In-house Model

As first step, we implemented a object detection method to detect objects in images using the Faster R-CNN model with a ResNet-50 backbone. This model effectively captures detailed features and precise object borders on a image. The model was fine-tuned to detect 23 specific classes relevant to our dataset, such as door, window, wall, ceiling and stair. Our Predicate Detection Model identifies predicates between objects in images using their locations and labels. It begins with a Multi-Layer Perceptron (MLP) to predict potential predicates, which are then refined by post-processing filters that eliminate irrelevant predicates and ensure logical spatial topology. The MLP model processes bounding box coordinates and object labels, which are predicted by the Object Detection Model, to detect various predicate types. It uses ReLU activation and dropout to prevent over-fitting, transforming input features into a robust 64-dimensional space, and finally predicts predicates with a softmax function. During training, the MLP model addresses class imbalance by assigning more weight to less frequent classes in the dataset. For instance, our dataset contains numerous instances of lighting but very few of hand-rails. This balancing approach enhances prediction accuracy and prevents overfitting. The process involves feeding data batches, calculating loss, and adjusting model parameters over multiple epochs.

Post-processing consists of two steps: Overlap Criteria and Building Context. The Overlap Criteria filtration calculates the intersection area between bounding boxes of two objects, classifying predicates into high, low, or nearly-none overlap categories. This validation ensures that predicted predicates make logical spatial sense, e.g., "mounted-on" requiring high overlap and "adjacent-to" needing minimal overlap, aligning with typical building and construction arrangements. Building Context filtration ensures the model's predictions are relevant by limiting them to a predefined set of valid subject-predicate-object triplets specific to the building domain. These predefined triplets, representing possible subject-predicate-object combinations, were created from our manual annotations on the

images. These triplets prevent irrelevant predicates and ensure contextual accuracy within the building domain.

4 Results

4.1 SGG with Fine-tuned Benchmark Model

As illustrated in Figure 1, The SGGVG model builds upon the Mask R-CNN framework [9] by adding a relation head for scene graph generation. By toggling specific components, the model supports three modes of inference: Predicate Classification (PredCls), using only the Relation Head for predicate prediction, Scene Graph Classification (SGCls), using the Box Head and Relation Head for predicates and bounding box prediction, and Scene Graph Detection (SGDet), using all components for all prediction including scene graph.

Table 1: Comparison of Original (left) and Fine-tuned (right) Model performance (Recall@k)

Mode	R@20	R@50	R@100
PredCls	59.5	66.0	67.9
SGCls	35.8	39.1	39.9
SGDet	25.1	23.1	36.9

Mode	R@20	R@50	R@100
PredCls	94.24	95.57	95.57
SGCls	48.17	50.90	51.89
SGDet	8.44	8.99	9.21

In the PredCls mode, we observed a significant improvement of about 35%, achieving an accuracy of over 90%. The SGCls mode also showed an enhancement, with accuracy increasing by approximately 10%. However, in the SGDet mode, the fine-tuned model unexpectedly achieved lower scores than the original model. This underperformance in SGDet mode can be attributed to several factors. The bbox head used for predictions in this mode struggled to detect objects in our dataset. This difficulty originates from the pre-trained Faster R-CNN model's initially low accuracy (around 35%) for object detection. Additionally, there was a mismatch between our building-centric dataset and the Visual Genome dataset used for pre-training, which lacks building-related objects. Furthermore, resizing our images from 2592x1728 to 1028x684 which was configured for training the model before resulted in the loss of small objects such as switches and lighting fixtures.

4.2 SGG with In-house Model

The Jaccard index, also known as the Intersection over Union (IoU), was calculated for the object detection component of the project, resulting in a score of 92.12%. This high score indicates a strong overlap between the detected objects and the ground truth, as detailed in the accompanying Table 2.

Table 2: Performance metrics of the Object Detection Model in the In-house Model

Model	Input	Output	IoU (Jaccard Index)
fine-tuned Faster RCNN	Images	Object bbox coordinates and labels	92.12

For the Predicate Detection Model, two sets of inputs were tested: the first using ground truth bounding boxes and labels, and the second using predicted boxes and labels from the previous Object Detection Model. The use of ground truth bounding boxes and labels resulted in higher accuracy metrics. This may be due to some missing or mislabeled objects in the Object Detection Model.

Table 3: Performance metrics of the Predicate Detection Model in the In-house Model

Input	Output	Recall	Precision	F1-Score
GT BBox and Labels	Triplets (s,v,o)	71.17	38.03	49.59
Predicted BBox and Labels	Triplets (s,v,o)	65.12	36	46.36

5 Discussion

Our research compares two scene graph generation methodologies: a fine-tuned benchmark model (SGGVG) and an in-house model. The SGGVG model, pre-trained on a dataset with 75% background pairs as non-relationships [4], predicts and ranks predicates and handles the case of non-relationship. It presents top K final predictions with confidence scores, including possible non-relationships, and is therefore evaluated using Recall@K. Unlike the SGGVG model, our in-house model does not handle non-relationship cases. Instead, it assumes that a relationship exists for all input pairs. It predicts predicates between each object in the images and then applies a confidence level filtration to eliminate those below 50%. Subsequently, it applies two post-processing filters: Building Context and Overlap Criteria. Training the in-house model with non-relationship pairs may enhance its effectiveness and make it more applicable to other datasets. We offer a qualitative comparison to highlight each model's strengths.

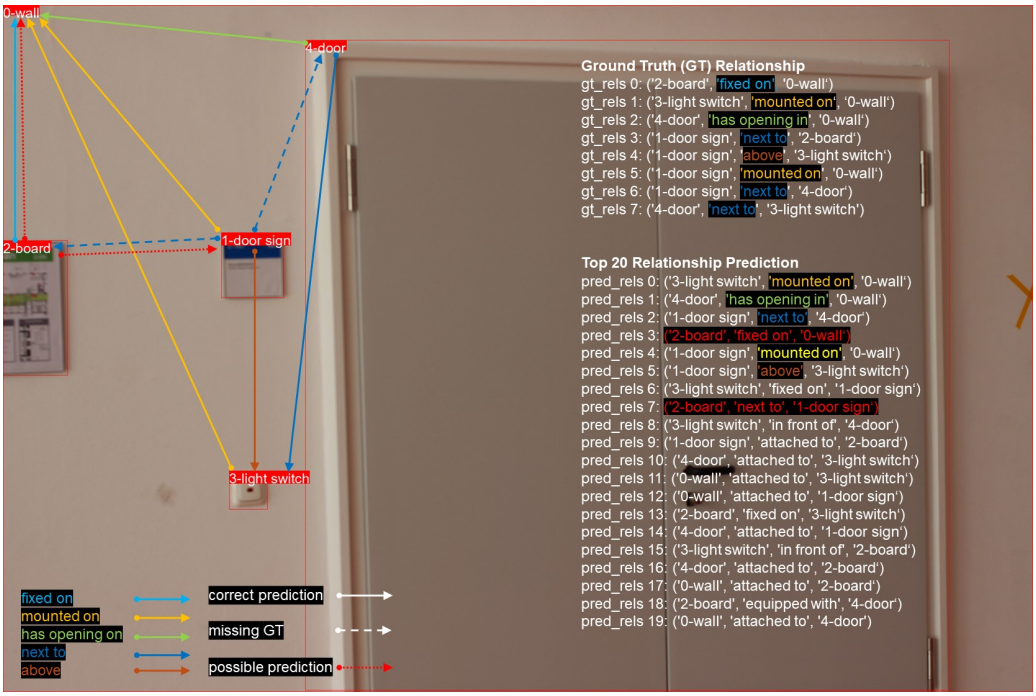

Figure 3: Visualization in predicate prediction from SGG with Fine-tuned Benchmark Model

For qualitative comparison, we visualized the predictions of both models on a sample image. Figure 3 shows the inference results of the fine-tuned SGGVG model in PredCls mode. Out of 8 total predictions, the top 6 match the ground truth, while the model failed to detect 2 'next to' predicates. This is likely due to the 'next-to' class having an accuracy of 85.22%, significantly lower than other classes which exceed 95% accuracy. Moreover, the model's 8th ranked prediction was 'board' 'next-to'

'door sign', which, although not present in the ground truth annotation, is semantically correct. This demonstrates the model's ability to not only predict predicates missing from the ground truth but also to evaluate non-relationships by ranking predictions. Such capability allows for a more comprehensive assessment of the model's performance beyond the limitations of ground truth annotations. Figure 4 shows the inference results of the In-House model. It's predictions cover all ground truth predicates. In addition to these predictions, it has detected four extra predicates which are reasonable.

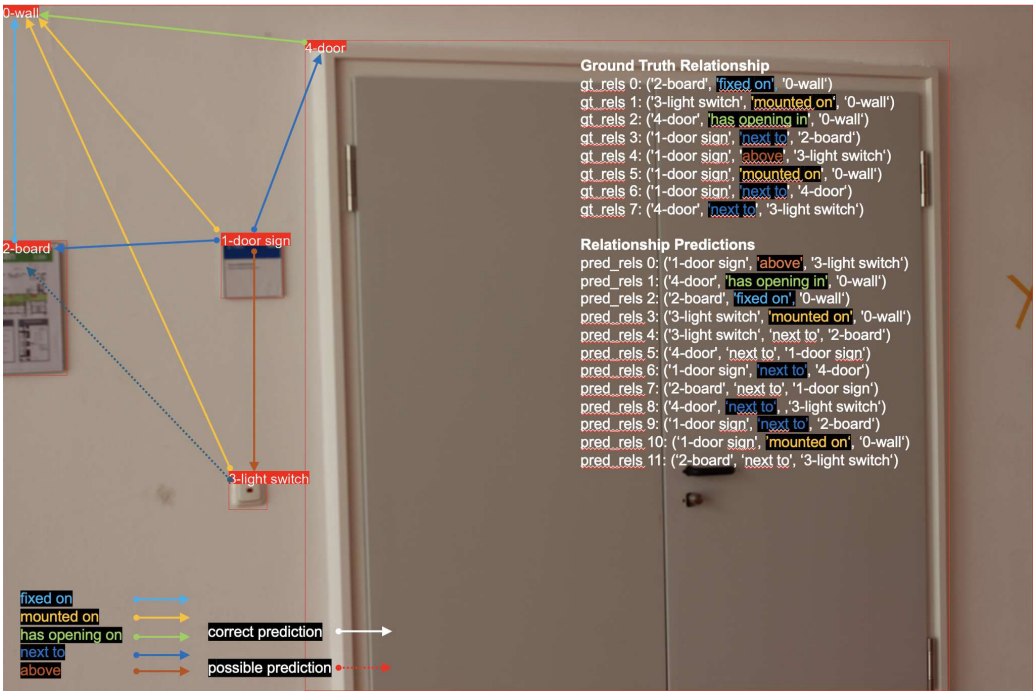

Figure 4: Visualization in predicate prediction from SGG with In-house Model

Future improvements for SGGVG could include updating its object detection model, while our in-house model could be enhanced by adding non-relationship prediction and addressing training bias.

6 Conclusion

This study investigated the application of scene graph generation techniques in the built environment domain, revealing both potential and limitations. Our findings highlight that while open-source computer vision models are effective for general object and relationship recognition, they face constraints in specialized domains like the built environment, emphasizing the need for domain-specific training data. We demonstrated methods for applying scene graph generation to built environment applications through fine-tuning a benchmark model and developing a domain-specific model, analyzing the merits of each approach. Our research indicates the potential of scene graph generation to enrich the Scan-to-BIM process with topological information, while also highlighting the need for further investigation into which specific relationships (e.g., "mounted-on", "next-to", "attached-to") are most critical for BIM topology. Future work should focus on developing built environment-specific datasets, improving results with modern object detection models, and integrating these technologies into Scan-to-BIM workflows. In conclusion, this study lays the groundwork for enriching the Scan-to-BIM process with

topology, which is crucial for creating parametric models and enhancing spatial reasoning capabilities. By capturing relationships between building elements, we contribute to the digital transformation of the built environment sector, enabling more sophisticated and context-aware digital building models.

Author Contributions

Nayun Kim, Atacan Kural Avgoren, and Mohammad Adnan Mohammad Alrabab'h contributed equally. Kim led the "SGG with Fine-tuned Benchmark Model" section, Avgoren the "SGG with in-house model" section, and Alrabab'h the "Data preparation" and "Literature Review" sections. Fiona Collins and Changyu Du supervised the research.

References

[1] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. DOI: 10.48550/arXiv.1711.06640. arXiv: 1711.06640 [cs.CV]. [Online]. Available: https://doi.org/10.48550/arXiv.1711.06640.

[2] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Learning to compose dynamic tree structures for visual contexts", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 7, pp. 1801–1814, 2019.

[3] M. Bueno and F. Bosché, "Pre-processing and analysis of building information models for automated geometric quality control", in *Automation in Construction*, vol. 165, 2024. DOI: https://doi.org/10.1016/j.autcon.2024.105557. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580524002930.

[4] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 1872–1888, 2020.

[5] R. Krishna, *Visual genome: Connecting language and vision using crowdsourced dense image annotations*, https://paperswithcode.com/dataset/visual-genome, 2017.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", 2015.

[7] Y. Pan, A. Braun, I. Brilakis, and A. Borrmann, "Enriching geometric digital twins of buildings with small objects by fusing laser scanning and ai-based image recognition", *Automation in Construction*, vol. 129, p. 103 808, 2022.

[8] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Github - scene-graph-tf-release", 2017.

[9] F. et al., *Github - maskrcnn-benchmark: Fast, modular reference implementation of instance segmentation and object detection algorithms in pytorch*, https://github.com/facebookresearch/maskrcnn-benchmark, 2019.

[10] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, *Github repository - scene graph benchmark*, https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch, 2020.