TB-HSU: Hierarchical 3D Scene Understanding with Contextual Affordances

Wenting Xu[1], Viorela Ila[2], Luping Zhou[1], Craig T. Jin[1]

[1]School of Electrical and Computer Engineering, The University of Sydney
[2]School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney
{wenting.xu, viorela.ila, luping.zhou, craig.jin}@sydney.edu.au

Abstract

The concept of function and affordance is a critical aspect of 3D scene understanding and supports task-oriented objectives. In this work, we develop a model that learns to structure and vary functional affordance across a 3D hierarchical scene graph representing the spatial organization of a scene. The varying functional affordance is designed to integrate with the varying spatial context of the graph. More specifically, we develop an algorithm that learns to construct a 3D hierarchical scene graph (3DHSG) that captures the spatial organization of the scene. Starting from segmented object point clouds and object semantic labels, we develop a 3DHSG with a top node that identifies the room label, child nodes that define local spatial regions inside the room with region-specific affordances, and grand-child nodes indicating object locations and object-specific affordances. To support this work, we create a custom 3DHSG dataset that provides ground truth data for local spatial regions with region-specific affordances and also object-specific affordances for each object. We employ a Transformer Based Hierarchical Scene Understanding model (TB-HSU) to learn the 3DHSG. We use a multi-task learning framework that learns both room classification and learns to define spatial regions within the room with region-specific affordances. Our work improves on the performance of state-of-the-art baseline models and shows one approach for applying transformer models to 3D scene understanding and the generation of 3DHSGs that capture the spatial organization of a room. The code and dataset are publicly available.

Code, Dataset — Github.com/WentingXu3o3/TB-HSU

Introduction

Understanding visual scenes is fundamental to machine perception, requiring not only accurate object representation, but also consideration of how these objects might be arranged in their functional zones. In this work, we take a navigational and task-oriented perspective to scene understanding to assist robots in deciding which areas to explore to complete a task. In this light, functional considerations require appropriate representation and consideration of the potential usage of objects within the scene, often referred to as object affordances. The annotation of affordances can be labor-intensive and thus many 3D scene datasets (Chang et al. 2017; Dai et al. 2017; Straub et al. 2019) lack affordance annotations for objects, with only a few, such as AI2-THOR (Kolve et al. 2017), 3DSSG (Wald et al. 2020) and 3D AffordanceNet (Deng et al. 2021) providing affordance annotations. In AI2-THOR (Kolve et al. 2017), a fixed affordance list with actionable attributes is assigned to each object class, such as "ArmChair - Receptacle, Moveable" and "Book - Openable, Pickupable". The 3DSSG (Wald et al. 2020) dataset similarly defines fixed affordance annotations with actionable attributes for each object class, with examples such as "Armchair - Lying on, Sleeping, Moving" and "Book - Reading, Storing in, Storing on." Additionally, 3D AffordanceNet (Deng et al. 2021) takes a different approach and annotates specific parts of the object with affordance labels. For example, for a mug, different vertices are associ-

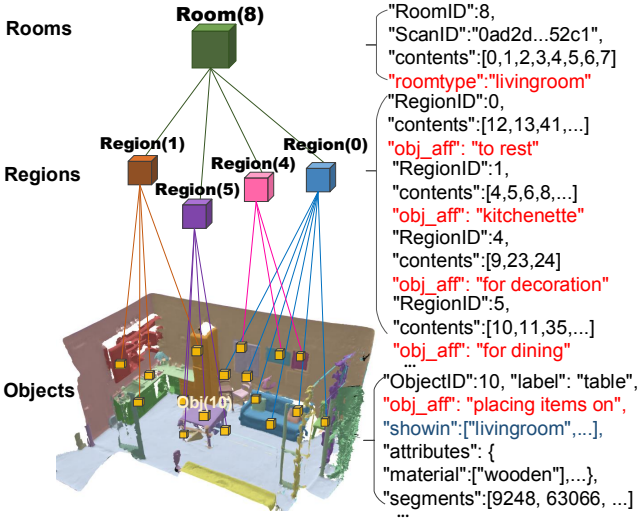

Figure 1: **Overview:** The affordances of objects vary depending on the granularity of the context. We propose a multi-task method, Transformer Based Hierarchical Scene Understanding (TB-HSU), employing a transformer-based model for 3D scene understanding to classify the room and identify the region, forming a 3D Hierarchical Scene Graph (3DHSG) with three layers: Objects, Regions, and Rooms.

ated with affordances such as "pour, wrap, contain". However, all current datasets with affordance annotations provide either a fixed affordance list or vary the affordance based on the particular physical contact points without considering how affordances change depending on context.

In this work, we assume that the affordances of objects should vary depending on the spatial context, i.e., the precision and extent of the surroundings and location. For example, chairs placed beside a table and the table situated next to a kitchen stove primarily serve the purpose of dining. "Narrowing down" or "zooming in" on the region around the chairs and table would indicate the chair serves as a place for sitting, and the table serves as a surface for placing items. Similarly, chairs around a sofa facing a TV are primarily used for resting, but when we "zoom in" to the region around the chairs and sofa, the chair and the sofa both serve the purpose of sitting. Continuing in this fashion, a bed with a pillow within a room primarily serves the purpose of sleeping, while narrowing down on the bed and pillow indicates that the pillow serves the purpose of propping the head or body, while the bed serves the purpose of lying on. A pillow on an armchair serves the purpose of resting; "zooming in" on the pillow indicates the purpose of propping, while the armchair indicates the purpose of sitting. While the changes in affordances are transparent and obvious to human observers, these transitions in the affordance of objects have not really been addressed in existing 3D scene datasets.

Contributions

Firstly, we introduce the **3D H**ierarchical **S**cene **G**raph (3DHSG) dataset that extends the 3DSSG dataset (Wald et al. 2020), which itself extends the 3RScan dataset (Wald et al. 2019). 3DHSG captures the spatial organization for a 3D scene in a three-layered graph, where nodes represent objects, regions within rooms, and rooms. Object nodes include context-specific affordances, while region nodes cluster objects with the same region-specific affordances, and room nodes contain the type of room.

Secondly, we develop a **T**ransformer **B**ased **H**ierarchical **S**cene **U**nderstanding (TB-HSU) model, which automatically constructs the 3DHSG for a room using instance-segmented point cloud data and object semantic labels within a multi-task learning framework. The TB-HSU model generates a hierarchical 3D scene graph that captures the spatial organization of a room.

Thirdly, we validate our proposed TB-HSU method on both our custom 3DHSG dataset and two public benchmarks and demonstrate our promising performance over multiple baseline models consistently. We also showcase how 3DHSG contributes to improving GPT-4o's capacity in question answering, *e.g.* finding an object not in the scene.

Related Work

3D Scene Graphs

Current work often uses hierarchical 3D Scene Graphs (Hughes, Chang, and Carlone 2022; Hughes et al. 2024; Rosinol et al. 2021, 2020; Kim et al. 2020; Wald et al. 2020; Wu et al. 2021) to represent information such as object relationships within 3D scenes. A 3D scene graph is a layered graph where nodes represent spatial concepts at multiple levels of abstraction (from objects to regions and rooms, etc.), and edges represent relationships between these concepts (*e.g.*, Figure 1). Armeni et al. (Armeni et al. 2019) pioneered the use of 3D scene graphs in comprehensive semantic scene understanding and proposed the first algorithms to construct a graph that includes the semantics of objects, rooms, and cameras for each layer. The Hydra model (Hughes, Chang, and Carlone 2022; Hughes et al. 2024) and SceneGraphFusion model (Wu et al. 2021) introduced a 3D hierarchical scene graph for indoor environments from sensor data. The SceneGraphFusion model incrementally builds a semantic scene graph simultaneously while performing 3D mapping. The Hydra model incrementally builds a 3D hierarchical scene graph in real-time that captures the spatial organization of a scene, e.g., it creates a graph with layers such as buildings, rooms, places, objects and agents, and a semantic 3D mesh. Object segmentation, identification, and relationships are a significant focus for many 3D scene graph models. The affordance of objects is often considered a node attribute in these 3D scene graphs. For example, in (Wald et al. 2020), 3D semantic scene graphs are created where nodes contain a hierarchy of object classes and also object affordances. In this work, we focus on affordances and consider affordance as a concept that varies with spatial organization and that can assist with solving the 3D scene understanding problem. Our focus stems from a navigational and task-oriented perspective in which varying the affordance with spatial organization can assist with decision-making.

In this work, we follow the tradition of using hierarchical 3D scene graphs to represent indoor environments with varying levels of spatial organization (e.g. objects, regions, and rooms) in a three-layer 3D scene graph (see Figure 1).

LLMs and VLMs In 3D Scene Reasoning

The success of large language models (LLMs) such as GPT (OpenAI 2024) and LLaMa (Touvron et al. 2023) and vision-language models (VLMs) such as CLIP (Radford et al. 2021) has inspired their application to 3D scenes (Hong et al. 2023; Gu et al. 2024; Jatavallabhula et al. 2023; Maggio et al. 2024). For example, OpenScene (Peng et al. 2023) embeds dense 3D point features with image pixels and text to answer open-vocabulary queries. An instance of a query can be to find points that are conceptually associated with words such as "soft", "kitchen", or "work". ConceptFusion (Jatavallabhula et al. 2023) creates a 3D map with "pixel-aligned" features that permit multiple kinds of open queries. These multimodal queries can relate to arbitrary concepts, including affordance-related questions, and can return spatial regions consistent with the query. Moreover, ConceptGraph (Gu et al. 2024) builds open-vocabulary 3D scene graphs emphasizing spatial relationships between objects through reasoning with LLMs. Their scene graphs can interface with LLMs to give useful facts to robots about surrounding objects's traversability and utility. Consider also the real-time robotics algorithm Clio (Maggio et al. 2024), designed to build task-driven 3D scene graphs with embed-

ded open-set semantics. Clio forms task-relevant clusters of object primitives based on a set of natural language tasks, clustering the scene into task-relevant semantic regions such as "Kitchenette", "Workspace," and "Conference Room." We find our ideas most similar to Clio, which also considers semantic concepts as varying in "granularity". For Clio, the focus is task-driven with the objective of creating 3D scene graphs that retain task-relevant objects and regions only. This approach leaves users or robots with the issue of determining a suitable task list for navigational purposes, which could be particularly difficult when they first encounter a new environment. We certainly agree with the perspective that object affordances should be considered from a task-dependent framework but suggest that object affordances should relate to spatial context as well. We suggest that a 3D hierarchical scene graph representing spatial organization with varying affordances can be used for exploratory as well as task-driven purposes and will likely be useful in queries to LLMs and VLMs.

Methods

The 3DHSG Dataset

Overview of the 3DHSG Dataset In order to explore the machine analysis of 3D scenes based on the affordances of objects at varying levels of spatial context, we create a custom dataset referred to as the hierarchical scene graph (3DHSG) dataset. To be clear, we take a navigational framework for the classification of the affordance of objects. While the juxtaposition of navigation and object affordance may seem odd, this approach stems from the fact that a robot with a particular task objective in focus will need to consider that within a building, there are different rooms serving differing and multiple purposes and holding objects of varying affordances. In this way, a decision can be made about which rooms to approach based on a room affordance level to evaluate the potential for task completion. Further, upon reaching a particular room, the robot will need to evaluate whether local regions within the room provide the possibility for task completion. This evaluation would be based on more specific aspects of object affordances that match the level of detail associated with the local region of the room. In this way, we integrate a navigational approach into solving task objective problems that systematically refine the spatial and functional information, with the understanding that appropriate object affordance data should be available at each stage to evaluate the likelihood of task completion. Depending upon whether the navigational information is available as stored data or requires direct exploration determines whether the robot must explore the area directly.

To better understand the dataset, it is helpful for us to begin by describing three of its aspects: (i) the data available, (ii) the two machine learning tasks associated with the dataset, and (iii) the development of a hierarchical scene graph (3DHSG) capturing the spatial organization of the scene. The top node of the graph is simply a building that consists of a collection of rooms. A room is then characterized by collections of point clouds that have object labels, and Cartesian coordinate data available. Once a room has been selected, the first machine-learning task is room classification based on the objects within the room. A second machine learning task is to then divide the room into local spatial regions that define region-specific affordances. These local spatial regions form the nodes for the second level of the 3DHSG. The first layer of the 3DHSG consists of nodes representing the individual objects (and their locations) with object-specific affordances. In a sense, we solve the two machine-learning tasks in order to develop the 3DHSG.

To complete the descriptive overview of the dataset, we describe the data that are stored with the object nodes. There are two levels of affordance data: object-specific and region-specific. These two levels of object affordances are stored in a two-component vector associated with each object. Each object node also contains a list of rooms that are commonly associated with the object.

Details of the 3DHSG Dataset An outline sketch of the custom 3DHSG dataset is provided in Figure 1. To create this dataset, we start with the real-world 3D scene dataset 3RScan (Wald et al. 2019), which comprises 1482 3D reconstructions of 478 natural indoor environments. Instance-segmented point cloud data and object semantic labels are provided with the 3RScan dataset, while object attributes come from 3DSSG (Wald et al. 2020). For a given room, we describe the room category and provide each object with a region-specific affordance and object-specific affordance. Objects with the same region-specific affordance form a local region. Further, we also provide a room list for each object indicating the rooms in which it is usually found. The data are then organized into a 3DHSG as described above.

In order to determine both the region-specific and object-specific affordance data along with the room category, we use the descriptive data provided by GPT (OpenAI 2024), ConceptNet (Speer, Chin, and Havasi 2017) and 3DSSG (Wald et al. 2020). These descriptive data for the object affordances were then evaluated manually by humans for their suitability for region-specific and object-specific affordances. Specifically, we conducted initial region groups based on the intersections of objects' bounding boxes and then manually refined the groups. The manually selected affordance data for each object was stored as a tuple with two elements: <region-specific affordance, object-specific affordance>. For our dataset, we define twelve different room types, 27 different region-specific affordances, and 87 different object-specific affordances. More details are provided in the Appendices.

Similarly to (Hughes et al. 2024), the graph in our 3DHSG is defined as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where the set of nodes $\mathcal{N}$ can be partitioned into $\ell = 3$ layers. Specifically, $\mathcal{N} = \bigcup_{i=1}^{\ell} \mathcal{N}_i$, where the lowest layer, $\mathcal{N}_1$, describes the objects; the next layer, $\mathcal{N}_2$, describes the regions; and the top layer, $\mathcal{N}_3$, describes the rooms. Each node $n \in \mathcal{N}_i$ in layer i has only a single parent, meaning it shares an edge with at most one node in the layer $\mathcal{N}_{i+1}$ above. This reflects that each object belongs to a single region, and each region belongs to a single room. Additionally, each node $n \in \mathcal{N}_i$ in layer i only shares edges with nodes in adjacent layers, i.e., $\mathcal{N}_{i-1}$ or $\mathcal{N}_{i+1}$. Thus, edges in the second layer connect objects to

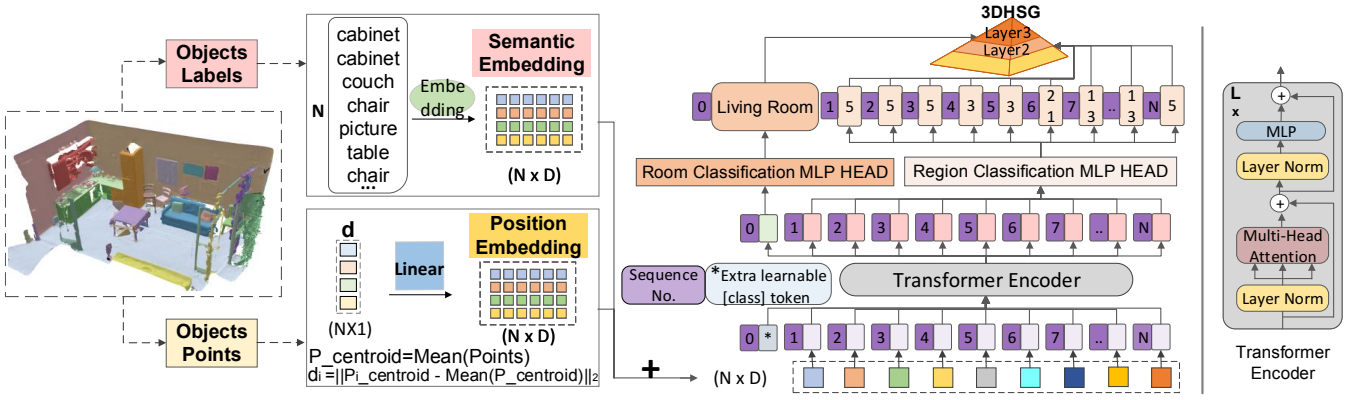

Figure 2: **TB-HSU Model Overview:** The model automatically constructs the 3DHSG for a room by completing room and region classifications, with pairs of instance-segmented point cloud and object semantic labels as inputs. The semantic embedding is derived from object labels, while position embedding is derived from object points.

regions and regions to rooms. For any nodes $u, n \in \mathcal{N}_i$, the children of u and n, denoted as $\mathcal{C}(n)$ and $\mathcal{C}(u)$, are disjoint, sharing no nodes or edges. This ensures that objects in one region are not connected to objects in another region and that regions in one room do not share edges with regions in other rooms. Details of nodes in each layer are described below:

Layer One: Objects. Nodes in the object layer, $\mathcal{N}_1$, describe objects, where each node stands for an object, with an object ID, semantic label, attributes, its segments ID, and a 2D affordance vector describing both region-specific and object-specific affordances along with a list of common room categories where the object can be found.

Layer Two: Regions. Nodes in the region layer, $\mathcal{N}_2$, describe spatial regions corresponding to a region-specific affordance. Nodes contain objects with the same region-specific affordance, a region ID, contained object IDs, and a region centroid. The region centroid is calculated as the centroid of the children's object centroids.

Layer Three: Rooms. Nodes in the room layer, $\mathcal{N}_3$, describe rooms where each node represents a room, with a room ID, scan ID, contained region IDs, and a room type.

In summary, our dataset provides a three-layered 3DHSG that describes the spatial organization of a scene with room regions defined by region-specific affordances and objects with object-specific affordances. There are 120 scene graphs with 3967 nodes and 3847 edges.

The TB-HSU Model

We refer to the network model that learns the 3DHSG as the TB-HSU model, and it is depicted in Figure 2. A multi-task learning framework is used to perform the combined task of room classification and specifying local spatial regions with region-specific affordances. Our model employs a transformer encoder architecture (Vaswani et al. 2017) that draws inspiration from the ViT (Dosovitskiy et al. 2021) and organizes the 3D scene data into a 1D sequence of token embeddings capturing both the object semantic labels and object segmented point cloud data. The 3D scene input data for the model is a $N \times D$ tensor, where N represents the number of objects in the room (zero-padded if necessary) and D is the size of the token embedding.

The details for computing the token embedding for an object are now described. The object semantic labels are represented as one-hot vectors and mapped to a semantic embedding, $\mathbf{E}_{\text{sem}} \in \mathbb{R}^{N \times D}$, using either a standard Embedding layer (or CLIP's text embedding (Radford et al. 2021) in Appendices). With regard to the segmented point cloud data for each object, we calculate the centroid of the object as the mean position across all points. We then calculate a centroid for the room as the mean across all objects. We then calculate the distance between the object centroids and the centroid of the room as d_i, where i is an index representing the i-th object. The distance data are then mapped to a position embedding, $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{N \times D}$, with a linear projection. The token embedding is then derived by adding together the semantic embedding and the position embedding. We also prepend a learnable embedding, $\mathbf{x}_{\text{class}} \in \mathbb{R}^{1 \times D}$, to the token embedding in order to learn the room classification. This procedure is similar to methods used in ViT's (Dosovitskiy et al. 2021) and BERT's (Devlin et al. 2019). More specifically, we have as input to our transformer, $\mathbf{z}_0$, which is defined as:

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}, \mathbf{E}_{\text{sem}} + \mathbf{E}_{\text{pos}}]. \tag{1}$$

We follow the structure of the transformer encoder in (Radford et al. 2021), which consists of alternating layers of multiheaded self-attention (MSA) blocks and MLP blocks with a Layernorm (LN) attached before every block and a residual connection after every block. The MLP used here contains two layers with a QuickGELU non-linearity and dropout functions. The transformer network structure is shown below:

$$\begin{aligned} \mathbf{z}'_l &= \text{MSA}(\text{LN}(\mathbf{z}_{l-1})) + \mathbf{z}_{l-1} \\ \mathbf{z}_l &= \text{MLP}(\text{LN}(\mathbf{z}'_l)) + \mathbf{z}'_l \,, \end{aligned}$$

where layers of the transformer are indexed by the subscript l. The output of the transformer, $\mathbf{y}$, is computed as:

$$\mathbf{y} = \text{LN}(\mathbf{z}_L) \,, \tag{2}$$

where L is the last layer of the transformer network. Note

that the output $\mathbf{y}$ contains both the room and region classifications:

$$\mathbf{y} = [\mathbf{y}_{\text{room}}, \mathbf{y}_{\text{region}}]\,. \tag{3}$$

Consistent with the multi-learning framework, the loss function, $\mathcal{L}(\mathbf{y})$, for our network combines the loss associated with the room classification, $\mathcal{L}(\mathbf{y}_{\text{room}})$, and the loss associated with the region classification, $\mathcal{L}(\mathbf{y}_{\text{region}})$, as follows:

$$\mathcal{L}(\mathbf{y}) = \lambda\,\mathcal{L}(\mathbf{y}_{\text{room}}) + (1-\lambda)\mathcal{L}(\mathbf{y}_{\text{region}})\,, \tag{4}$$

where $\lambda = \frac{\mathcal{L}(\mathbf{y}_{\text{room}})}{\mathcal{L}(\mathbf{y}_{\text{room}})+\mathcal{L}(\mathbf{y}_{\text{region}})}$.

Experiments

We evaluate our TB-HSU model by comparing it with two non-neural network models, three baseline neural network models, and some published models. We run the comparisons using three different datasets.

Datasets

We employ the three datasets in the evaluation experiments.

3DHSG Dataset We split our custom 3DHSG dataset into 96 scenes (80%) for training and 24 scenes (20%) for testing. We exclude object labels such as "wall", "floor", and "ceiling" because these regions are not annotated.

ScanNet (Dai et al. 2017) ScanNet is an RGB-D video dataset containing 1513 scans annotated with instance-level semantic segmentations. We split it into 1013 scenes for training and 500 scenes for testing, the same as (Huang, Usvyatsov, and Schindler 2020). There are 21 different room types in the dataset. This dataset comes in two varieties: ScanNet20 with 20 object semantic labels and ScanNet200 with 200 object semantic labels.

Matterport3D (Chang et al. 2017) Matterport3D is an RGB-D dataset consisting of 90 reconstructions of indoor building-scale scenes with 2194 rooms. There are 30 room types in the dataset. We split the dataset in the same way as the benchmark and discarded rooms that contain less than 3 objects, the same as (Hughes, Chang, and Carlone 2022).

The Comparison Models

Non-Neural Network Baseline Models We use a Random Forest (RF) classifier to perform room classification based on the object semantic labels. For the room region classification, we were inspired by two different methods: a *Term Frequency-Inverse Document Frequency (TF-IDF)* approach as described in (Heikel and Espinosa-Leal 2022); and a *Neighbor-Vote* method (Chu et al. 2021).

TF-IDF approach: We consider each region-specific affordance as a term; the collection of different region-specific affordances for an object as a document, and all region-specific affordances for all objects collectively as the entire set of documents. We calculate for each object a set of TF-IDF scores representing the probability that the object belongs to a particular region-specific affordance. The region-specific affordance with the highest TF-IDF score is chosen as the object's region-specific affordance.

Neighbor-Vote method: We consider the TF-IDF scores for the object and its neighboring objects when predicting the object's region-specific affordance. The close neighbors of an object are identified as having bounding boxes overlapping with the object, where the bounding boxes are determined from the object point clouds. We then calculate an object's TF-IDF score as $\alpha = 0.8$ times the TF-IDF score for the object plus $(1-\alpha) = 0.2$ times the mean TF-IDF score for the close neighboring objects. As previously, the region-specific affordance with the highest score is chosen as the object's region-specific affordance.

Neural Network Baseline Models For the three baseline neural network models, we use the same semantic embedding and position embedding as our TB-HSU model.

MLP Model: In each layer block, we utilize two linear layers separated by a QuickGELU activation, a layer normalization, and a dropout. The number of layers matches that of the transformer layer.

CNN Model: Six 1D convolution layers are applied along the object dimension. Batch normalization and ReLU activation are used between each convolution.

Custom ResNet Model: We configured a ResNet (He et al. 2016) with three 1D convolutional layers, each followed by batch normalization and ReLU activation. This is followed by five residual blocks composed of two 1D convolutional layers with batch normalization and ReLU activation.

Note that all network models are trained with the SGD optimizer, with a base learning rate of 1×10^{-3}, except for the TB-HSU model trained on ScanNet20, which uses a base learning rate of 1×10^{-4}, on a single NVIDIA GeForce GTX 3070 within 500 training epochs, except for the TB-HSU model trained on Matterport3D within 30 epochs.

Published Reference Models We found two published reference models for the room classification task. The second version of the Hydra (Hughes et al. 2024) model utilizes pre-trained word2vec (Mikolov et al. 2013) vectors to represent object semantic labels and concatenates them with the geometric feature vectors to perform room classification. It has been tested on the Matterport3D (Chang et al. 2017) dataset. A published point class histogram model (PCH) (Huang, Usvyatsov, and Schindler 2020) has been tested on the ScanNet20 dataset. We also explore the room classification and region classification performance of GPT-4o, released by OpenAI in May 2024. According to OpenAI's official introduction, GPT-4o is the most powerful online chatbot to date. We prompt GPT-4o with figures of the scene and object semantic labels along with object centroid positions. The prompts are provided in the Appendices.

The Performance Metrics

We evaluate the performance of the models on the room and region classification tasks by reporting accuracy (Acc) and mean of intersection-over-union metric (mIoU), where IoU is $\frac{\text{TP}}{\text{TP+FP+FN}}$ and mIoU is average of IoU across all classes.

Results

We describe the model performance results for the room classification and region classification tasks. For the room classification task, we have the following datasets available: 3DHSG, Matterport3D, ScanNet20, and Scannet200.

	Methods	T1 Acc%	T1 mIoU%	T2 Acc%	T2 mIoU%
Non-NN	RF+TF-IDF	83.3	58.3	62.26	50.38
	RF+Neighbor-Vote	83.3	58.3	62.62	50.48
NN	MLP	29.17	3.24	44.56 ± 0.59	25.74 ± 0.34
	CNN	86.11	68.58	83.91 ± 1.14	72.35 ± 1.09
	ResNet	87.50	72.02	85.02 ± 0.95	73.24 ± 0.99
Proposed TB-HSU	w/o P. E.	90.28	73.74	84.95 ± 0.70	74.87 ± 1.23
	with P.E.	**91.67**	**74.60**	**87.27** ± 0.98	**78.55** ± 2.29
	GPT-4o	**91.67**	**74.60**	44.83	33.44

Table 1: Room and Region classification results for the 3DHSG dataset. T1 refers to the room classification task – layer three of the 3DHSG; T2 refers to the region classification task – layer two of the 3DHSG.

For the region classification task, we have only our custom 3DHSG dataset available. The TB-HSU model employs 4 transformer layers with 384 dimensions across all experiments, adapting the room classification head size (12 for 3DHSG, 30 for Matterport3D, 21 for ScanNet20 and ScanNet200), the number of kinds of object labels (191 for 3DHSG, 41 for Matterport3D, 20 for ScanNet20, and 200 for ScanNet200), and input sequence length (77 for 3DHSG, 230 for Matterport3D, 62 for ScanNet20, and 121 for ScanNet200), maintaining 7.62 ± 0.05 million parameters.

Room Classification

Consider now the room classification task. Table 2 shows results for the Matterport3D, ScanNet20, and ScanNet200 datasets. For the Matterport3D dataset, the proposed TB-HSU model obtains a performance accuracy of 62.19% compared to the published result of 57.67% for the Hydra model (Hughes, Chang, and Carlone 2022). For the ScanNet20 dataset, the proposed TB-HSU model performs with an accuracy of 86% or 86.8% with or without the position embedding and better than the published result for the PCH baseline model (Huang, Usvyatsov, and Schindler 2020), which obtains an accuracy of 85%. For the ScanNet200 dataset, compared to ScanNet20, we find that the expanded list of object semantic labels positively impacts the TB-HSU model's performance with an increase in accuracy of 2.8%.

Dataset	Models	Inputs	Acc%
Matterport3D	Hydra	S+P	57.67 ± 0.57
	Proposed TB-HSU	S+P	**62.19** ± 0.35
ScanNet20	PCH	S*	82.8
		S	85.0
	Proposed TB-HSU	S	86.0
		S+P	**86.8**
ScanNet200	Proposed TB-HSU	S	88.6
		S+P	**89.6**

Table 2: Room Classification Accuracy Results. S refers to object semantic labels and P refers to object position data. S* refers to non-ground truth object labels. The Hydra model description can be found in (Hughes et al. 2024). The PCH description can be found in (Huang, Usvyatsov, and Schindler 2020).

Room and Region Classification

In Table 1, we compare model performances on the room and region classification task using our custom 3DHSG dataset. Let us consider the room classification (Task one) first. We see that the proposed TB-HSU transformer model performs better than the non-neural network models and the three baseline neural network models, obtaining a performance accuracy of 91.67%. Note that the position encoding does not seem to strongly influence the room classification performance. This is reasonable because the room type is much more strongly dependent on the type of objects within the room rather than their position within it. Note that the GPT-4o model performs similarly to the TB-HSU model. This stands to reason because the GPT-4o model was used with manual supervision to obtain the room labels as ground truth for the 3DHSG dataset. With regards to the region classification (Task two), the proposed TB-HSU model is again the best-performing model with an accuracy of 87.27% when using the position embedding, although the CNN and ResNet models with position embedding perform similarly when TB-HSU uses no position embedding. It is interesting to observe that the position embedding does seem to contribute some useful information for the region classification task. We note that the GPT-4o model does not perform well on the region classification task, indicating that it lacks the ability to understand the spatial concept of the object. Additional support can be found in Section Discussion.

Discussion

The capability of large language models (LLMs) is increasing at a rapid rate. For example, previous methods such as ConceptFusion (Jatavallabhula et al. 2023), SayCan (Ahn et al. 2022), and 3D-LLM (Hong et al. 2023) have demonstrated the viability of large language models as knowledge bases that can be queried for generating task-level plans. Along these lines, we suggest that the TB-HSU model and its resulting 3DHSG may be useful as a prompt input to LLMs. Consider the interesting case of finding an object that is not visible in the scene. Figure 3 illustrates a scenario from the 3RScan (Wald et al. 2019) interactive household simulator. The system is tasked with finding an object of interest (shampoo) that is not contained within the map because it is concealed within a receptacle or too small to detect. We investigate the behavior of GPT-4o. As seen in

3DHSG Assist GPT-4o	Non-Assisted GPT-4o With Objects Annotation	Non-Assisted GPT-4o Without Objects Annotation
User: I am in this room, where instance segmented point cloud is shown in ***Fig(a)***. The Object ID and Label are displayed in ***Fig(b)*** with with their labels and respective positions:[4shower curtain [-1.1729037 0.59439874 -0.54456013]]...32shelf [1.4467084 0.9029187 -0.84502167]]. ***Fig(c)*** presents a three-layer hierarchical scene graph. The nodes in this graph contain room layer with room type, region layer with region-specific affordances and object layer with object-specific affordances. I want to get a shampoo. Rank Object IDs by the likelihood of where I should go? Succeed! **Assistant:** Given the hierarchical scene graph and the relative locations of Object IDs with their corresponding labels and affordances, you can infer where to find a shampoo based on the typical placement of such items in a bathroom. Here is a ranked list of the regions and objects where you might likely find a shampoo: 1.**Region for Shower** - **Object:** 4 - Shower curtain - **Object:** 5 - Shower -**Object:** 16 - Rack 2. **Region for Washing** - **Object:** 8 - Sink ### Explanation: - **Region for Shower**:- **4 - Shower curtain** and **5 - Shower**: Shampoo is commonly placed within the shower area, either on racks attached to the shower or on the bathtub rim.- **16 - Rack**: Racks, particularly in the shower area, are often used to hold toiletries such as shampoo. - **Region for Washing**:- **8 - Sink**: The sink area may also have toiletries, though it is less common compared to the shower area.	**User:** I am in this room, where instance segmented point cloud is shown in ***Fig(a)***. The Object ID and Label are displayed in ***Fig(b)*** with with their labels and respective positions:[4shower curtain [-1.1729037 0.59439874 -0.54456013]]...32shelf [1.4467084 0.9029187 -0.84502167]]. I want to get a shampoo. Rank Object IDs by likelihood where I should go? FAIL! **Assistant:** To rank the likelihood of where you might find a shampoo in the room, consider typical locations where a shampoo would be stored or placed. Here's my ranking based on probability and common use: 1. **24dispenser**: This is the most probable location as it could be a dispenser specifically for shampoo and liquid products. 2. **13bin**: Although it says "bin," which usually denotes a trash bin, in some contexts, it could be a storage bin for products like shampoo. 3. **32shelf**: Shelves are commonly used to store toiletry items including shampoo.	**User:** I am in this room, where instance segmented point cloud is shown in ***Fig(a)***. I want to get a shampoo. Rank Object IDs by likelihood where I should go? FAIL! **Assistant:** To find the shampoo in a room based on an instance segmented point cloud, you generally have to think about common locations or objects associated with shampoo storage. Here are some likely candidates ranked from highest to lowest likelihood: 1. **Object ID 6 (Purple-colored item, likely a sink or cabinet)**- Sinks or cabinets around bathroom areas often store personal hygiene items, including shampoo. 2. **Object ID 5 (Brown-colored item, a cabinet or shelf)**- Cabinets or shelves are common places to store shampoo bottles. 3. **Object ID 11 (Pink-colored area, likely a bathtub or shower area)**- Shampoo is likely to be found near bathing areas for easy access during use.

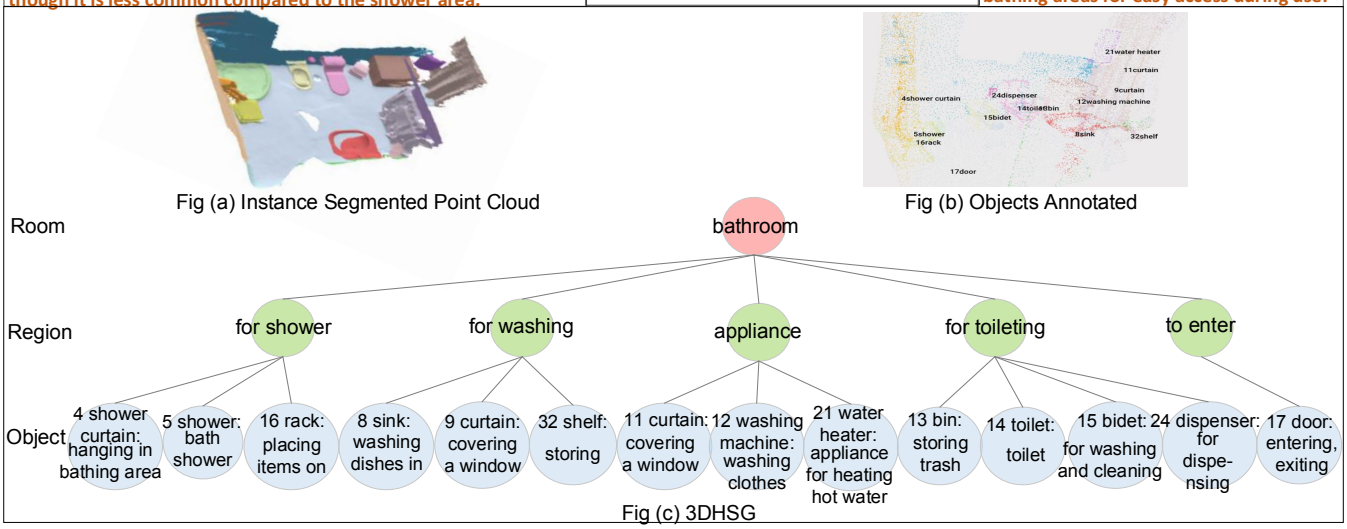

Figure 3: 3DHSG from TB-HSU assist GPT-4o in a Question-Answering task to find an object not visible within the scene. *Fig(a), Fig(b), Fig(c)* are inserted appropriately place within the prompts.

Figure 3, GPT-4o, when provided only with an image from an instance-segmented point cloud, struggles to identify objects in the scene. However, with labeled point clouds and object semantic labels, objects can be clearly identified but there is still difficulty in suggesting where to find the missing object. When we include the 3DHSG that results from the TB-HSU model, we find that the outputs from the GPT-4o model are much more reasonable.

Conclusion

In conclusion, this study on Hierarchical 3D Scene Understanding with Contextual Affordances demonstrates that the affordances of objects vary with different levels of spatial context. We introduce the 3DHSG dataset annotated with region-specific and object-specific affordances and organized into three spatial layers: Objects, Regions, and Rooms. We propose the TB-HSU model for solving the multi-task problem of room classification and region classification, with a promising performance over multiple baselines. The TB-HSU model produces a 3D hierarchical scene graph that is useful for evaluating task objectives based on a spatial and functional framework that allows affordances to vary with the spatial context. Additionally, the spatial organization of the 3DHSG dataset enhances the performance of large language models (LLMs) in question-answering tasks. In future work, we will improve our dataset and model, e.g., reducing the variety of objects and distinguishing similar regions.

Acknowledgments

We would like to thank the Australian government for their funding support via a CRC Projects Round 11 grant.

References

Ahn, M.; Brohan, A.; Brown, N.; Chebotar, Y.; Cortes, O.; David, B.; Finn, C.; Fu, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Ho, D.; Hsu, J.; Ibarz, J.; Ichter, B.; Irpan, A.; Jang, E.; Ruano, R. J.; Jeffrey, K.; Jesmonth, S.; Joshi, N. J.; Julian, R.; Kalashnikov, D.; Kuang, Y.; Lee, K.-H.; Levine, S.; Lu, Y.; Luu, L.; Parada, C.; Pastor, P.; Quiambao, J.; Rao, K.; Rettinghouse, J.; Reyes, D.; Sermanet, P.; Sievers, N.; Tan, C.; Toshev, A.; Vanhoucke, V.; Xia, F.; Xiao, T.; Xu, P.; Xu, S.; Yan, M.; and Zeng, A. 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. arXiv:2204.01691.

Armeni, I.; He, Z.-Y.; Zamir, A.; Gwak, J.; Malik, J.; Fischer, M.; and Savarese, S. 2019. 3D Scene Graph: A Structure for Unified Semantics, 3D Space, and Camera. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 5663–5672. Seoul, Korea (South): IEEE.

Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Niebner, M.; Savva, M.; Song, S.; Zeng, A.; and Zhang, Y. 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *2017 International Conference on 3D Vision (3DV)*, 667–676. Qingdao: IEEE.

Chu, X.; Deng, J.; Li, Y.; Yuan, Z.; Zhang, Y.; Ji, J.; and Zhang, Y. 2021. Neighbor-Vote: Improving Monocular 3D Object Detection through Neighbor Distance Voting. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, 5239–5247. New York, NY, USA: Association for Computing Machinery.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Niessner, M. 2017. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2432–2443. Honolulu, HI: IEEE.

Deng, S.; Xu, X.; Wu, C.; Chen, K.; and Jia, K. 2021. 3D AffordanceNet: A Benchmark for Visual Object Affordance Understanding. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1778–1787. Nashville, TN, USA: IEEE.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Gu, Q.; Kuwajerwala, A.; Morin, S.; Jatavallabhula, K. M.; Sen, B.; Agarwal, A.; Rivera, C.; Paul, W.; Ellis, K.; Chellappa, R.; Gan, C.; de Melo, C. M.; Tenenbaum, J. B.; Torralba, A.; Shkurti, F.; and Paull, L. 2024. ConceptGraphs: Open-Vocabulary 3D Scene Graphs for Perception and Planning. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, 5021–5028. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 770–778. IEEE Computer Society.

Heikel, E.; and Espinosa-Leal, L. 2022. Indoor Scene Recognition via Object Detection and TF-IDF. *Journal of Imaging*, 8(8): 209.

Hong, Y.; Zhen, H.; Chen, P.; Zheng, S.; Du, Y.; Chen, Z.; and Gan, C. 2023. 3D-LLM: Injecting the 3D World into Large Language Models. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Huang, S.; Usvyatsov, M.; and Schindler, K. 2020. Indoor Scene Recognition in 3D. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8041–8048. Las Vegas, NV, USA: IEEE.

Hughes, N.; Chang, Y.; and Carlone, L. 2022. 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.

Hughes, N.; Chang, Y.; Hu, S.; Talak, R.; Abdulhai, R.; Strader, J.; and Carlone, L. 2024. Foundations of spatial perception for robotics: Hierarchical representations and real-time systems. *Int. J. Robotics Res.*, 43(10): 1457–1505.

Jatavallabhula, K. M.; Kuwajerwala, A.; Gu, Q.; Omama, M.; Iyer, G.; Saryazdi, S.; Chen, T.; Maalouf, A.; Li, S.; Keetha, N. V.; Tewari, A.; Tenenbaum, J. B.; de Melo, C. M.; Krishna, K. M.; Paull, L.; Shkurti, F.; and Torralba, A. 2023. ConceptFusion: Open-set multimodal 3D mapping. In Bekris, K. E.; Hauser, K.; Herbert, S. L.; and Yu, J., eds., *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*.

Kim, U.-H.; Park, J.-M.; Song, T.-j.; and Kim, J.-H. 2020. 3-D Scene Graph: A Sparse and Semantic Representation of Physical Environments for Intelligent Agents. *IEEE Transactions on Cybernetics*, 50(12): 4921–4933.

Kolve, E.; Mottaghi, R.; Gordon, D.; Zhu, Y.; Gupta, A.; and Farhadi, A. 2017. AI2-THOR: An Interactive 3D Environment for Visual AI. *CoRR*, abs/1712.05474.

Maggio, D.; Chang, Y.; Hughes, N.; Trang, M.; Griffith, J. D.; Dougherty, C.; Cristofalo, E.; Schmid, L.; and Carlone, L. 2024. Clio: Real-Time Task-Driven Open-Set 3D

Scene Graphs. *IEEE Robotics Autom. Lett.*, 9(10): 8921–8928.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. In Bengio, Y.; and LeCun, Y., eds., *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

OpenAI. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Peng, S.; Genova, K.; Jiang, C.; Tagliasacchi, A.; Pollefeys, M.; and Funkhouser, T. 2023. OpenScene: 3D Scene Understanding with Open Vocabularies. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–824. Vancouver, BC, Canada: IEEE.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.

Rosinol, A.; Gupta, A.; Abate, M.; Shi, J.; and Carlone, L. 2020. 3D Dynamic Scene Graphs: Actionable Spatial Perception with Places, Objects, and Humans. In *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation.

Rosinol, A.; Violette, A.; Abate, M.; Hughes, N.; Chang, Y.; Shi, J.; Gupta, A.; and Carlone, L. 2021. Kimera: From SLAM to spatial perception with 3D dynamic scene graphs. *The International Journal of Robotics Research*, 40(12-14): 1510–1546.

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In Singh, S.; and Markovitch, S., eds., *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 4444–4451. AAAI Press.

Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C. Y.; Verma, S.; Clarkson, A.; Yan, M.; Budge, B.; Yan, Y.; Pan, X.; Yon, J.; Zou, Y.; Leon, K.; Carter, N.; Briales, J.; Gillingham, T.; Mueggler, E.; Pesqueira, L.; Savva, M.; Batra, D.; Strasdat, H. M.; De Nardi, R.; Goesele, M.; Lovegrove, S.; and Newcombe, R. A. 2019. The Replica Dataset: A Digital Replica of Indoor Spaces. *CoRR*, abs/1906.05797.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR*, abs/2302.13971.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Wald, J.; Avetisyan, A.; Navab, N.; Tombari, F.; and Nießner, M. 2019. RIO: 3D Object Instance Re-Localization in Changing Indoor Environments. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 7657–7666. IEEE.

Wald, J.; Dhamo, H.; Navab, N.; and Tombari, F. 2020. Learning 3D Semantic Scene Graphs From 3D Indoor Reconstructions. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3960–3969. Seattle, WA, USA: IEEE.

Wu, S.; Wald, J.; Tateno, K.; Navab, N.; and Tombari, F. 2021. SceneGraphFusion: Incremental 3D Scene Graph Prediction From RGB-D Sequences. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 7515–7525. Computer Vision Foundation / IEEE.