Learning 3D Semantic Scene Graphs from 3D Indoor Reconstructions

Johanna Wald [1, *] Helisa Dhamo [1, *] Nassir Navab [1] Federico Tombari [1,2]
[1] Technische Universität München [2] Google

Abstract

Scene understanding has been of high interest in computer vision. It encompasses not only identifying objects in a scene, but also their relationships within the given context. With this goal, a recent line of works tackles 3D semantic segmentation and scene layout prediction. In our work we focus on scene graphs, a data structure that organizes the entities of a scene in a graph, where objects are nodes and their relationships modeled as edges. We leverage inference on scene graphs as a way to carry out 3D scene understanding, mapping objects and their relationships. In particular, we propose a learned method that regresses a scene graph from the point cloud of a scene. Our novel architecture is based on PointNet and Graph Convolutional Networks (GCN). In addition, we introduce 3DSSG, a semi-automatically generated dataset, that contains semantically rich scene graphs of 3D scenes. We show the application of our method in a domain-agnostic retrieval task, where graphs serve as an intermediate representation for 3D-3D and 2D-3D matching.

1. Introduction

3D scene understanding relates to the perception and interpretation of a scene from 3D data, with a focus on its semantic and geometric nature, which includes not only recognizing and localizing the objects present in the 3D space therein, but also their context and relationships. This thorough understanding is of high interest for various applications such as robotic navigation, augmented and virtual reality. Current 3D scene understanding works include perception tasks such as instance segmentation [12, 21, 44, 50], semantic segmentation [34, 36, 5, 38] as well as 3D object detection and classification [40, 34, 35, 54]. While these works mostly focus on object semantics, their context and relationships are primarily used to improve the per-object class accuracy.

Scene understanding from images has recently explored the use of scene graphs to aid understanding object relationships in addition to characterizing objects individually. Before that, scene graphs have been used in computer graphics to arrange spatial representations of a graphical scene, where nodes commonly represent scene entities (object instances), while the edges represent relative transformations between two nodes. This is a flexible representation of a scene which encompasses also complex spatial relations and operation grouping. Some of these concepts where successively adapted or extended in computer vision datasets, such as support structures [32], semantic relationships and attributes [19] and hierarchical mapping of scene entities [3]. Scene graphs have been shown to be relevant, for instance, for partial [46] and full matching [17] in image search, as well as image generation [16].

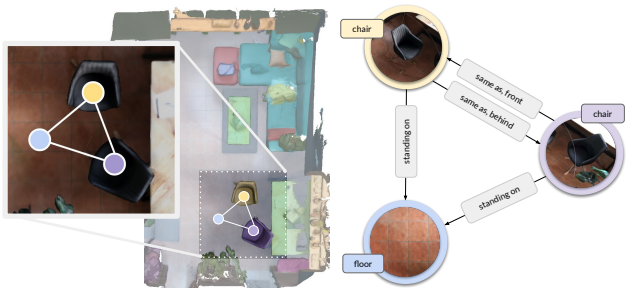

Figure 1. **Overview** Given a class-agnostic instance segmentation of a 3D scene (left) our graph prediction network infers a semantic scene graph $\mathcal{G}$ (right) from a point cloud.

In 3D, scene graphs have only recently gained more popularity [3]. In this work, we want to focus on the *semantic* aspects of 3D scene graphs as well as their potential. Our goal is to obtain dense graphs with labeled instances (nodes), semantically meaningful relationships (edges) such as `lying on` or `same as` and attributes including *color, shape or affordances* (see Fig. 1). These resemble the scene graph representation of [17], associated with images. We believe semantic scene graphs are especially important in 3D since a) they are a compact representation, that describes a (potentially large) 3D scene, b) they are robust towards small scene changes and noise and c) they close the gap between different domains, such as text or images. These properties make them suitable for cross domain tasks such as 2D-3D Scene Retrieval or VQA.

*the authors contributed equally to this paper

We believe that the capability of regressing the scene graph of a given 3D scene can be a fundamental piece for 3D scene understanding, as a way to learn and represent object relationships and contextual information of an environment. For this purpose, we propose a learned method, based on PointNet [34] and Graph Convolutional Networks (GCNs) [18], to predict 3D semantic graphs. Given a class-agnostic instance segmentation of a 3D point cloud, we jointly infer a 3D scene graph composed of nodes (scene components) and edges (their relationships). For this purpose, we introduce a 3D semantic scene graph dataset that features detailed semantics in the nodes (instances) including attributes and edges (relationships), which will be publicly released[1]. Generating 3D semantic scene graphs from real-world scans is particularly challenging due to missing data and clutter and the complexity of the relationships between objects. For instance, two chairs that are of the `same style` could have very different appearances, while a jacket `lying on` one of them might occlude most of its visible surface. While our method outperforms the baseline, it operates end-to-end and is able to predict multiple relationships per edges. We further show how – in a cross-domain scenario – scene graphs serve as a common encoding between 3D and 2D in a scene retrieval task in changing conditions. Given a single image the task is to find the matching 3D model from a pool of scans. Scene graphs suit particularly well because they are inherently robust towards dynamic environments, which manifest illumination changes and (non-)rigid changes introduced by human activity. In summary, we explore the prediction and application of semantic scene graphs in 3D indoor environments with the following contributions:

- We present 3DSSG, a large scale 3D dataset that extends 3RScan [45] with semantic scene graph annotations, containing relationships, attributes and class hierarchies. Interestingly, 2D scene graphs can be obtained by rendering the 3D graphs, which results in 363k graph-image pairs.
- We propose the first learned method that generates a semantic scene graph from a 3D point cloud.
- We show how 3D semantic scene graphs can be used in cross-domain retrieval, specifically 2D-3D scene retrieval of changing indoor environments.

2. Related Work

Semantic Scene Graphs with Images. Johnson *et al*. [17] introduced scene graphs – motivated by image retrieval – as a representation that semantically describes an image, where each node is an object while edges represent interactions between them. Additionally, the object nodes contain attributes that describe object properties. Later, Visual Genome [19], a large scale dataset with scene graph annotations on images, gave rise to a line of deep learning based advances on scene graph prediction from images [48, 11, 37, 51, 25, 49, 24, 33]. These methods propose diverse strategies for graph estimation and processing, such as message passing [48], graph convolutional networks (GCN) [49], permutation invariant architectures [11] and attention mechanisms [37]. Most of these methods rely on an object detector to extract node- and edge-specific features prior to the graph computation [48, 49, 24]. Recent works explore the reverse problem of using scene graphs to generate new images [16] or manipulate existing images [6].

3D Understanding: From Objects to Relationships. An active research area within 3D scene understanding focuses on 3D semantic segmentation [34, 4, 36, 7, 38, 12] and object detection and classification [41, 34, 35, 52]. These works mostly focus on object semantics and context is only used to improve object class accuracy. Holistic scene understanding [40] on the other side predicts not only object semantics, but also the scene layout and sometimes even the camera pose [13]. Scene context is often represented through a hierarchical tree, where the leaves are typically objects and the intermediate nodes group the objects in scene components or functional entities. A line of works use probabilistic grammar to parse scenes [28, 53] or control scene synthesis [15]. Shi *et al*. [39] show that the object detection task benefits from joint prediction of the hierarchical context. GRAINS [23] explore hierarchical graphs to synthesize diverse 3D scenes, using a recursive VAE that generates a layout, followed by object retrieval. In a 3D from single image scenario, Kulkarni *et al*. [20] consider relative 3D poses between objects (as edges), which are shown to outperform neighbour-agnostic 6D pose estimation.

Another line of works incorporate graphs structures for object-level understanding, rather than entire scenes. Te *et al*. [43] use a Graph CNN for semantic segmentation of object parts. StructureNet [31] represent the latent space of an object as a hierarchical graph of composing parts, with the goal of generating plausible shapes. However, all of these works are either focused on object parts, or do not consider semantic relationships that go beyond generic edges (without semantic labels) or relative transformations. In the context of semantic scene graphs on synthetic data, Fisher *et al*. [9] use graph kernels for 3D scene comparison, based on support and spatial relationships. Ma *et al*. [30] parse natural language into semantic scene graphs, considering pairwise and group relationships, to progressively retrieve sub-scenes for 3D synthesis.

Only recently the community started to explore seman-

[1] https://3DSSG.github.io

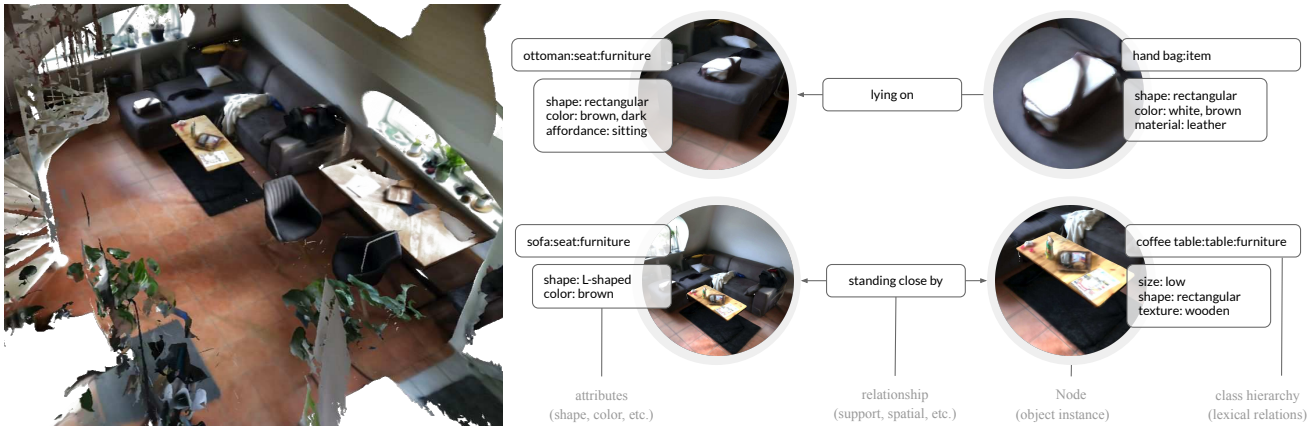

Figure 2. **Scene graph representation in 3DSSG** including hierarchical class labels c and attributes A per node, as well as relationship triplets between nodes.

tic relationships in 3D and on real world data. Armeni *et al*. [3] present a hierarchical mapping of 3D models of large spaces in four layers: camera, object, room and building. While they feature smaller graphs (see Tbl. 1) their focus is not on semantically meaningful inter-instance relationships such as support. Moreover, the absence of changing scenes, does not enable the proposed 3D scene retrieval task.

3D Scene Retrieval Many image-based 3D retrieval works focus on retrieving 3D CAD models from RGB images: IM2CAD generates a 3D scene from a single image by detecting the objects, estimating the room layout and retrieving a corresponding CAD model for each bounding box [14]. Pix3D on the other hand propose a dataset for single image 3D shape modeling based on highly accurate 3D model alignments in the 2D images [42]. Liu *et al*. show improved 2D-3D model retrieval by simulating local context to generate false occlusion [27]. The SHREC benchmark [1, 2], enables 2D-3D retrieval of diverse scenes (beach, bedroom or castle), while [30] and [9] operate on indoor environments but also only focus on synthetic data rather than real 3D reconstructions.

3. 3D Semantic Scene Graphs

With this work, we release 3DSSG which provides 3D semantic scene graphs for 3RScan [45], a large scale, real-world dataset which features 1482 3D reconstructions of 478 naturally changing indoor environments. A semantic scene graph $\mathcal{G}$ in 3DSSG, is a set of tuples $(\mathcal{N}, \mathcal{R})$ between nodes $\mathcal{N}$ and edges $\mathcal{R}$ (see Fig. 2). Nodes represent specific 3D object instances in a 3D scan. In contrast to previous works [19, 3, 4, 45], our nodes are not assigned a single object category $\mathcal{C}$ only, but instead are defined by a hierarchy of classes $c = (c_1, ..., c_d)$ where $c \in \mathcal{C}^d$, and d can vary. Additionally to these object categories each node has a set of attributes A that describe the visual and physical appearance of the object instance. A special subset of the attributes are affordances [47]. We consider them particularly important since we deal with changing environments. The edges in our graphs define semantic relationships (predicates) between the nodes such as `standing on`, `hanging on`, `more comfortable than`, `same material`. To obtain the data in 3DSSG we combine semantic annotations with geometric data and additional human verification to ensure high quality graphs. In summary, our dataset features 1482 scene graphs with 48k object nodes and 544k edges. An interesting feature of 3D scene graphs is that they can easily be *rendered* to 2D. Given a 3D model and a camera pose, one can filter the graph nodes and edges that are present in that image. Support and attribute comparison relations remain the same, while directional relationships (`left`, `right`, `behind`, `front`) must be updated automatically for the new viewpoint. Given the 363k RGB-D images with camera poses of 3RScan, this results in 363k 2D scene graphs. A comparison of our dataset with the only other real 3D semantic scene graph dataset, namely Armeni *et al*. [3] is listed in Tbl. 1. More information and statistics about 3DSSG are provided in the supplementary. In the following a detailed description of the different entities of our 3D semantic scene graphs are given.

Table 1. Semantic 3D scene graph comparision.

dataset	size	instances	classes	obj. rel.
Armeni *et al*. [3]*	35 buildings 727 rooms	3k	28	4
3DSSG (Ours)	1482 scans 478 scenes	48k	534	40

* We compare against the 3D scene graph dataset on the tiny Gibson split, the most recent release at the time of the submission

3.1. Nodes

The nodes in our graph are per definition 3D object instances, and each instance is assigned to a 3D scene. Each instance is defined by a class hierarchy where the class of order 1, c_1, in c is the corresponding annotated label. The subsequent class labels are acquired by recursively parsing the lexical definition for hypernyms of c_1 using WordNet [8]. The definition *"chair with a support on each side for arms"* gives us c_{n+1} = `chair` as a hypernym for c_n = `armchair`. Lexical ambiguities result in multiple interpretations of a class label (lexeme); therefore a selection step is required to get only a single definition per class that is most likely in an indoor setting. Given the fact that the 1.5k 3D reconstructions feature 534 different class labels, 534 lexical descriptions and their corresponding class hierarchy are provided. Fig. 3 visualizes the lexical relationships on a small subset of classes. A more complete graph can be found in the supplementary.

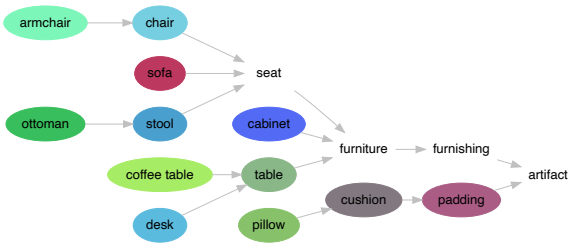

Figure 3. Simplified graphical visualization of the lexical relationships on a small subset of classes

3.2. Attributes

Attributes are semantic labels that describe object instances. This includes static and dynamic properties, as well as affordances. Due to the large number of object instances and the desired semantic diversity of the attributes, an efficient extraction and annotation design is crucial. In the following we define the different types of attributes and their acquisition.

Static Properties include visual object features such as the color, size, shape or texture but also physical properties *e.g.* the (non-)rigidity. Geometric data and class labels are utilized to identify the relative size of the object in comparison with other objects of the same category. Since some features are class specific, we assign them on the class level. An example is an automatic attribute extraction from the lexical descriptions *e.g.* a ball is `spherical`. The remaining, more complex, attributes such as the material (`wooden, metal`), shape (`rectangular, L-shaped`) or the texture (color or pattern) are instance specific and manually annotated by expert annotators with an interface that was specifically designed for this purpose. We annotate static attributes in the reference scan and copy to each rescan, since they are not subject of change.

Dynamic Properties are particularly important object attributes, which we refer to as states, such as `open / closed`, `full / empty` or `on / off`. We define a state category to be class specific, while its current condition is a matter of instance and therefore also annotated with the aforementioned interface, together with generic static properties. Since state properties of objects can change over time specific instances in the rescans are separately annotated.

Affordances Following previous works [10, 47, 3] we define affordances as interaction possibilities or object functionalities of nodes of a specific object class *e.g.* a `seat` is for `sitting`. We however condition them with their state attribute: only a `closed door` can be `opened`. This is particularly interesting since our 3D scans are from changing scenes. These changes often involve state changes caused by human interaction (see examples in the supplementary material). Overall, 3DSSG features 93 different attributes on approx. 21k object instances and 48k attributes in total.

3.3. Relationships

3DSSG has a rich set of relationships classifiable into a) spatial / proximity relationships b) support relations and c) comparative relationships.

Support Relationships Support relationships indicate the supporting structures of a scene [32]. By definition, an instance can have multiple supports; walls are by default supported by the floor and the floor is the only instance that, by definition, does not have any support. Automatically extracting support relationships is quite challenging due to the noisy and partial nature of real 3D scans. For each object in the scene, we consider neighbouring instances in a small radius (*e.g.* 5cm) support candidates. These support candidates then undergo a verification procedure to a) eliminate incorrect supports and b) complete missing candidates. Remaining class-to-class (*e.g.* `bottle-table`) support pairs are then annotated with a so called *semantic support* (*e.g.* `standing, lying`) and then specified for each instance in the dataset.

Proximity Relationships Proximity relationships describe the spatial relationships (*e.g.* `next to, in front of`) with respect to a reference view. To limit

redundancy, we only compute proximity relationships between the nodes that share a support parent. A bottle on a table therefore has no proximity relationship with a chair but the supporting table does, since the proximity relationship of the bottle can automatically be derived from its support parent.

Comparative Relationships The last group of relationships are derived from comparison of attributes, *e.g.* `bigger than`, `darker than`, `cleaner than`, `same shape as`. We use aforementioned attributes, see Section 3.2, to generate these.

4. Graph Prediction

Given the point set $\mathcal{P}$ of a scene s and the class-agnostic instance segmentation $\mathcal{M}$, the goal of the Scene Graph Prediction Network (SGPN) is to generate a graph $\mathcal{G} = (\mathcal{N}, \mathcal{R})$, describing the objects in the scene $\mathcal{N}$ as well as their relationships $\mathcal{R}$, Fig. 4. We base our learning method in a common principle in scene graph prediction [29, 48, 49], which involves extraction of visual features for every node ϕ_n and edge ϕ_r. We use two PointNet [34] architectures for the extraction of ϕ_n and ϕ_r, which we dub namely ObjPointNet and RelPointNet. For a scene s, we extract the point set of every instance i separately, masked with $\mathcal{M}$

$$\mathcal{P}_i = \{\delta_{m_k i} \odot p_k\}_{k=1,|\mathcal{P}|} \tag{1}$$

where δ represents the Kronecker delta [2], p, m are instances of $\mathcal{P}, \mathcal{M}$ and $|\cdot|$ is the cardinality of $\mathcal{P}$, *i.e.* the number of points. Each of the individual point sets $\mathcal{P}_i$ is the input to ObjPointNet .

Additionally, we extract a point set for every pair of nodes i and j, using the union of the respective 3D bounding boxes $\mathcal{B}$

$$\mathcal{P}_{ij} = \{p_k | p_k \in (\mathcal{B}^i \cup \mathcal{B}^j)\}_{k=1,|\mathcal{P}|}. \tag{2}$$

The input to RelPointNet is a point set $\mathcal{P}_{ij}$, concatenated with the respective mask $\mathcal{M}_{ij}$, which is one if the point corresponds to object i, two if the point corresponds to object j and zero otherwise. Preserving the orientation of the edge context $\mathcal{P}_{ij}$ is important to infer proximity relationships like `left` or `right`. Therefore, we disable rotational augmentation. We normalize the center of the object and edge point clouds, before feeding them to the respective networks. We arrange the extracted features in a graph structure, in the form of relationship triples (`subject, predicate, object`), where ϕ_n occupy subject / object units, while edge features ϕ_r occupy the predicate units.

We employ a Graph Convolutional Network (GCN) [18], similar to [16], to process the acquired triples. As scenes come with diverse complexities, we want the GCN to allow flexibility in the number of input nodes. Each message-passing layer l of the GCN consists of two steps. First, each triplet ij is fed in an MLP $g_1(\cdot)$ for information propagation

$$(\psi_{s,ij}^{(l)}, \phi_{p,ij}^{(l+1)}, \psi_{o,ij}^{(l)}) = g_1(\phi_{s,ij}^{(l)}, \phi_{p,ij}^{(l)}, \phi_{s,ij}^{(l)}) \tag{3}$$

where ψ represent the processed features, s indicates subject, o indicates object, and p predicate. Second, for a certain node, in an aggregation step, the signals coming from all the valid connections of that node (either as a subject or an object) are averaged together

$$\rho_i^{(l)} = \frac{1}{|\mathcal{R}_{i,s}| + |\mathcal{R}_{i,o}|} \Big(\sum_{j \in \mathcal{R}_s} \psi_{s,ij}^{(l)} + \sum_{j \in \mathcal{R}_o} \psi_{o,ji}^{(l)} \Big) \tag{4}$$

where $|\cdot|$ denotes cardinality and $\mathcal{R}_s$ and $\mathcal{R}_o$ are the set of connections of the node as subject and as objects respectively. The resulting node feature is fed in another MLP $g_2(\cdot)$. Inspired by [22], we adapt a residual connection to overcome potential Laplacian smoothing on graphs and obtain the final node feature as

$$\phi_i^{(l+1)} = \phi_i^{(l)} + g_2(\rho_i^{(l)}). \tag{5}$$

The final features $\phi_{s,ij}^{(l+1)}, \phi_{p,ij}^{(l+1)}, \phi_{o,ij}^{(l+1)}$ are then processed by the next convolutional layer l, in the same fashion. After each layer l, the node visibility is propagated to a further neighbour level. Hence, the number of layers equals the order of relations that the model can capture.

The last part of the GCN consists of two MLPs for the prediction of the node and predicate classes.

Losses We train our model end-to-end, optimizing an object classification loss $\mathcal{L}_{\text{obj}}$ as well as a predicate classification loss $\mathcal{L}_{\text{pred}}$

$$\mathcal{L}_{\text{total}} = \lambda_{obj} \mathcal{L}_{\text{obj}} + \mathcal{L}_{\text{pred}} \tag{6}$$

where λ_{obj} is a weighting factor. We assume that, realistically, for a certain object pair there are multiple valid relationships that describe their interaction. For instance, in Fig. 1, a chair can be `front of` another chair, while simultaneously having the same appearance (`same as`). Therefore, we formulate $\mathcal{L}_{\text{pred}}$ as per-class binary cross entropy. This way, it is judged independently whether an edge should be assigned a certain label (*e.g.* `standing on`) or `none`. To deal with class imbalance, for both loss terms we use a focal loss [26]

$$\mathcal{L} = -\alpha_t (1 - p_t)^\gamma \log p_t \tag{7}$$

where p_t represents the logits of a prediction and γ is a hyper-parameter. α_t is the normalized inverse frequency for the multi-class loss ($\mathcal{L}_{obj}$) and a fixed edge / no-edge factor for the per-class loss ($\mathcal{L}_{pred}$).

Implementation details are provided in the supplement.

[2] $\delta_{ij} = 1 \iff i = j$

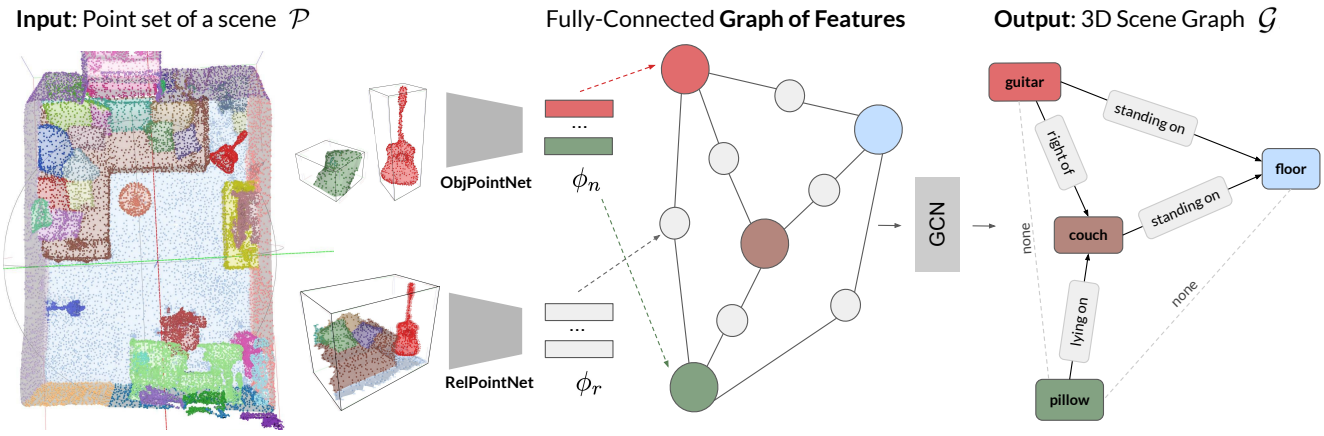

Figure 4. **Scene Graph Prediction Network** Given a point set $\mathcal{P}$ of a scene, annotated with instance segmentation $\mathcal{M}$, we infer a scene graph $\mathcal{G}$. *Left:* Visual point features ϕ are extracted per object (color-coded) and per edge. *Center:* The features ϕ are arranged in a graph structure for further processing from a GNN. *Right:* The predicted graph, consisting of labeled object nodes and directed labeled edges.

5. Scene Retrieval

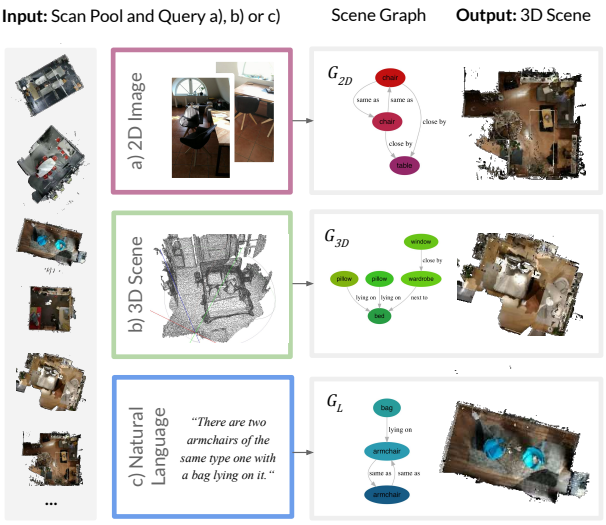

Figure 5. **Cross Domain 2D-3D Scene Retrieval:** scene graphs are used in our cross-domain scene retrieval task to close the domain gap between 2D images, 3D scenes and other modalities

We introduce a new cross-domain task named *image based 3D scene retrieval in changing indoor environments* which is about identifying the 3D scene from a list of scans given a single 2D image with potential global and local changes (see Fig. 5). This is particularly challenging since it involves a) multiple domains (2D images and 3D models) and b) scene changes (moving objects, changing illumination). For the evaluation we select semantically rich 2D images from the rescan sequences in 3RScan [45]. Due to the domain gap between 2D images and 3D we propose carrying out this novel retrieval task through scene graphs – which are by definition more stable towards scene changes – and serve as a shared domain between 2D and 3D. Such an approach also allows to retrieve 3D scenes from any input domain from which a scene graph can be generated *e.g.* natural language or 3D directly. We show how different similarity metrics can be used to successfully find the correct 3D scene using not only object semantics but also the scene context in form of semantically meaningful relationships between object instances. Computing the similarity between graphs is a NP-complete problem, so instead of matching graphs directly via their graph edit distance we first transform our scene graphs into multisets containing node classes and their (semantic) edges / tuples. Please note that these potentially have repetition of elements. To get the similarity of of two graphs, a similarity scores τ is applied on the corresponding multisets $s(\mathcal{G})$ respectively. For our tests we explore two different similarity functions: Jaccard $\tau_J(A, B)$, eq. 8 and Szymkiewicz-Simpson $\tau_S(A, B)$, eq. 9.

$$\tau_J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{8}$$

$$\tau_S(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)} \tag{9}$$

While the Jaccard coefficient is a widely used metric, the Szymkiewicz-Simpson coefficient can provide more meaningful similarity scores especially when the two sets A and B have very different sizes which is often the case in a 2D-3D scenario. When matching two graphs $\mathcal{G}$ and $\mathcal{G}'$ we combine the similarity metric of the object semantics, generic node edges $\mathcal{E}$ as well as semantic relationships $\mathcal{R}$ and obtain

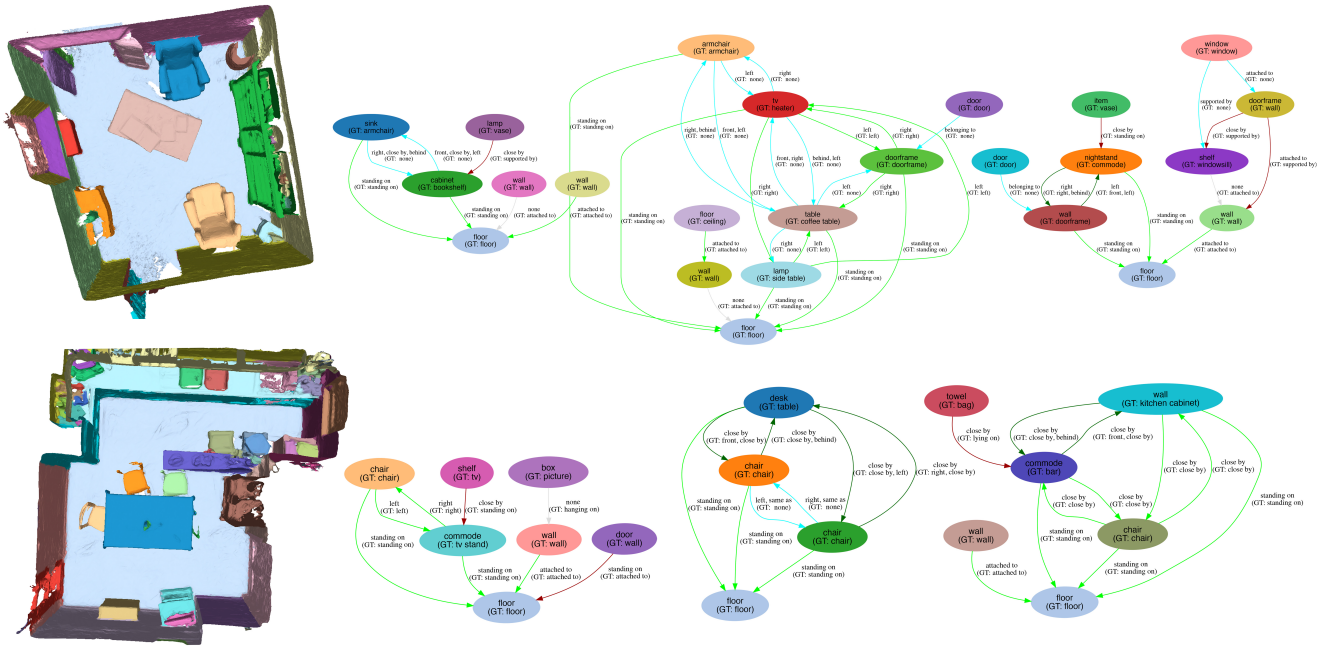

Figure 6. **Qualitative results of our scene graph prediction model** (best viewed in the digital file). *Green*: correctly predicted edges, *blue*: missing ground truth, *red*: miss-classified edges, *gray*: wrongly predicted as *none* when GT is a valid relationship.

$$f(\hat{\mathcal{G}}, \hat{\mathcal{G}}') = \frac{1}{|\hat{G}|} \sum_{i=1}^{|\hat{\mathcal{G}}|} \tau(s(\hat{\mathcal{G}}^{(i)}), s(\hat{\mathcal{G}}'^{(i)}))^3 \qquad (10)$$

where τ is either the Jaccard or Szymkiewicz-Simpson coefficient and $\hat{\mathcal{G}}$ is defined as the augmented graph $\hat{\mathcal{G}} = (\mathcal{N}, \mathcal{E}, \mathcal{R})$ where $\mathcal{E}$ are binary edges. Interestingly, one can use our retrieval method to find rooms that fullfill certain requirements such as the available of objects *e.g.* meeting room with a `TV, whiteboard` but could also include affordances: `sitting` for 20 people.

6. Evaluation

In the following, we first report results of our 3D graph prediction by comparing it against a relationship prediction baseline, inspired by [29], on our newly created 3DSSG-dataset. We re-implemented and adapted their method to work with 3D data. The baseline extracts node and edge features from an image, which we translate to PointNet features in 3D, similar to our network. The edge and node features are passed directly, namely to a predicate and object classifier. For evaluation we use the same train and test splits as originally proposed by [45]. We validate the effectiveness of our multi predicate classifier and GCN in our proposed network in an ablation study. In the second section, we evaluate different graph matching functions in 2D-3D as well as 3D-3D retrieval by matching changed scenes.

[3] we define f_S and f_J to use τ_S and τ_J respectively.

6.1. Semantic Scene Graph Prediction

Here, we report the results of our scene graph prediction task. Following previous works [48] we first separately evaluate the predicate (relationship) prediction in isolation from the object classes. The overall scene graph prediction performance is evaluated jointly where the relationship as well as the object categories are to be predicted given a set of localized objects. Since our method predicts the relationship as well as the object categories independently from another, we obtain an ordered list of triplet classification scores by multiplying the respective scores [49]. Similarly to the predicate prediction, the performance of the object categories is reported. We adopt the recall metric used in [29] to evaluate most confident (`subject, predicate, object`) triplets against the ground-truth in a top-n manner. Tbl. 2 shows that we outperform the baseline in graph related metrics, while being on par in object classification. Additionally, as expected, the multiple predicate prediction model leads to a higher predicate accuracy, which we attribute to the inherent ambiguity in a single classification problem, when multiple outputs are plausible. Moreover, we compare two versions of our model, in which the object classification is performed a) directly on the PointNet features ϕ_n and b) to the output of the GCN. We observe a slight improvement in the object and predicate accuracy for the former. Fig. 6 illustrates the predicted scene graphs. In all edges and nodes we show the predictions together with the ground truth in brackets. More examples can be found in the supplement.

Table 2. Evaluation of the scene graph prediction task on 3DSSG. We present triples prediction, object classification as well as predicate prediction accuracy.

	Relationship Prediction		Object Class Prediction		Predicate Prediction	
Method	R@50	R@100	R@5	R@10	R@3	R@5
① Relation Prediction Baseline	0.39	0.45	0.66	0.77	0.62	0.88
Single Predicate, ObjCls from PointNet Features	0.37	0.43	**0.68**	**0.78**	0.42	0.58
② Multi Predicate, ObjCls from PointNet Features	**0.40**	**0.66**	**0.68**	**0.78**	**0.89**	**0.93**
Multi Predicate, ObjCls from GCN Features	0.30	0.60	0.60	0.73	0.79	0.91

6.2. Scene Retrieval

Tbl. 3 and 4 report two scene retrieval tasks. The goal is to match either a single 2D image (Tbl. 4) or a 3D rescan of an indoor scene (Tbl. 3) with the most similar instance from a pool of 3D reference scans from the validation set of 3RScan. We compute the scene graph similarity between each rescan (2D or 3D) and the target reference scans. We then order the matches by their similarity and report the top-n metric, *i.e*. the percentage of the true positive assignments, placed in the top-n matches from our algorithm. In our experiment, we either use ground truth or predictions for the query and source graphs (see Graph-column in Tbl. 3 and 4). To measure the effect of the different similarity functions, decoupled from the graph prediction accuracy, we first evaluate $\tau_J(A, B)$ and $\tau_S(A, B)$ using ground truth graphs. Since the size of image and 3D scene graphs are significantly different, using the Szymkiewicz-Simpson coefficient in 2D-3D matching leads to better results while the performance of the Jaccard coefficient is on par or better in the 3D-3D scenario. We observe that adding semantic relationships to the graph matching improves the scene retrieval. The results also confirm that our predicted graphs ② achieve higher matching accuracy compared to the baseline model ①. Note that for the purpose of this experiment, predicted 2D graphs are obtained by rendering the predicted 3D graphs as described in Section 3.

Table 3. Evaluation: 3D-3D scene retrieval of changing 3D rescans to reference 3D scans in 3RScan.

	Graph	Top-1	Top-3	Top-5
$\tau_S(s(\mathcal{N}_{3D}), s(\mathcal{N}_{3D}))$	GT	0.86	0.99	1.00
$f_S(\mathcal{G}_{3D}, \mathcal{G}_{3D})$	GT	0.96	1.00	1.00
$\tau_J(s(\mathcal{N}_{3D}), s(\mathcal{N}_{3D}))$	GT	0.89	0.95	0.95
$f_J(\mathcal{G}_{3D}, \mathcal{G}_{3D})$	GT	0.95	0.96	0.98
$\tau_J(s(\mathcal{N}_{3D}), s(\mathcal{N}_{3D}))$	①	0.15	0.40	0.45
$f_J(\mathcal{G}_{3D}, \mathcal{G}_{3D})$	①	0.29	0.50	0.59
$\tau_J(s(\mathcal{N}_{3D}), s(\mathcal{N}_{3D}))$	②	0.32	0.46	0.50
$f_J(\mathcal{G}_{3D}, \mathcal{G}_{3D})$	②	0.34	0.51	0.56

Table 4. Evaluation: 2D-3D scene retrieval of changing rescans to reference 3D scans in 3RScan.

	Graph	Top-1	Top-3	Top-5
$\tau_J(s(\mathcal{N}_{2D}), s(\mathcal{N}_{3D}))$	GT	0.49	0.75	0.84
$\tau_S(s(\mathcal{N}_{2D}), s(\mathcal{N}_{3D}))$	GT	0.98	0.99	1.00
$f_J(\mathcal{G}_{2D}, \mathcal{G}_{3D})$	GT	0.55	0.85	0.86
$f_S(\mathcal{G}_{2D}, \mathcal{G}_{3D})$	GT	1.00	1.00	1.00
$\tau_S(s(\mathcal{N}_{2D}), s(\mathcal{N}_{3D}))$	①	0.17	0.36	0.42
$f_S(\mathcal{G}_{2D}, \mathcal{G}_{3D})$	①	0.10	0.25	0.32
$\tau_S(s(\mathcal{N}_{2D}), s(\mathcal{N}_{3D}))$	②	0.17	0.36	0.41
$f_S(\mathcal{G}_{2D}, \mathcal{G}_{3D})$	②	0.13	0.38	0.42

7. Conclusion

In this work, we explore 3D semantic scene graphs. We release 3DSSG a 3D scene graph dataset with semantically rich relationships based on 3RScan [45]. We use our data to train a graph prediction network for 3D scenes that is able to estimate not only object semantics but also relationships between objects. Further, we show the usefulness of graphs in 3D scenes by applying it to a new cross-domain task called *image based 3D scene retrieval in changing indoor environments*. This shows how semantic scene graphs are useful to bridge the domain gap between 2D-3D; opening doors for new applications such as text-3D scene retrieval or VQA. We further believe that scene graphs (and their changes) could potentially help to better reason about human activities in changing indoor environments.

Acknowledgment

We would like to thank Mariia Gladkova, Alina Karimova and Premankur Banerjee for their help with the data preparation and annotations. This work was funded by the Deutsche Forschungsgemeinschaft (DFG) #381855581, the Bavarian State Ministry of Education, Science and the Arts in the framework of the Centre Digitisation Bavaria (ZD.B) and a Google AR/VR University Research Award.

References

[1] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, Yijuan Lu, Song Bai, Xiang Bai, Ngoc-Minh Bui, Minh N. Do, Trong-Le Do, Anh-Duc Duong, Xinwei He, Tu-Khiem Le, Wenhui Li, Anan Liu, Xiaolong Liu, Khac-Tuan Nguyen, Vinh-Tiep Nguyen, Weizhi Nie, Van-Tu Ninh, Yuting Su, Vinh Ton-That, Minh-Triet Tran, Shu Xiang, Heyu Zhou, Yang Zhou, and Zhichao Zhou. 2D Image-based 3D Scene Retrieval. In *Eurographics Workshop on 3D Object Retrieval*, 2018. 3

[2] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, Yijuan Lu, Tobias Schreck, Ngoc-Minh Bui, Trong-Le Do, Mike Holenderski, Dmitri Jarnikov, Khiem T. Le, Vlado Menkovski, Khac-Tuan Nguyen, Thanh-An Nguyen, Vinh-Tiep Nguyen, Tu V. Ninh, Perez Rey, Minh-Triet Tran, and Tianyang Wang. Extended 2D Scene Image-Based 3D Scene Retrieval. In Silvia Biasotti, Guillaume Lavou, and Remco Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*, 2019. 3

[3] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3D Scene Graph: A Structure for Unified Semantics, 3D Space, and Camera. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 3, 4

[4] Angela Dai, Angel Xuan Chang, Manolis Savva, Maciej Halber, Tom Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3

[5] Angela Dai and Matthias Nießner. 3DMV: Joint 3D-Multi-View Prediction for 3D Semantic Scene Segmentation. In *European Conference on Computer Vision (ECCV)*, 2018. 1

[6] Helisa Dhamo, Azade Farshad, Iro Laina, Nassir Navab, Gregory D. Hager, Federico Tombari, and Christian Rupprecht. Semantic Image Manipulation Using Scene Graphs. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[7] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. In *International Conference on Computer Vision (ICCV) Workshops*, 2017. 2

[8] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998. 4

[9] Matthew Fisher, Manolis Savva, and Pat Hanrahan. Characterizing structural relationships in scenes using graph kernels. *ACM Trans. Graph*, 2011. 2, 3

[10] James J. Gibson. *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin, 1979. 4

[11] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping Images to Scene Graphs with Permutation-Invariant Structured Prediction. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018. 2

[12] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[13] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. Cooperative Holistic Scene Understanding: Unifying 3D Object, Layout, and Camera Pose Estimation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018. 2

[14] Hamid Izadinia, Qi Shan, and Steven M Seitz. IM2CAD. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[15] Chenfanfu Jiang, Siyuan Qi, Yixin Zhu, Siyuan Huang, Jenny Lin, Lap-Fai Yu, Demetri Terzopoulos, and Song-Chun Zhu. Configurable 3D Scene Synthesis and 2D Image Rendering with Per-pixel Ground Truth Using Stochastic Grammars. *International Journal of Computer Vision (IJCV)*, 2018. 2

[16] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image Generation from Scene Graphs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 5

[17] J. Johnson, R. Krishna, M. Stark, L. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2

[18] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*, 2017. 2, 5

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalanditis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision (IJCV)*, 2017. 1, 2, 3

[20] Nilesh Kulkarni, Ishan Misra, Shubham Tulsiani, and Abhinav Gupta. 3D-RelNet: Joint Object and Relational Network for 3D Prediction. *International Conference on Computer Vision (ICCV)*, 2019. 2

[21] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R. Oswald. 3D Instance Segmentation via Multi-Task Metric Learning. In *International Conference on Computer Vision (ICCV)*, 2019. 1

[22] Guohao Li, Matthias Mller, Ali Thabet, and Bernard Ghanem. DeepGCNs: Can GCNs Go as Deep as CNNs? In *International Conference on Computer Vision (ICCV)*, 2019. 5

[23] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. GRAINS: Generative Recursive Autoencoders for Indoor Scenes. *ACM Transactions on Graphics (TOG)*, 2018. 2

[24] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable Net: An Efficient Subgraph-Based Framework for Scene Graph Generation. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[25] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene Graph Generation from Objects, Phrases and Region Captions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[26] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollr. Focal Loss for Dense Object Detection. In *International Conference on Computer Vision (ICCV)*, 2017. 5

[27] Mingming Liu, Yunfeng Zhang, Jingwu He, Jie Guo, and Yanwen Guo. Image-Based 3D Model Retrieval for Indoor Scenes by Simulating Scene Context. *International Conference on Image Processing (ICIP)*, 2018. 3

[28] Tianqiang Liu, Siddhartha Chaudhuri, Vladimir Kim, Qixing Huang, Niloy Mitra, and Thomas Funkhouser. Creating Consistent Scene Graphs Using a Probabilistic Grammar. *ACM Transactions on Graphics (TOG)*, 2014. 2

[29] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual Relationship Detection with Language Priors. In *European Conference on Computer Vision (ECCV)*, 2016. 5, 7

[30] Rui Ma, Akshay Gadi Patil, Matthew Fisher, Manyi Li, Sren Pirk, Binh-Son Hua, Sai-Kit Yeung, Xin Tong, Leonidas Guibas, and Hao Zhang. Language-Driven Synthesis of 3D Scenes from Scene Databases. In *SIGGRAPH Asia, Technical Papers*, 2018. 2, 3

[31] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. StructureNet: Hierarchical Graph Networks for 3D Shape Generation. *ACM Transactions on Graphics (TOG)*, 2019. 2

[32] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *European Conference on Computer Vision (ECCV)*, 2012. 1, 4

[33] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 2

[34] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 5

[35] Charles Ruizhongtai Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas Guibas. Volumetric and Multi-View CNNs for Object Classification on 3D Data. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2

[36] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 1, 2

[37] Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. Attentive Relational Networks for Mapping Images to Scene Graphs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[38] Dario Rethage, Johanna Wald, Jürgen Sturm, Nassir Navab, and Federico Tombari. Fully-Convolutional Point Networks for Large-Scale Point Clouds. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 2

[39] Yifei Shi, Angel Xuan Chang, , Zhelun Wu, Manolis Savva, and Kai Xu. Hierarchy Denoising Recursive Autoencoders for 3D Scene Layout Prediction. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[40] Shuran Song, Samuel Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2

[41] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view Convolutional Neural Networks for 3D Shape Recognition. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[42] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B. Tenenbaum, and William T. Freeman. Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[43] Gusi Te, Wei Hu, Amin Zheng, and Zongming Guo. RGCNN: Regularized Graph CNN for Point Cloud Segmentation. In *International Conference on Multimedia*, 2018. 2

[44] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas J. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *International Conference on Computer Vision (ICCV)*, 2019. 1

[45] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. RIO: 3D Object Instance Re-Localization in Changing Indoor Environments. In *International Conference on Computer Vision (ICCV)*, 2019. 2, 3, 6, 7, 8

[46] Miao Wang, Yu-Kun Lai, Yuan Liang, Ralph R. Martin, and Shi-Min Hu. BiggerPicture: Data-driven Image Extrapolation Using Graph Matching. *ACM Transactions on Graphics (TOG)*, 2014. 1

[47] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: Real-World Perception for Embodied Agents. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 3, 4

[48] Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. Scene Graph Generation by Iterative Message Passing. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 5, 7

[49] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for Scene Graph Generation. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 5, 7

[50] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J. Guibas. GSPN: Generative Shape Proposal Network for 3D Instance Segmentation in Point Cloud. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[51] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural Motifs: Scene Graph Parsing with Global Context. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[52] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3D Point Capsule Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[53] Yibiao Zhao and Song chun Zhu. Image Parsing with Stochastic Scene Grammar. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2011. 2

[54] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1