International Journal of Computer Vision (2022) 130:630–651
https://doi.org/10.1007/s11263-021-01546-9

Learning 3D Semantic Scene Graphs with Instance Embeddings

Johanna Wald[1] · **Nassir Navab**[1] · **Federico Tombari**[1,2]

Received: 27 February 2021 / Accepted: 26 October 2021 / Published online: 22 January 2022

Abstract
A 3D scene is more than the geometry and classes of the objects it comprises. An essential aspect beyond object-level perception is the scene context, described as a dense semantic network of interconnected nodes. Scene graphs have become a common representation to encode the semantic richness of images, where nodes in the graph are object entities connected by edges, so-called relationships. Such graphs have been shown to be useful in achieving state-of-the-art performance in image captioning, visual question answering and image generation or editing. While scene graph prediction methods so far focused on images, we propose instead a novel neural network architecture for 3D data, where the aim is to learn to regress semantic graphs from a given 3D scene. With this work, we go beyond object-level perception, by exploring relations between object entities. Our method learns instance embeddings alongside a scene segmentation and is able to predict semantics for object nodes and edges. We leverage *3DSSG*, a large scale dataset based on *3RScan* that features scene graphs of changing 3D scenes. Finally, we show the effectiveness of graphs as an intermediate representation on a retrieval task.

Keywords Scene graphs · 3D scene understanding · Semantic segmentation

1 Introduction

Rapid progress has been made in digitizing the real world in 3D with data obtained from cameras, scanners and depth sensors. Advanced 3D reconstruction algorithms paired with recent 3D sensor technology are able to robustly scan complex environments. Naturally, the focus of the research community shifted from capturing basic geometric properties towards extracting more abstract scene representations, motivated by the wealth of applications that require such high-level understanding. The fields of applications range from robotics in unstructured environments and autonomous driving, Augmented and Mixed Reality for gaming or education, to generating scene layouts for interior design and architecture. Understanding the 3D surroundings to a degree that allows autonomous interaction or sophisticated augmentation requires to robustly extract semantic details, such as scene parts and objects, together with their geometry and attributes (*e.g.*, pose), as well as with the relationships among each other. This aspect has been often overlooked due to its inherent complexity.

The research community recently focused on a variety of perception tasks, including 3D object detection (Zhou and Tuzel 2017) and recognition (Su et al. 2015; Song et al. 2015), instance segmentation (Hou et al. 2018; Lahoud et al. 2019; Thomas et al. 2019; Yi et al. 2019), 3D shape prediction (Najibi et al. 2020) as well as classification and semantic segmentation (Rosinol et al. 2020a; Qi et al. 2017a, b; Dai and Nießner 2018; Rethage et al. 2018; Liu et al. 2020). While these methods have the objective of obtaining object knowledge, contextual data is mainly used to advance object-level understanding and the semantics of the relationships themselves are mostly neglected. A direction worth noting here are methods that either estimate scene layouts or perform holistic scene parsing (Huang et al. 2018; Nie et al. 2020). Rather than focusing on the semantic aspects, they estimate the geometric properties of the environment as well as the individual pose of the scene entities. Scene graphs are abstract representations that store the semantics of a scene, where the graph nodes are scene entities and their connections are meaningful relation-

Communicated by Zuzana Kukelova.

✉ Johanna Wald
johanna.wald@tum.de

Nassir Navab
navab@in.tum.de

Federico Tombari
tombari@in.tum.de

[1] CAMP, Technical University of Munich, Munich, Germany

[2] Google Inc., Zürich, Switzerland

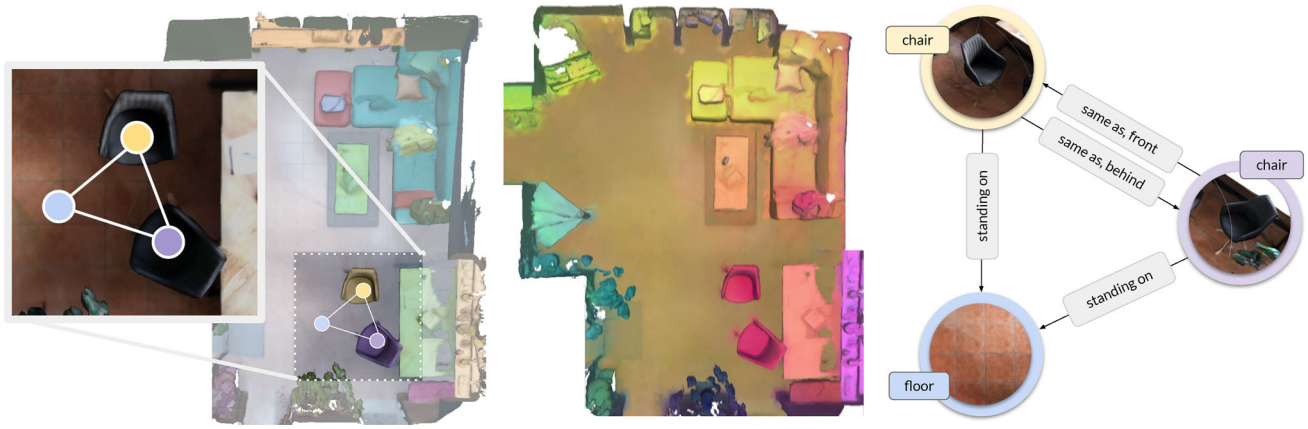

Fig. 1 Scene Graph Prediction: Given the 3D model of a scene (left), we leverage a 3D network to learn semantics and instance embeddings (center) that encode the points in the scene. We then infer a scene graph $\mathcal{G}$ by feeding these features into a graph prediction module that predicts class labels for instances and edges (right)

ships between them *e.g.* support relations (Nathan Silberman Derek Hoiem and Fergus 2012). Such a representation is frequently used in the image domain for higher-level task such as partial (Wang et al. 2014) and full image retrieval (Johnson et al. 2015), image generation (Johnson et al. 2018) or even manipulation (Mittal et al. 2019; Dhamo et al. 2020). While 2D scene graph datasets such as Visual Genome (Krishna et al. 2017) or NYUv2 (Nathan Silberman Derek Hoiem and Fergus 2012) are widely available and feature relationships between scene instances and often instance attributes, scene graphs in 3D have not been explored much.

Although, 3D graphs have been used in computer graphics for decades to store 3D mesh data, the respective edges usually do not represent semantic connections but rather relative transformations such that when a parent node is relocated, the change is applied in a hierarchical fashion to all child nodes. Only recently, *semantic* scene graphs have started to emerge in the 3D context (Gay et al. 2018; Armeni et al. 2019; Rosinol et al. 2020b). Armeni *et al.* construct graphs for buildings, including rooms, major objects, camera views and the relations between these entities (Armeni et al. 2019). Rosinol et al. (2020b) incorporates dynamics to this representation by additionally considering moving humans. Both (Armeni et al. 2019) and (Gay et al. 2018) propose multi-view graph prediction methods based on 2D masks (Armeni et al. 2019) and object detection networks (Gay et al. 2018). They estimate graphs from images while we operate on 3D data directly.

In this work, we explore semantically rich 3D graphs similarly to what has been successfully proposed and implemented in the image domain. We introduce a novel method based on sparse convolutions to predict 3D scene graphs from 3D data directly to ultimately gain high-level knowledge that goes beyond object understanding. Our method learns an instance embedding alongside semantic segmentation and is capable of predicting the class labels of both object nodes and edges by directly feeding the scene features into a graph prediction module, see Fig. 1. Notably, our network does not require any knowledge of the scene *e.g.* any segmentation at test time. For training and evaluation purposes, we utilize *3DSSG*, a large-scale dataset based on *3RScan* that features rich scene graphs of changing 3D scenes. *3DSSG* describes the semantics of scene entities and their attributes as nodes and relationships as edges. For research purposes, the dataset is publicly available for download.[1] Furthermore, we open source our scene graph prediction method.

The scene graphs in this work are semantically rich and particularly dense. This implies that all object instances *e.g. chairs, sofas* or *bags* as well as the structural components of a room *e.g.* the *floor*, different *walls* or the *ceiling* are represented as independent nodes in the graph. For structures, this specifically means, that each planar entity is represented as a different instance in the scene. A regular room, see Fig. 4, consists of 1 floor and 4 walls while a multi-floor scan has at least two floor instances and several walls. The nodes are described by attributes such as the color, shape or affordances and the connections between them are semantically meaningful relationships *e.g. lying on*, *same as*, see Fig. 2. Notably, this scene graph representation is inspired by the image graphs proposed by Johnson et al. (2015). In contrast to images, the dimensionality and context of 3D data is quite comprehensive, resulting in large-scale scene graphs. Despite – or because of – this, we believe graphs are particularly suited for 3D since they are a human-readable, compact representation that includes all major scene information. However, to learn scene graphs from 3D data turns

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

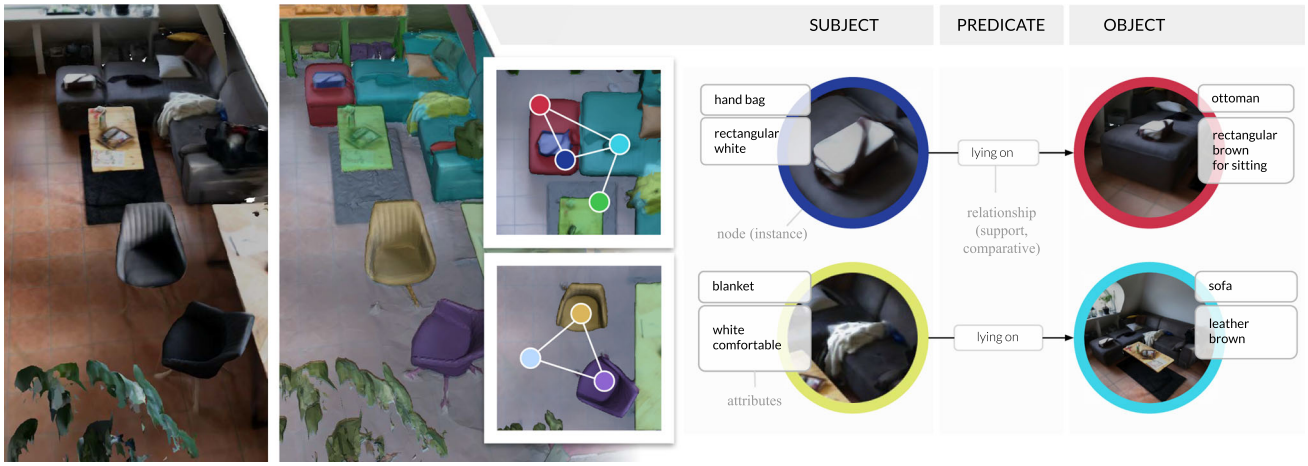

Fig. 2 Scene Graph Representation in *3DSSG* consists of hierarchical class labels and attributes of scene nodes as well as semantic relationship between them. A scene graph tuple connects a subject with an object node with a predicate

out to be quite challenging since it requires not only handling real world data with noise but also ambiguities in the node and relationship descriptions. This includes various scanning patterns and clutter as well as data labels that might not be unique and distinctive. The surface of a *blanket* that *covers* a *tidy bed* is technically also *part of it* and might sometimes even be labelled as such. Actually, the fact that the *blanket* in Fig. 2 is *lying on* the couch gives us a hint about its class while it might be identified as *towel* if found in a *bathroom*. On the other side, a set of *chairs* that *lookalike* could drastically vary in appearance if occluded and their neighborhood and connections differ depending on where they are positioned. We believe graphs can be particularly beneficial in changing indoor scenes *e.g.* when matching a single 2D image against a pool of 3D scenes taken at a different time possible including lighting and object changes such as rigid and non-rigid changes and even (dis-)appearance of scene entities. We demonstrate how they are effectively used as an intermediate representation when computing scene similarity. Furthermore, our experiments show how they are fundamentally resilient to dynamic environments.

In summary, our contributions are three-fold: a) we propose an embedding based method to learn semantic scene graphs from a raw 3D point cloud. b) We further publish scene graphs for the localization benchmark *RIO10* complementary to the large scale 3D scene graph dataset, *3DSSG* (Wald et al. 2020a)[2]. The datasets are an extension of *RIO10* Wald et al. 2020b and *3RScan* (Wald et al. 2019) and include graph annotations in form of relationships, instance attributes and class label hierarchies for each instance. c) We finally show the effectiveness of such graphs in a retrieval application. Compared to our previous publication (Wald et al. 2020a) we propose a new method that can predict 3D scene graphs from 3D scenes directly, not requiring any prior knowledge such as segmentation masks. While (Wald et al. 2020a) uses a PointNet backbone to encode objects and relationships based on the segmented point cloud we utilize a 3D backbone architecture based on sparse convolutions. We incorporate surface normals and color to learn semantic features and an embedding space for node segmentation. While (Wald et al. 2020a) assumes a ground truth class-agnostic segmentation, we initialize graph nodes with segmented clusters making the method applicable in real-world setups.

2 Related Work

Semantic Scene Graphs and Images Scene graphs were originally introduced by Johnson et al. (2015) with a novel dataset of 5, 000 images and are today, also thanks to the success of Visual Genome (Krishna et al. 2017), a common, compact representation for many scene understanding tasks. By definition, the scene entities are grounded to different regions of the image and, while Visual Genome is quite large, the edges that describe the connections between nodes are rather sparse. On top, attributes highlight the properties of the object in more detail but are rarely used in practice. The effectiveness of scene graphs has been demonstrated when solving different scene understanding tasks including image retrieval (Liu et al. 2007; Johnson et al. 2015), scene captioning (Yang et al. 2019), visual question answering (Teney et al. 2017) or image generation from graphs alone (Johnson et al. 2018), interactively (Ashual and Wolf 2019) or for image editing tasks (Mittal et al. 2019; Dhamo et al. 2020). Many of these methods either rely, or build upon, image-based scene graph prediction, a particularly well studied problem (Lu et al. 2016a; Peyre et al. 2017; Xu et al. 2017; Newell and Deng 2017; Li et al. 2017; Yang et al. 2018; Zellers et al. 2018; Li et al. 2018c; Herzig et al. 2018; Qi et al.

[2] https://waldjohannau.github.io/RIOSG

2019; Zareian et al. 2020). Classical approaches usually follow a multi-stage process: first, nodes are initialized with an off-the-shelf object detector, such as Faster R-CNN (Ren et al. 2015). In the second stage, the predicates are predicted based on object proposals. This stage is commonly designed as a predicate/relationship classification task that takes features of the entities as input. The features for the nodes and edges are either low-level features *e.g.* the bounding box and their relative configuration, directly extracted from a CNN (Xu et al. 2017; Yang et al. 2018) or a combination of these (Peyre et al. 2017). Lu et al. (2016a) and Qi et al. (2019) go beyond visual features and incorporate linguistic knowledge by leveraging language priors when predicting relationships. To improve efficiency, Yang et al. (2018) prune relationships and only keep meaningful tuples before computing the predicates. Li et al. (2018c) instead propose a bottom-up clustering method to factorize the scene into sub-graphs while maintaining spatial information. Relationship predictions – often simply termed visual relationship detection (Lu et al. 2016a; Peyre et al. 2017) – is commonly implemented as a local process and computed independently for object pairs.

Recently, some methods build a graph to iteratively refine the edge and/or node features *e.g.* using attentional graph neural networks (Yang et al. 2018; Qi et al. 2019) or message passing with a recurrent neural network (Xu et al. 2017). Similarly, in Li et al. (2017), parallel and sequential message passing is used for information propagation among objects and relationships, while other works demonstrate the importance of permutation invariance (Herzig et al. 2018), suggest embedding based architectures (Newell and Deng 2017) or a graphical contrastive loss (Zhang et al. 2019). Contrarily to all the above, (Zareian et al. 2020) propose to learn how to bridge scene graphs and knowledge graphs by means of an iterative graph-based neural network.

Notably, datasets such as Visual Genome (Krishna et al. 2017) or Visual Relationship Detection (VRD) (Lu et al. 2016a) offer scene graphs for a fairly large amount of images, enabling the implementation and evaluation of the aforementioned methods. Many leverage deep learning and therefore require large quantities of training data. Even though impressive progress has been made in the image domain in the last few years, scene graph prediction is still considered a challenging task, due to the complexity and interdependence of object detection and relationship/predicate prediction. Besides the lack of a large-scale 3D graph dataset, many of the presented concepts are not directly transferable to 3D due to the complexity and memory restrictions when using higher dimensional data.

Learning 3D Semantics and Instances 3D scene understanding involves the extraction of knowledge from 3D environments, including its objects and structure, their categories and spatial and semantic relationships with each other. One of the most common 3D scene understanding tasks is 3D semantic segmentation where a single label from a fixed set of classes is assigned to each voxel or point of the 3D scene. Early methods process the dense volumetric data directly in form of occupancy grids or TSDF volumes (Dai et al. 2017). Dai and Nießner (2018) show that incorporating multi-view features is beneficial while Huang *et al.* utilize the textured 3D mesh directly (Huang et al. 2019). More recently, Kundu et al. (2020) has proposed a virtual multi-view fusion technique that – compared to previous approaches – achieves significantly improved segmentation accuracy.

Another popular line of research has focused on lightweight point network architectures (Qi et al. 2017a; Li et al. 2018b) while incorporating hierarchical context (Qi et al. 2017b; Engelmann et al. 2017), hybrid architectures (Rethage et al. 2018), 3D capsule networks (Zhao et al. 2019) and efficient 3D sparse convolutions (Graham et al. 2018; Choy et al. 2019), which enable effective processing of large scale 3D data. Notably, these methods are among the state of the art on challenging benchmarks (Dai et al. 2017). We utilize the sparse convolutions proposed by (Graham et al. 2018) as backbone features of our method.

It is important to note that semantic segmentation alone does not enable reasoning about object instances. In contrast, 3D instance segmentation focuses on foreground objects, where, additionally to the semantic label, instance masks are computed. Instance segmentation methods can roughly be categorised into bottom-up and top-down approaches, where in the latter proposals are generated from the input data. These proposals are often filtered *e.g.* via NMS (non-maximum suppression) and are leveraged to compute bounding boxes and/or 3D masks. Hou *et al.* suggest a proposal-based instance segmentation, similar to 2D Mask R-CNN (He et al. 2017), that contrarily uses dense, volumetric 3D data paired with multi-view features (Hou et al. 2018). VoteNet, on the other hand, directly predicts the center of the object bounding boxes via a novel voting scheme (Qi et al. 2019). Alternatively, proposal-free methods were suggested (Lahoud et al. 2019; Han et al. 2020; Jiang et al. 2020), which use metric learning to generate embeddings that are trained to be similar on points/voxels of the same instance and different for other instances. For his purpose, multi-task learning (Lahoud et al. 2019) or an occupancy loss (Han et al. 2020) were proposed. Given rich instance embeddings, instances are obtained by clustering similar features. PointGroup (Jiang et al. 2020) proposes an improved grouping scheme while 3D-MPA (Engelmann et al. 2020) combines bottom-up and top-down approaches by learning object centers that are grouped via a graph neural network, avoiding the classical NMS.

Notably, instance segmentation methods usually ignore background elements such as walls and the floor. While in panoptic segmentation (Kirillov et al. 2019; Narita et al. 2019) walls are segmented, but are still not recognized as

instances. In contrast to aforementioned works we include those structural entities since they are required to identify the majority of support relationships *e.g.* a picture that hangs on a particular wall plane.

3D Object Context and Scene Layout The works discussed so far are mostly object-focused and incorporate context only to improve overall segmentation performance. On the contrary, holistic scene understanding perceives scenes as a whole by combining several related tasks, such as the prediction of scene layout (Song et al. 2015; Avetisyan et al. 2020), camera pose (Huang et al. 2018) or object pose and shape or reconstruction (Nie et al. 2020; Najibi et al. 2020). To parse a 3D scene, different representations have been proposed including stochastic grammars (Zhao et al. 2011; Liu et al. 2014), dependency graphs or tree structures where, by definition, leave nodes are independent scene entities (or object parts) and intermediate parent nodes represent functional entities (Liu et al. 2014). Zhao et al. (2011) use a stochastic grammar with three production rules: AND, OR and SET to model the scene layout, detected objects, planes and the background.
Scene synthesis methods that aim to generate realistic scene models and layouts incorporate knowledge about an objects' context and the scene composition either directly or indirectly (Jiang et al. 2018; Shi et al. 2019). Jiang et al. (2018) describe a configurable 3D scene synthesis pipeline based on stochastic grammars, so-called spatial and-or graphs. GRAINS Li et al. (2018a) combine a recursive VAE with object retrieval to iteratively generate a layout and objects. Shi et al. (2019) also suggests an iterative approach based on a novel variational recursive autoencoder. Kulkarni et al. (2019) on the other hand, create a 3D scene given a 2D image. They show that predicting relative transformations between objects improves their pose predictions compared to a neighbourhood-independent computation. Fisher et al. (2011) use kernel functions to compare and retrieve similar 3D scenes by incorporating relationships such as support and proximity. Similarly, Ma et al. (2018) parse natural language into graphs and retrieve 3D scenes that fulfill requested compositions.

Graph structures have also been used for object understanding. In such a setup, different nodes represent different object parts of *e.g.* a chair, such as *chair leg* or *backrest*. Te et al. (2018) solve semantic part segmentation by using a graph neural network. StructureNet (Mo et al. 2019) goes even further and represent a shape as a hierarchical graph of embeddings where each object is a latent graph of its composing parts to ultimately be able to sample and interpolate aiming to generate new, novel shapes. They however learn a graph for each object category and their utilized relationships are restricted to relative transformations and known physical connections.

Only a few works have explored scene graphs in 3D. Gay et al. (2018) propose a 2.5D graph dataset based on ScanNet (Dai et al. 2017), Armeni et al. (2019), on the other hand, suggest hierarchical 3D scene graphs. They split the different components of a scene into 4 different layers: cameras, objects, buildings and rooms. Rosinol et al. (2020b) propose an additional dynamic layer to model humans. Armeni et al. (2019) does not include RGB-D sequences, and more importantly, structural components such as walls or floors are not included in their graphs and they therefore lack some inter-instance relationships such as support. A comparison of these – as well as related 2D datasets (Johnson et al. 2015; Krishna et al. 2017; Nathan Silberman Derek Hoiem and Fergus 2012) – is given in Table 1.

Additionally to the data, Armeni et al. (2019) and Gay et al. (2018) propose graph prediction methods. Armeni et al. (2019) sample images from a panoramic camera and apply a regularization technique to 2D mask predictions aiming to obtain improved 3D object nodes. Gay et al. (2018) on the other hand, feed object features extracted from a continuous image sequence into a recurrent neural network. They operate in 2.5D on a static setup while *3RScan* (Wald et al. 2019) has dynamically changing scenes which enables new, challenging tasks such as the newly introduced 2D-3D scene retrieval.

In our previous publication, Wald et al. (2020a), we proposed a meth-od to predict 3D scene graphs by incorporating the ground truth (class-agnostic) segmentation. In this work, we remove this assumption and operate on the raw point cloud directly. We use a 3D network on the full scene and extract object-level features from point features instead of parsing each ground truth object one by one. This makes our method more scalable, especially on dense graphs. In contrast to Wald et al. (2020a), where only scene graphs are predicted, our method additionally estimates 3D semantic and node segmentation and does not require any prior knowledge about the scene, therefore it is directly applicable in real-world scenarios.

3D Scene Retrieval Visual retrieval has a long history in computer vision, and is often embedded in other tasks such as object detection, object or scene alignment or camera pose estimation (Gálvez-López and Tardós 2011; Torii et al. 2015; Glocker et al. 2015; Anoosheh et al. 2019; Arandjelović et al. 2016; Deng et al. 2016; Lu et al. 2016b). Retrieving a source given some query data becomes most challenging when they do not share the same domain (Dahnert et al. 2019; Abdul-Rashid et al. 2018, 2019; Avetisyan et al. 2019, 2020) including natural scene changes (Wald et al. 2020b, 2019). An extensive literature exists on retrieval of CAD models given an image (Izadinia et al. 2017; Sun et al. 2018) or a 3D scene (Avetisyan et al. 2019, 2020). Reviewing the full literature goes beyond the scope of this paper. To motive the

Table 1 Comparison of *3DSSG* with related 2D and 3D scene graph datasets. (*) *Spaces* refers to buildings, floors and rooms

Dataset	2D	3D	Data source	Graph nodes	Graph edges
Johnson et al. (2015)	✓	✗	YFCC100m Thomee et al. (2016), COCO Lin et al. (2014)	Sparse 2D objects	Semantic
Visual Genome Krishna et al. (2017)	✓	✗	108k images	Sparse 2D objects	Semantic
NYUv2 Nathan Silberman Derek Hoiem and Fergus (2012)	✓	✗	1449 images	Dense 2D instances	Vertical and horizontal support
3D Scene Graph Armeni et al. (2019)	✓	✓	Gibson Xia et al. (2018)	Sparse 3D objects, spaces(*)	Occlusion, spatial, relative volume
3D Dynamic Scene Graph Rosinol et al. (2020b)	✓	✓	uHumans	Sparse 3D objects, humans, spaces(*)	Inclusion, adjacency, spatio-temporal
Scannet-SGG Gay et al. (2018)	✓	✓	ScanNet v1 Dai et al. (2017)	Sparse 3D objects	Same set, plane, part, support
3DSSG Wald et al. (2020a)	✓	✓	*RIO* Wald et al. (2019)	Dense 3D instances	Semantic

usage and suitability of our 3D graphs in high-level tasks we use them as an intermediate representation for 2D-3D scene retrieval of changing indoor scenes.

3 3D Semantic Scene Graphs

The method proposed in this work is trained and evaluated on *3DSSG* which is based on *3RScan* (Wald et al. 2019), a collection of every day 3D indoor scenes with 1482 sequences with semantically segmented 3D models. It features approximately 450 unique, diverse indoor environments, captured over a long period of time with change annotations (Wald et al. 2019). Unlike any other dataset, this allows reasoning about object instances and their changes but also about their relationships. Besides the full scene graphs, a smaller dataset that features a subset of objects and predicates is also made available.[3] Additionally to *3DSSG* that has been released in Wald et al. (2020a) we also provide 2D semantic scene graphs for the camera re-localization benchmark, *RIO10* (Wald et al. 2020b). In summary, we offer (a) scene graphs, (b) RGB-D sequences with camera poses and intrinsics, (c) textured 3D models with point coordinates and surface normals $\{p_i\}_{i=1}^{N}$ where $p_i = (x, y, z, n_x, n_y, n_z) \in \mathbb{R}^6$, and an instance-level semantic segmentation defined as $\{l_i\}_{i=1}^{N}$ where l_i describes the label of p_i. Finally, the data provides d) change annotations such as scene and object alignments and bounding boxes.

Formally, semantic scene graphs $\mathcal{G}$ are defined by a set of nodes $\mathcal{N}$ with attributes $\mathcal{A}$ and edge triplets $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. The nodes $\mathcal{N} = \{n_i\}_{i=1}^{L}$, are consistent across re-scans of the same environment and correspond to the instance IDs in $\{l_i\}$. Specifically, each node n_i is described by a set of properties $\mathcal{A}$, as well as by a hierarchy of classes $c = (c_1, ..., c_d)$, where $c \in \mathbb{C}$ and $\mathbb{C}$ is a set of all valid class labels. Given the 3D instance segmentation, 3D geometry and depth can be obtained for each node. Some of the nodes are connected by means of edges based a set of predicates $\mathbb{P}$ such as *standing on, hanging on* or *more comfortable than*. We define $\mathcal{E} \subseteq \mathcal{N} \times \mathbb{P} \times \mathcal{N}$ where a relationship tuple, *subject-predicate-object* $(n_s, p, n_o) \in \mathcal{E}$ (see Fig. 2), directionally connects a subject node $n_s \in \mathcal{N}$ to an object node $n_o \in \mathcal{N}$ such that

$$\mathcal{E} \subseteq \{(n_s, p, n_o) | n_s, n_o \in \mathcal{N}, o \neq s \text{ and } p \in \mathbb{P}\}. \quad (1)$$

In the following we provide a detailed overview about the nodes and their attributes (Sect. 3.1) as well as the graph edges (Sect. 3.2). More details about the annotation procedure of *3RScan* and *3DSSG* can be found in Wald et al. (2019) and Wald et al. (2020a) respectively.

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

3.1 Nodes and Attributes

The most important graph entities are the nodes described by coarse-to-fine class labels *e.g. armchair* → *chair* → *seat* → *furniture* → *artefact*. The human annotation represents the lowest and finest level of our class hierarchy which is recursively parsed based on a lexical dictionary extracted from WordNet (Fellbaum 1998).

Object properties give more details about the visual and physical appearance of an instance. Overall, *3DSSG* consists of 93 different attributes and are split into 3 different groups: static properties that stay the same over time, dynamic attributes that possibly change and affordances that describe the functionality of an object. While some properties require manual annotation, others are obtained automatically from the object's geometry. In the following paragraphs, more details are given about the different types of attributes.

Static and Dynamic Properties The first category describe the visual appearance of an entity. This includes geometric properties such as shape, size and rigidity, as well as color and texture. The size of the object class relative to all other objects of the same category is computed automatically by comparing instance masks while other more complex attributes are manually annotated, including the texture, material, color or shape using a custom annotation interface. While Static properties, *e.g.* an object's appearance usually do not change, dynamic attributes can change over time. They describe the state of an entity such as *open/closed* or *on/off*. Some state categories are class specific *e.g.*, appliances such as televisions, refrigerators or ovens can be turned *on* and *off* while a bed cannot. Interestingly, dynamic properties provide insights about potential human activity, see Fig. 3.

Affordances The interaction possibilities of a scene entity can been described by using affordances (Gibson 1979; Xia et al. 2018; Armeni et al. 2019). In *3DSSG* they are associated to object classes, *e.g.* a *seat* is for *sitting*. Notably, some affordances are only viable if the object is in a specific state *e.g.* only a *closed door* can be *opened*, which is a direct link to the dynamic attributes and is of relevance in presence of scene dynamics.

3.2 Relationships

2D scene graph datasets often describe a human action occurring in an image *e.g. girl-throwing-frisbee* or *boy-reading-book*. Since our scenes do not include humans directly the attention shifts away from those actions to the following three main relationship categories: a) support b) proximity and c) comparative relationships; all described in the following.

Support Relationships are important connections between objects (Nathan Silberman Derek Hoiem and Fergus 2012) as they give hints about physical stability and object dependencies and are therefore of relevance in robotics applications, where robot-scene interaction is carried out. By definition, all entities are supported by at least one other node, excluding the floor, which is the root node in our representation and, as such, does not require any support. The support relationships in *3DSSG* are assigned by automatically computing a list of support candidates, followed by a manual correction and semantic annotation to produce desired relationship tuples – so-called *semantic support* relations – such as *chair-standing on-floor* or *cabinet-hanging on-wall*.

Fig. 3 Two example scenes at two different observations where object states changed due to some human action. *Top:* someone might have slept in the bed (*bed* is *tidy/messy*), *Bottom:* someone might have used the toilet (toilet seat is *down/up*)

Proximity Relationships describe the spatial arrangement of objects on the same support level. Proximity relationships such as *left* or *right* require a reference view, which we choose to be a top-down bird view with +x as right and +y as front. Spatial relationships are automatically computed and only valid in 3D and therefore require re-computation in 2D.

Comparative Relationships are connections related to object properties, see Sect. 3.1. They are computed from node annotations and include, but are not limited to comparisons of size (*e.g. bigger/higher than*), shape or material (*e.g. same shape/material*) color (*e.g. darker than, same color*) or state (*e.g. cleaner than*).

3.3 2D Scene Graphs

Since ground truth instance segmentation as well as RGB-D image sequences with corresponding camera poses $\mathbf{P}_i$ are provided, 2D graphs $\mathcal{G}_{2D}$ are directly obtainable from the 3D counterpart $\mathcal{G}_{3D}$ using a simple rendering procedure, see Fig. 4. Given the 2D instance image $\mathbf{I}_{s,i}$, rendering the 3D graph implies filtering out nodes and edges that do not include a

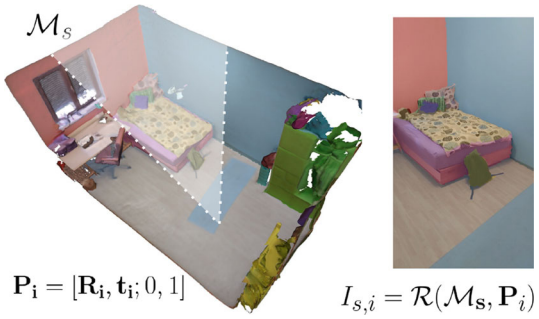

Fig. 4 2D scene graph generation using a rendered 3D scene from *3RScan*.

visible entity. Support and comparative relationships can be directly transferred while proximity relationships need to be recomputed, since they are viewpoint-dependent.

The rendering pipeline and 2D graphs for the camera re-localization benchmark, *RIO10* (Wald et al. 2020b) will be made publicly available[4]. Compared to some other 2D scene graph datasets (Krishna et al. 2017) our data provides depth images and semantic instance masks – additionally to bounding boxes – similarly to the much smaller NYUv2 (Nathan Silberman Derek Hoiem and Fergus 2012).

4 Methodology

In the following section we first introduce the problem statement (see Sect. 4.1) giving a high-level overview of the different tasks and challenges involved when predicting 3D semantic scene graphs, before diving into details of our proposed method (see Sect. 4.2).

4.1 Problem Statement

Scene graph prediction from 3D data is a complex problem that requires solving several interdependent tasks. While our 3D reconstructions do not capture on-camera motion, a few other factors make this task challenging: The 3D graphs are – compared to the 2D counterpart – quite large and densely connected while the underlying data usually covers a relatively large space. Simply applying techniques developed for images to the 3D domain is therefore often unfeasible.

3D scene graph prediction, first and foremost, requires the 3D space to be encoded with meaningful features that incorporate long-range semantic information (P1). This feature space is the foundation of the scene graph prediction which relies on the identification of scene entities including objects and scene structure. Our scene graphs are dense, therefore, every single 3D point has to be assigned to a node in the graph while the number of nodes is unknown (P2). Finally, semantics is obtained by classifying the detected nodes and their connecting edges given a list of object class and predicate labels (P3). Notably, *3DSSG* features a long list of different object classes with an unbalanced long-tail distribution: a few labels occur regularly, while the majority is relatively rare (P4). Specifically, the top-12 most common object classes appear approximately as often as all the remaining classes together.

Ultimately, the goal is to end up with scene graphs that are rich and meaningful enough for high-level tasks *e.g.* visual question answering or scene retrieval (P5).

4.2 3D Scene Graph Embedding Network

An overview of our method is given in Fig. 5. It operates end-to-end and consists of two main parts; a 3D network and a graph network. Given the 3D model of a scene, we identify scene graph nodes $\mathcal{N}$ by learning a semantic instance segmentation (see Sect. 4.2.1). The network assigns the 3D points of the scene to an entity by processing its coordinates, surface normals and texture colors. Our segmentation aims to produce similar features for points on the same instance and different features for points on different instances, Fig. 5b. In the second stage (see Sect. 4.2.2), a graph is build from the extracted scene nodes in a fully connected fashion $\mathcal{E} = \mathcal{N} \times \mathcal{N}$, Fig. 5d. When constructing the graph, the output features from the first stage are aggregated for the corresponding nodes and edges respectively. The object classes are refined and predicates (if any) are predicted for the object pairs, see Fig. 5c. The output of our method is a 3D semantic instance segmentation as well as a 3D scene graph, Fig. 5e.

4.2.1 3D Instance and Semantic Network

Instead of computing features for each object node and relationship separately, as in Wald et al. (2020a), we process the entire scene at once and obtain object-level features from the points by grouping them afterwards using instance masks. This makes our features more descriptive than Wald et al. (2020a). Furthermore, such a procedure is more scalable since the points are only processed once and not redundantly for all its edges which is particularly favorable in dense graphs.

We augment each element of the point cloud with its surface normal. Additionally, the color value of our textured 3D models is extracted by querying the image texture at the associated pixel coordinates. Our input is therefore a $(N \times 9)$-dimensional feature, where N is the cardinality of the point cloud. *3RScan* consists of high-poly meshes of real-world reconstructions which contain faces and vertices of the scene surface. While the RGB extraction operates on meshes, our network is also able to process colored point clouds directly.

[4] https://waldjohannau.github.io/RIOSG

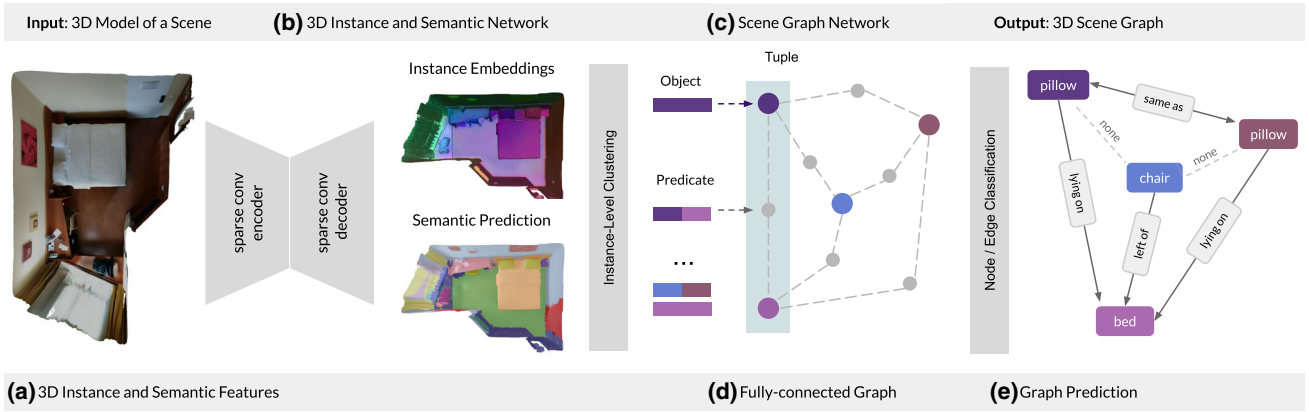

Fig. 5 Scene Graph Prediction Network: Given the 3D model of a scene (left), we infer a scene graph $\mathcal{G}$ (right). Visual point features (**a**) are extracted with a 3D network (**b**). The features are grouped and arranged in a graph structure (**d**) for further processing within our scene graph network (**c**). The predicted graph (**e**), consisting of labeled object nodes and directed labeled edges

The meshes are reconstructions of real-world environments and therefore consist of several thousands of polygons and vertices. During training, the data loader randomly samples 3D points from the data. We feed this input into a sparse convolutional encoder-decoder network with a backbone size of 256, similarly to the one proposed by Najibi et al. (2020). The output feature dimensionality of the $3 \times 3 \times 3$ convolutional layers is 64, 96, 128, 160, 192 and 224 respectively. After each layer of the encoder a max pool operation is applied. In the decoder, features are upsampled and corresponding encoder features are concatenated via skip connections à la U-Net (Ronneberger et al. 2015; Çiçek et al. 2016). Conversely to other instance segmentation works, we do not consider any points as background (*e.g.* walls and floors), instead we purposely include them as unique instances since we want to represent structural components as independent nodes in the graph. We predict semantic logits f_i and instance embeddings e_i, $\{(f_i, e_i)\}_{i=1}^{N}$ for all given points without any masking. They are obtained by applying two sparse convolutional layers on the output of the decoder with a ReLu and batch norm.

Inspired by the success of bottom-up instance segmentation, we similarly learn an embedding space and obtain instances by clustering its features. During training, we uniformly sample point indices $|V|$ on all object instances to counteract data unbalance caused by objects of varying sizes before an N-pair metric learning loss (Sohn 2016) is computed

$$L_e = -\frac{1}{|V|} \sum_{i,j \in V} l_e(i, j). \tag{2}$$

The loss uses a pairwise similarity metric $s(i, j)$ between i and j such as

$$l_e(i, j) = \begin{cases} log(s(i, j)), & \text{if } l_i = l_j \\ log(1 - s(i, j)) & \text{otherwise} \end{cases} \tag{3}$$

$$s(i, j) = \frac{2}{1 + \exp(||e_i - e_j||)}. \tag{4}$$

where i and j are sampled point indices in the input and e_i and e_j represent their respective embedding vectors. Figure 5b, Fig. 6b visualize the learned point features on example scenes. We map the embedding vectors to RGB color space by applying PCA for visualization purposes. It can be seen that our features are able to distinguish points on different object instances.

Additionally to learning the embedding space, we use a cross-entropy loss $\mathcal{L}_s$ to learn semantic classes per point. We jointly learn the semantic and instance embeddings and weight their losses equally such that

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_e. \tag{5}$$

The final semantic segmentation (Sect. 5.1) is obtained by applying a regularization technique that averages the semantic outputs of regions with similar embeddings. During training we randomly sample points on the point clouds and process them in a voxelized fashion with a resolution of $(0.02m \times 0.02m \times 0.02m)$ within our sparse convolutional network. While our network processes data in a discretized fashion, voxels are mapped back to the input points during test time. Translation augmentation is applied with a maximum offset of $0.1m$, scenes are scaled with a factor between 0.9 and 1.1, and rotational augmentation around the z axis ranges from $-180°$ and $180°$. We implement our network with tensorflow (Abadi et al. 2015) based on the open source implementation of tf3d (Google Research 2021).

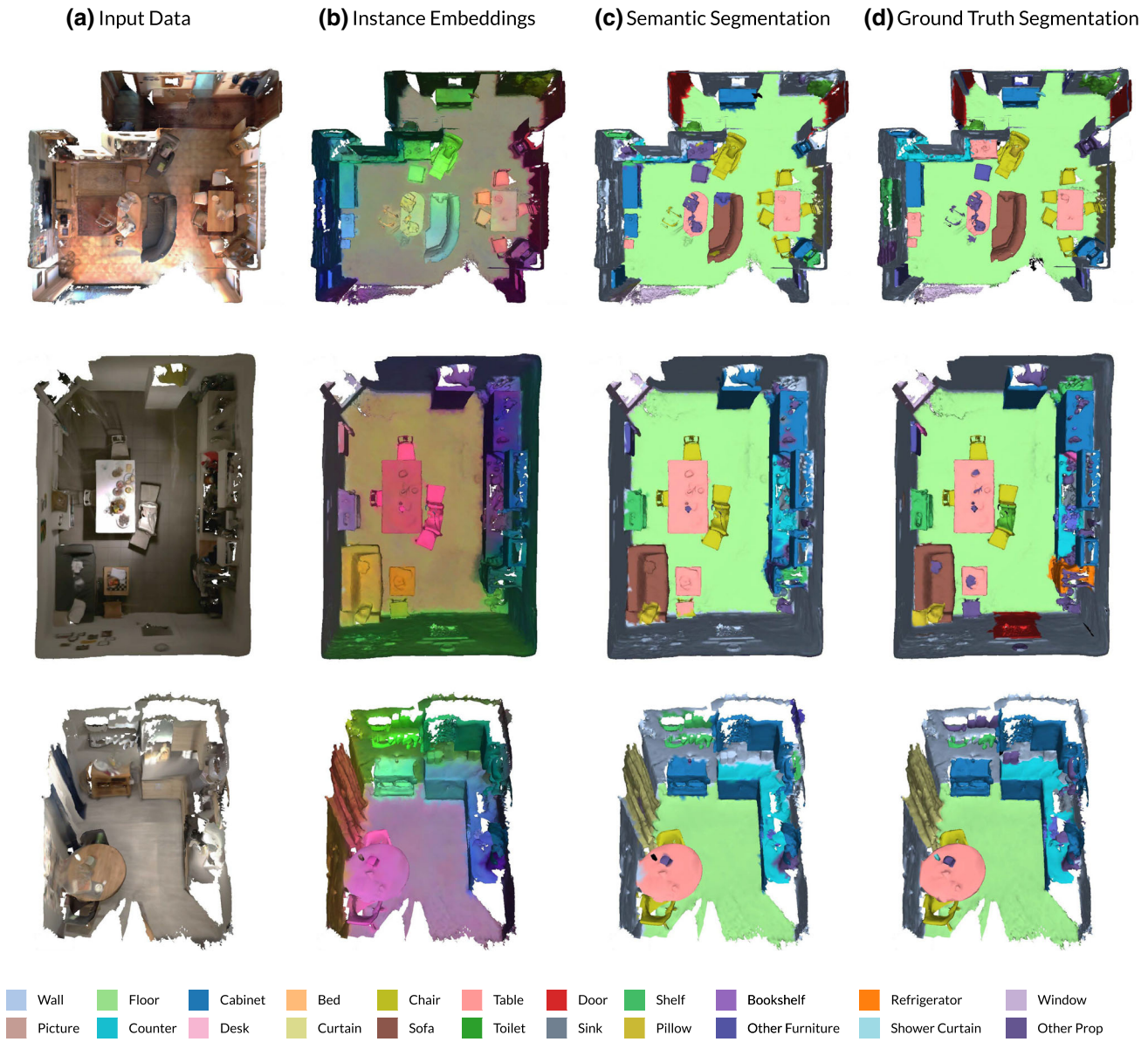

Fig. 6 Qualitative results on example scenes of the validation set of *3RScan*. (**b**) Extracted embeddings, (**c**) semantic segmentation (see legend) and the corresponding ground truth segmentation (**d**). The input data of our method is the 3D model, shown on the left (**a**)

4.2.2 Scene Graph Network

During training the scene graph is constructed with ground truth instance segmentation $l = \{l_i\}_{i=1}^{N}$. Contrarily, at test time, a clustering function ρ is used to cluster points into instance estimates

$$\hat{l} = \{\hat{l}_i\}_{i=1}^{N} = \rho(\{(f_i, e_i)\}_{i=1}^{N}). \tag{6}$$

In our implementation we use kmeans++ (Arthur and Vassilvitskii 2007), though any clustering function could be applied to our embedding space. Similar to other graph prediction works (Lu et al. 2016a; Xu et al. 2017; Yang et al. 2018) visual features are extracted for each node ϕ_k and edge ϕ_r respectively. We experiment with different node encodings obtained from aggregated and concatenated features of the 3D network. In our experiments we set $\phi_k = [\bar{e}_k, \bar{f}_k]$ where the 3D features are averaged for all the points of an instance k such that

$$\bar{e}_k = \frac{1}{|\{p_i\}_{i_k}|} \sum_{i \in i_k} e_i \quad \text{where } i_k = \{i = l : k\}, \tag{7}$$

with $\bar{f}_k$ is computed in a similar fashion. We arrange the nodes in a graph structure, building relationship triples (*subject, predicate, object*) to form a fully connected graph

with subject ϕ_s, object ϕ_o and predicate units ϕ_r. ϕ_r is derived from the subject ϕ_s and object ϕ_o node features, their relative position – computed from the centroids $\bar{p}_s$ and $\bar{p}_o$ – as well as their relative bounding boxes such that $\phi_r = [\phi_s, \bar{p}_s - \bar{p}_o, b_s - b_o, \phi_o]$. Class labels are learned with a cross-entropy loss that instead of predicting only a single semantic class per node has two semantic output heads that use different semantic class sets learned jointly in a coarse-to-fine fashion. This means our network learns a very coarse semantic class *e.g. other furniture* (from NYUv2 (Nathan Silberman Derek Hoiem and Fergus 2012)) as well as uncommon but more descriptive classes such as *baby bed* or *storage unit*.

In a real-world setup, certain object pairs might have multiple valid relationships that describe their interactions. In Fig. 2, *e.g.* one chair can be *behind* the other and simultaneously have the same visual appearance, represented as a *same as* relationship. We therefore formulate $\mathcal{L}_{\text{pred}}$ as a 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 no label (*none*). In summary, we train our graph model end-to-end and optimize the object $\mathcal{L}_{\text{obj}}$ and predicate classification loss $\mathcal{L}_{\text{pred}}$ jointly

$$\mathcal{L}_{\text{total}} = \lambda \cdot \mathcal{L}_{\text{obj}} + \mathcal{L}_{\text{pred}} \tag{8}$$

where λ is a weighting factor between object and relationship prediction and set to 0.5 in our experiments. We ensure consistency of proximity relationships, like *left* or *right*, across re-scans by avoiding rotation augmentation during training. Instead, re-scans are aligned with the respective references using the provided scan-to-scene transformations. Notably, our method does not require any filtering of the ground truth graph data and is able to process all nodes and edges of a scene at once.

The object and relationship predictors have four and six fully-connected layers followed by batch norm and ReLu. For training we use an SGD optimizer with a learning rate of 10^{-2}. Please note that, while in practice both networks can be trained jointly, we train them separately for the sake of easier convergence. Specifically, we first train the 3D network and then freeze its layers when training the scene graph prediction.

5 Evaluation

In this section we present different experiments to analyse the main aspects of our proposal. First, we evaluate our 3D semantic segmentation network on *3RScan* (Wald et al. 2019), see Sect. 5.1 to validate the quality of our underlying features (see Sect. 4.1.P1) and then ablate the node segmentation quality (see Sect. 4.1.P2) and the effect of the embedding dimension and input features on segmentation performance (Sect. 5.2). Further, we show the performance of our graph prediction method (see Sect. 4.1.P3) on *3DSSG* by comparing it against a baseline approach as well as the method proposed in (Wald et al. 2020a) and report per-class evaluation and and analysis of rare and often occurring class labels, see Sect. 5.3. Finally, we show the application of scene graph prediction (see Sect. 4.1.P5) for the task of scene retrieval in Sect. 5.4.

Table 2 3D semantic segmentation on the validation split of *3RScan* based on 27 class categories

Method	mIoU^{27}
P4Contrast (3D context) Liu et al. (2020)	40.8
P4Contrast (2D-3D context) Liu et al. (2020)	44.2
Ours (3D semantics)	**44.8**

Bold indicates the best performing model/method

5.1 3D Semantic Segmentation

Following the evaluation scheme of Liu et al. (2020), Tables 2 and 4 lists the average IoU of the 3D semantic segmentation on *3RScan* (Wald et al. 2019) using 27 object categories. We compare our method against Liu et al. (2020), as it was also evaluated on this dataset and outperform them by a small margin. Table 3 reports the per-class performance using the NYU40 (Nathan Silberman Derek Hoiem and Fergus 2012) class set. Notably, our method is able to successfully segment challenging classes such as doors and windows. Qualitative results of our 3D semantic segmentation on *3RScan* are shown in Fig. 6. (a) is the input of our method, (b) visualizes the learned point embeddings mapped to color space and (c) and d) are the predicted and ground truth semantic segmentation respectively. This experiment shows that our 3D network produces meaningful features to be further processed within the graph prediction network.

5.2 3D Node Segmentation

In the following we evaluate how well scene nodes are detected in 3D scenes using different embedding dimensions, see Table 4. It can be observed that the dimensionality of the embedding vector has only a small impact on the segmentation quality. The embedding dimension of 256 gives the best performance, although the margin is relatively small.

In the experiment, we adapt the commonly used Mean Average Precision metric where the Average Precision (AP) determines the area under the precision-recall curve. First, the IoU is computed between each ground truth and predicted segment of the same class. Each prediction mask is compared with the ground truth to obtain an IoU and is considered true positive or false positive based on a thresh-

Table 3 3D semantic segmentation on the validation split of *3RScan* on the NYU40 class set

Method	mIoU	Recall	Prec.	Tub	Bed	Shelf	Cab.	Chair	Counter	Curtain	Desk	Door	Floor	Other	Pic	Fridge	Shower	Sink	Sofa	Table	Toilet	Wall	Wind.
Ours (3D semantics)	53.0	65.8	72.4	76.6	41.8	22.5	52.4	74.3	21.1	70.0	16.7	36.2	81.9	29.5	23.1	33.4	78.5	61.9	67.4	54.2	84.7	71.2	61.7

Table 4 3D node segmentation on the validation split of *3RScan* with different embedding dimensions

	AP	mAP25	mAP50
embedding dimension 64	8.0	18.6	41.6
embedding dimension 128	9.3	19.8	41.6
embedding dimension 256	**9.4**	**21.7**	**44.0**

Bold indicates the best performing model/method

old (25% for mAP25 and 50% for mAP50). The evaluation scheme was adapted from ScanNet (Dai et al. 2017) and CityScape (Cordts et al. 2016) which is inspired by the evaluation scheme of COCO (Lin et al. 2014). The corresponding precision and recall per class is given in Table 5.

As expected, the method achieves best scores for common and distinctive classes chairs, tables and toilets and worst for the categories counter, desk and bookshelf which are either ambiguous (desk vs. table, counter vs. cabinet) or cluttered (shelf, counter).

We further evaluate the effect of different features on segmentation quality. In Tables 6 and 8 we cluster instances based on point embeddings alone (1), embeddings and the semantic features (2), embeddings and the point cloud coordinates (3) as well as all features combined (4) while keeping the embedding dimension 256 fixed.

The clustering seems to be most accurate when only embedding features are used which might sound counter-intuitive at first. Our intuition is that spatial information potentially cause over-segmentation of bigger objects *e.g.* couches or walls and adding semantic information might result in unwanted merging of close-by instances of the same classes *e.g.* chairs, see Fig. 7. Please note that the embedding network was specifically trained to produces a distinctive feature space by parsing (a) color, and (b) geometry and jointly learns (c) semantics, and (d) instance embeddings.

Even for challenging scenes with many semantically and visually similar objects, the feature space is quite distinctive, see also Fig. 8.

An additional qualitative evaluation of the node features can be found in Fig. 9 where we use t-SNE (van der Maaten and Hinton 2008) to qualitatively visualize the features of our scene nodes. Each element in the plot represents a scene entity in the validation set and its color corresponds to the respective NYU40 class. We can observe how objects of the same category are nicely clustered together and away from categories including different shapes, *e.g.* (a) toilets or (b) curtains and similar classes (desks and tables) are closer together.

5.3 Semantic Scene Graph Prediction

In the following we compare the performance of our proposed scene graph embedding network ③ with two variants of our ②

Table 5 Precision and recall per NYU40 category of 3D node segmentation on the validation split of *3RScan*

	Average	Wall	Floor	Cabinet	Bed	Chair	Sofa	Table	Door	Window	Shelf	Pic	Counter	Desk	Curtain	Fridge	Shower	Toilet	Sink	Tub	Other
Precision @ IoU50	16.4	8.1	9.6	12.8	12.5	41.7	5.2	37.8	20.9	9.6	0.0	22.6	3.1	1.8	8.1	6.7	11.5	60.0	20.3	10.0	25.6
Recall @ IoU50	30.0	22.0	43.3	29.0	16.7	52.0	22.4	47.3	28.6	24.7	0.0	11.9	6.5	8.3	23.9	22.2	42.9	88.9	22.0	50.0	37.0
Precision @ IoU25	31.4	22.7	17.5	29.1	50.0	57.7	17.4	56.9	43.8	22.5	18.2	41.9	6.2	11.8	20.4	30.0	15.4	62.5	45.3	15.0	43.0
Recall @ IoU25	62.6	61.7	79.3	66.0	66.7	72.0	75.0	71.3	59.8	57.6	57.1	22.0	12.9	54.2	60.2	100.0	57.1	92.6	49.2	75.0	62.0

Table 6 3D node segmentation on the validation split of *3RScan* using different input features when clustering

	AP	mAP25	mAP50
(1) embedding	**9.4**	**21.7**	**44.0**
(2) embedding, semantic	3.8	9.8	27.7
(3) embedding, spatial	6.9	16.4	41.3
(4) embedding, semantic, spatial	4.9	11.7	28.0

Bold indicates the best performing model/method

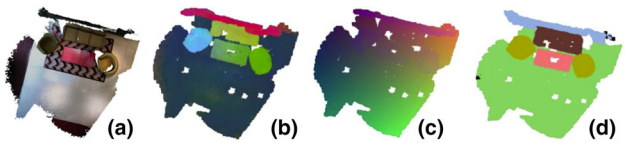

Fig. 7 **a** Colored 3D model and **b** embedding, **c** spatial and **d** semantic input features used for 3D node segmentation of an example scene, see Table 6

scene graph point network (Wald et al. 2020a) and a baseline ① inspired by the visual relationship prediction proposed by Lu et al. (2016a). We re-implemented (Lu et al. 2016a) and adapted the method to operate on 3D using an underlying PointNet backbone. Similar to Wald et al. (2020a), node and edge features are extracted from the point cloud for each node and edge respectively. We follow the same train and validation split proposed by Wald et al. (2019).

Evaluating semantic scene graphs is non-trivial as it involves several interdependent tasks: detection and segmentation of object instances, prediction of the semantic class labels as well as visual relationship detection. When evaluating large 3D scene graphs we indeed evaluate object, predicate and relation (triples) independently as commonly done in 2D scene graph literature. Even though, our method does not require any ground truth segmentation, we utilize it in this experiment to fairly compare against the methods in Wald et al. (2020a).

Previous graph prediction works propose complex evaluation schemes (Lu et al. 2016a; Xu et al. 2017; Yang et al. 2018; Wald et al. 2020a) that consider a match correct if it ranks within the top-n predictions. Such an evaluation scheme helps in challenging scenarios or when dealing with ambiguous class categories and large label sets. In this work, we only adopt the strictest top-1 metric where a sample is considered correct iff it exactly matches the ground truth, see Table 7 where the top-1 score for objects and predicates as well as relations is reported – where the latter one evaluates (*subject, predicate, object*) triplets. In this experiment, our network outperforms all other methods when predicting object and predicate labels as well as relations but still leaves some room for improvements due to the challenging setup and high variability of the graphs.

Fig. 8 Embedding space of an example scene with many similar and spatially close repetitive objects

To further analyse the networks' predictions we report a broken down performance score of the predicates, see Table 8. Rare predicates *e.g. cover, hanging in, lying in* are most challenging which is likely related to the unbalanced data distribution and complexity of the graph data.

We furthermore report the node classification score of the 20 most and least occurring object classes, see Table 9 to better understand the networks behaviour when working with diverse and imbalance data. It is no surprise that great scores are accomplished for common classes while the network fails when confronted with rare entities. Interestingly, misclassifications of small objects *e.g. pan, scale, bread, napkins* or *papers* are often confused with categories where they most commonly appear, accidental taking too much context into consideration *e.g. kitchen counter, table or shelf*.

A similar experiment where common and uncommon relations are analysed shows the same results, see Table 10. While the network achieves remarkable performance in predicting common relationships such as *chair-stand-ing on-floor* it struggles to predict any of the less common relations correctly. Interestingly, the network has learned strong and meaningful priors from the data and is therefore able to produce meaningful relations such as *pillow-lying on-sofa* or *clothes-hanging on-wall* for the relations *box-supported by-sofa* and *flowers-hanging on-wall* (rare occurrence, noisy reconstruction or an inconsistency in the ground truth).

To complement this set of experiment we analysed the most common mispedictions of relation triplets, see Table 11. Similarly, specific labels such as *side table* or *shower*

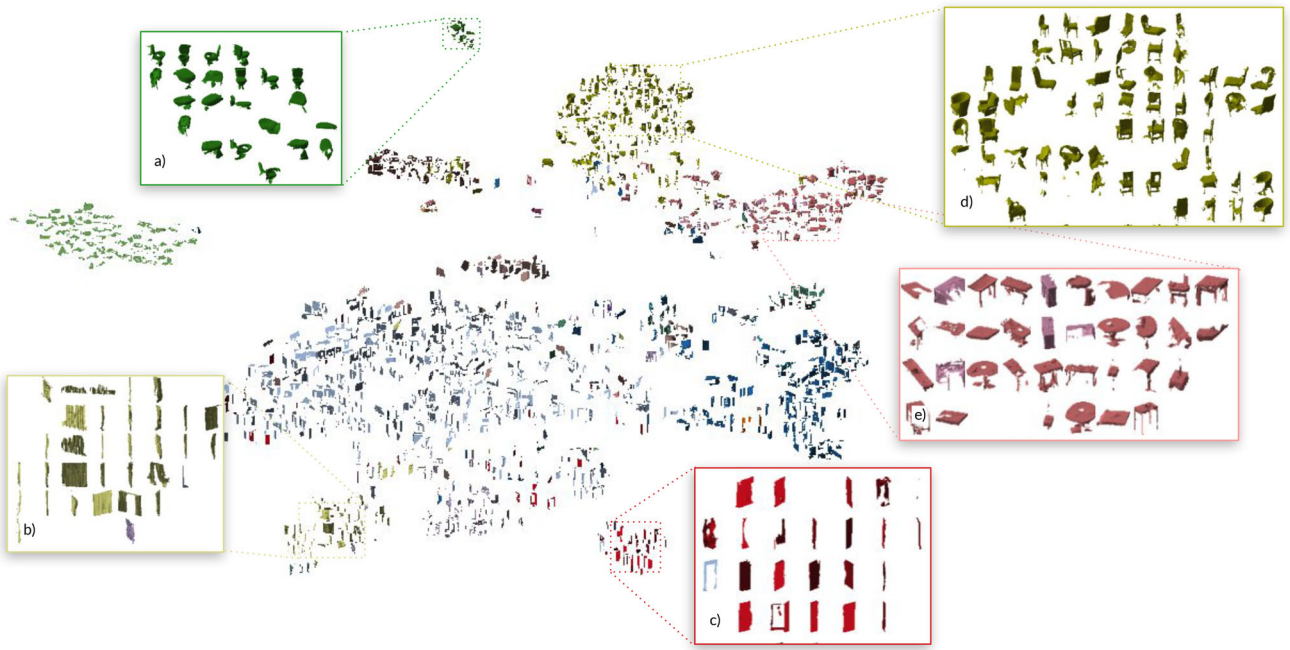

Fig. 9 Learned embedding space of the scene graph nodes. Colors correspond to the semantic NYU colors used in Fig. 6 **a** toilets, **b** curtains, **c** doors, **d** chairs, **e** tables/desks

Table 7 Evaluation of scene graph prediction on *3DSSG* using ground truth class-agnostic segmentation

Method	Object	Predicate	Relation
① Relation Pred. Baseline	35.1	15.0	5.4
② Multi Pred., GCN Feat Wald et al. (2020a)	39.9	58.5	20.3
② Multi Pred., PN Feat Wald et al. (2020a)	33.4	64.2	31.5
③ Ours	**52.0**	**71.2**	**42.5**

Bold indicates the best performing model/method

Table 8 Per category evaluation of the scene graph predicates of *3DSSG*

	Avg.	Hanging In	Part Of	Cover	Leaning Against	Standing In	Belonging To	Build In	Connected To	Lying In	Bigger Than	Smaller Than	Hanging On	Higher Than	Lower Than	Supported By	Lying On	Attached To	Standing On
Occurrences	–	0.02	0.21	0.18	0.67	0.81	0.45	0.68	0.59	0.39	6.94	6.94	4.00	13.89	13.89	2.48	5.89	21.06	20.92
Accuracy	42.5	0.0	83.3	0.0	0.0	8.0	46.9	90.9	64.7	0.0	2.4	8.2	83.8	57.4	67.7	10.8	67.2	87.6	84.8

wall are confused with their more general counterparts *table* and *wall*. In terms of predicates, the method seems to confuse *smaller than* and *lower than* and *bigger than* and *higher than*.

Finally, Fig. 10 shows example scene graph predictions generated by our method. Since dense graphs on large scenes are hard to parse and visualise, we only show support edges. For incorrect predictions, the ground truth is appended in round brackets after the predicted label. In the following section we finally show that our graphs are accurate and representative to be used within a retrieval application, see sect. 5.4.

5.4 Scene Retrieval

In the following we utilize graphs for *image-based 3D scene retrieval in changing indoor environments* as proposed in Wald et al. (2020a). The aim of the task is to identify the corresponding scene from a pool of 3D scans given a single 2D image acquired at a different point in time. The query and target data are not only from different domains but additionally undergo scene changes, *e.g.* moving objects, different illumination. To bridge the domain gap between images and 3D data, this task is carried out using scene graphs. We use different similarity metrics to match the graphs to the correct 3D scenes using object semantics as well as scene context. Computing the minimum edit-distance between two graphs is a complex problem, we therefore map our graphs to node and edge multi-sets – containing potential repetitive elements that occur more than once in the scene. Based on two graphs $\mathcal{G}$ and $\mathcal{G}'$, a similarity function τ is computed. In our tests we explore the Jaccard τ_J and Szymkiewicz-Simpson τ_S coefficient such that

$$f(\hat{\mathcal{G}}, \hat{\mathcal{G}}') = \frac{1}{|\hat{G}|} \sum_{i=1}^{|\hat{\mathcal{G}}|} \tau(s(\hat{\mathcal{G}}^{(i)}), s(\hat{\mathcal{G}}'^{(i)})) \tag{9}$$

where $\hat{\mathcal{G}}$ is an augmented graph $\hat{\mathcal{G}} = (\mathcal{N}, \mathcal{E}, \mathcal{R})$ with binary edges $\mathcal{E}$. Table 12 reports the matching accuracy using single 2D images or a 3D re-scan of an indoor scene. To do so, we compute the scene graph similarity between each re-scan (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. The size of image and 3D scene graphs are significantly different, the Szymkiewicz-Simp-son coefficient is therefore used in 2D-3D matching while f_S is chosen in the 3D-3D scenario. It can be observed that our scene graphs ③ significantly improve matching accuracy in 2D, as well as 3D, compared to our previous work ② and the baseline model ① (see Sect. 5.3).

Table 9 Scene graph node classification on *3DSSG*: Head-Tail-Effect, 20 most (head, top) and least (tail, bottom) occurring categories and the corresponding (*) most likely prediction per label

Ground truth node	Occurrences (%)	Accuracy (%)	Prediction(*)
wall	15.76	94.1	wall
chair	6.56	94.2	chair
pillow	4.41	84.2	pillow
box	4.28	29.9	box
shelf	3.92	73.4	shelf
floor	3.72	99.3	floor
ceiling	3.46	93.9	ceiling
plant	2.66	54.2	plant
door	2.44	62.0	door
table	2.42	83.7	table
window	2.41	61.5	window
item	2.23	15.2	plant
lamp	2.17	46.7	lamp
curtain	2.14	73.9	curtain
object	1.74	8.2	shelf
cabinet	1.68	45.7	cabinet
picture	1.59	33.3	picture
doorframe	1.05	60.7	doorframe
sink	1.00	76.7	sink
cushion	0.77	1.7	pillow
nightstand	0.46	0.0	pillow
stuffed animal	0.11	0.0	pillow
objects	0.06	0.0	cabinet
napkins	0.06	0.0	table
scale	0.06	0.0	kitchen counter
ladder	0.05	0.0	shelf
recycle bin	0.05	0.0	windowsill
pan	0.04	0.0	kitchen counter
shower floor	0.03	0.0	windowsill
footstool	0.02	0.0	box
bread	0.02	0.0	table
cup	0.02	0.0	item
papers	0.02	0.0	shelf
socket	0.02	0.0	windowsill
washing powder	0.02	0.0	item
grass	0.01	0.0	floor
jacket	0.01	0.0	clothes
magazine rack	0.01	0.0	item
rocking chair	0.01	0.0	chair
soap dish	0.01	0.0	box

Note that for the purpose of this experiment, predicted 2D graphs are obtained by rendering the predicted 3D graphs as described in Sect. 3.

6 Future Work

3D semantic scene graphs are a rich and compact representation for holistic 3D scene understanding and we believe they are an excellent representation for persistent 3D mapping of long-term/dynamic 3D environments. The prediction of entities is an essential requirement of persistent mapping where the representation of a space is updated based on new observations and detected changes. Dynamics could be captured by learning persistent features for association across time and augmenting the graphs with poses. A localization algorithm would then need to jointly detect changes

Table 10 Scene graph relation classification on *3DSSG*: Analysis of most (head, top) and least (tail, bottom) tuples and the corresponding (*) most likely prediction per relation

Ground truth relation	Accuracy (%)	Prediction(*)
wall-attached to-floor	91.1	wall-attached to-floor
chair-standing on-floor	94.1	chair-standing on-floor
chair-same as-chair	90.8	chair-same as-chair
ceiling-attached to-wall	94.7	ceiling-attached to-wall
cabinet-standing on-floor	43.0	cabinet-standing on-floor
table-standing on-floor	87.2	table-standing on-floor
shelf-attached to-wall	79.4	shelf-attached to-wall
pillow-lower than-pillow	63.5	pillow-lower than-pillow
pillow-higher than-pillow	57.1	pillow-higher than-pillow
plant-standing on-floor	78.4	plant-standing on-floor
decoration-standing on-couch table	0.0	plant-standing on-table
desk-supported by-wall	0.0	table-attached to-wall
flowers-hanging on-wall	0.0	clothes-hanging on-wall
item-leaning against-wall	0.0	clothes-hanging on-wall
item-lying on-counter	0.0	plant-standing on-shelf
kettle-standing on-kitchen counter	0.0	item-standing on-kitchen counter
lamp-standing on-cabinet	0.0	object-lying on-cabinet
organizer-standing on-table	0.0	item-standing on-desk
oven-attached to-kitchen counter	0.0	commode-build in-commode
oven-supported by-cabinet	0.0	cabinet-build in-commode
picture-attached to-wall	0.0	tv-hanging on-wall
box-supported by-sofa	0.0	pillow-lying on-sofa
bucket-standing on commode	0.0	box-standing on commode
cabinet-belonging to-wall	0.0	cabinet-attached to-wall
cleanser-supported by-wall	0.0	picture-attached to-wall

Table 11 Scene graph relation classification on *3DSSG*: Analysis of most common misclassifications

Ground truth relation	Accuracy (%)	Prediction(*)
pillow-smaller than-pillow	13.9	pillow-lower than-pillow
couch-standing on-floor	0.0	sofa-standing on-floor
pillow-bigger than-pillow	0.0	pillow-higher than-pillow
pillow-lying on-couch	0.0	pillow-lying on-sofa
shower wall-supported by-floor	0.0	wall-attached to-floor
commode-standing on-floor	22.6	cabinet-standing on-floor
kitchen cabinet-standing on-floor	0.0	cabinet-standing on-floor
desk-standing on-floor	17.7	table-standing on-floor
couch table-standing on-floor	0.0	table-standing on-floor
box-lower than-box	0.0	box-lower than-shelf
picture-standing on-wall	3.5	picture-hanging on-wall
ottoman-standing on-floor	0.0	chair-standing on-floor
side table-standing on-floor	0.0	table-standing on-floor
coffee table-standing on-floor	5.0	table-standing on-floor
box-higher than-box	0.0	shelf-higher than-box
radiator-connected to-wall	0.0	heater-connected to-wall
tv stand-standing on-floor	0.0	cabinet-standing on-floor
stool-standing on-floor	0.0	chair-standing on-floor
cushion-lying on-sofa	0.0	pillow-lying on-sofa
dining chair-standing on-floor	0.0	chair-standing on-floor

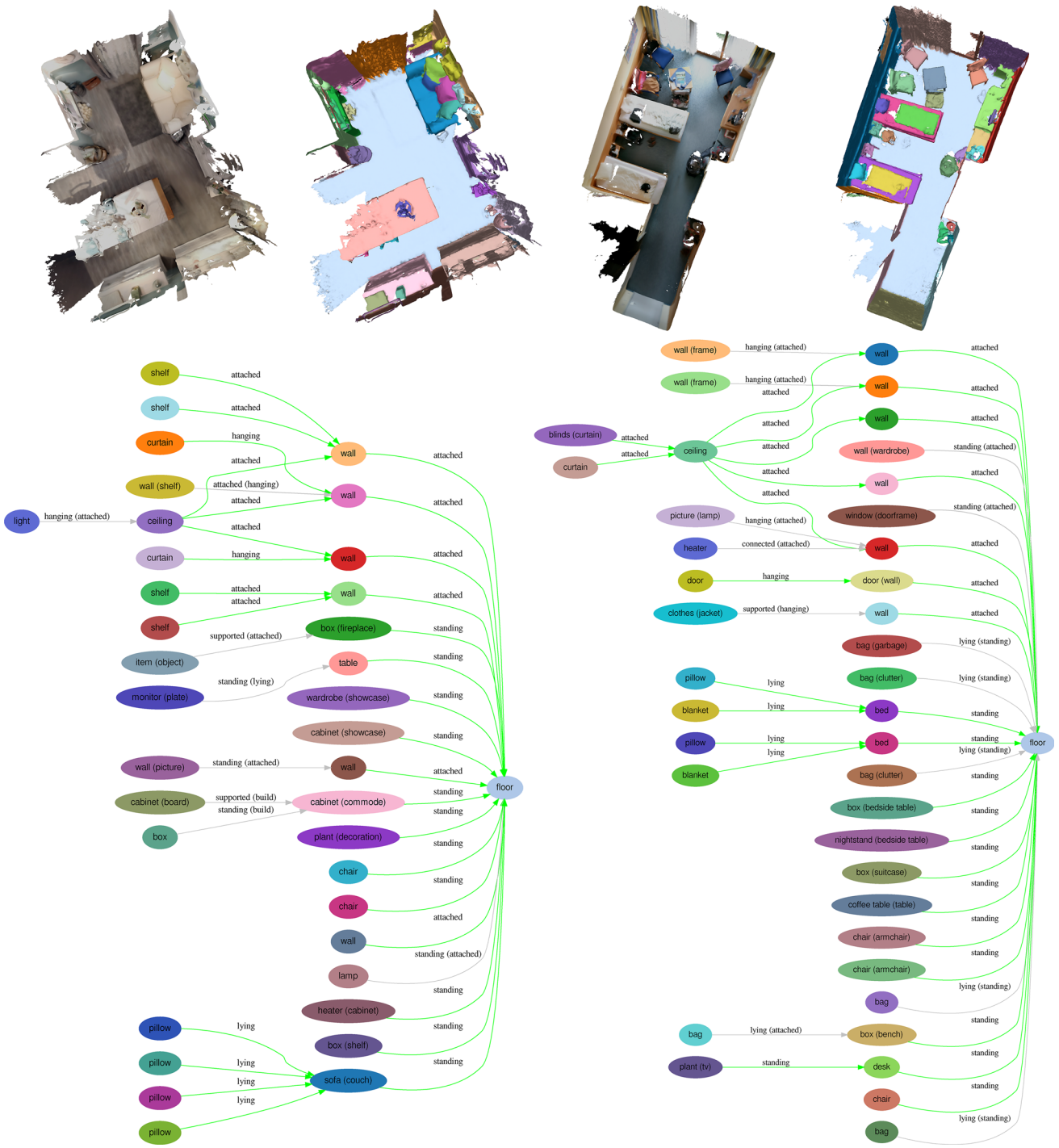

Fig. 10 Qualitative results of our scene graph prediction model (best viewed in the digital file)

in the graph and estimate the poses in this dynamic setup. This long-term knowledge could then be stored as additional connections in the graph structure. Another interesting direction is represented by augmenting the resulting 3D temporal scene graphs with comprehensive semantics beyond simple class labels and 6DoF object poses. This could provide the representation needed for robust and efficient persistent mapping. In this context, it might be worth exploring hierarchical scene graphs where the labels of parent nodes are carried on to the children enforcing it to be *more specific e.g.* using a novel loss. Notably, building scene graphs requires long-range attention; it is therefore important to rely on efficient networks and techniques *e.g.* sparse convolutions, especially when handling large-scale outdoor scenes.

Table 12 Evaluation: scene retrieval of changing re-scans (2D and 3D) to reference 3D scans

		Graph	Top-1	Top-3	Top-5
f_J	$\hat{\mathcal{G}}_{3D} \rightarrow \hat{\mathcal{G}}_{3D}$	①	0.29	0.50	0.59
f_J	$\hat{\mathcal{G}}_{3D} \rightarrow \hat{\mathcal{G}}_{3D}$	②	0.34	0.51	0.56
f_J	$\hat{\mathcal{G}}_{3D} \rightarrow \hat{\mathcal{G}}_{3D}$	③	**0.64**	**0.79**	**0.80**
f_S	$\hat{\mathcal{G}}_{2D} \rightarrow \hat{\mathcal{G}}_{3D}$	①	0.10	0.25	0.32
f_S	$\hat{\mathcal{G}}_{2D} \rightarrow \hat{\mathcal{G}}_{3D}$	②	0.13	0.38	0.42
f_S	$\hat{\mathcal{G}}_{2D} \rightarrow \hat{\mathcal{G}}_{3D}$	③	**0.43**	**0.71**	**0.74**

Bold indicates the best performing model/method

Ultimately, finding a scalable, weakly/self- or even unsupervised and generic solution for persistent mapping could solve many of the challenges of long-term and dynamic 3D scene understanding and eventually help bring some of the theoretical models into practice.

7 Conclusion

This work goes beyond classical object-level scene understanding and explores regression of 3D scene graph with a neural network. A novel graph prediction method is proposed based on the semantically rich scene graph dataset *3DSSG* which is build upon *3RScan* (Wald et al. 2019). Our method predicts nodes and edges representing the objects' semantic classes and their relationships, by directly operating on 3D scans of scenes. Notably, our work explores regressing these graphs from real-world 3D data without any priors. Our experiments show that the features learned with our 3D network enable the detection and segmentation of graph nodes while the underlying features are very descriptive and therefore useful for semantic scene graph prediction. We show that the unbalanced distribution and large number of different categories in real-world scenes introduces additional challenges, which require learning an even richer, fine-grained feature space given only a few training samples.
We believe scene graphs could ultimately serve as a persistent representation for long term 3D scene understanding and are useful to bridge domain gaps, as shown in the cross-domain task, *image-based 3D scene retrieval in changing indoor environments*. They could potentially even enable new useful applications of scene understanding, such as text-based search or VQA (Visual Question & Answer).

Acknowledgements This work was funded by 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.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015). https://www.tensorflow.org/. Software available from tensorflow.org

Abdul-Rashid, H., Yuan, J., Li, B., Lu, Y., Bai, S., Bai, X., Bui, N.M., Do, M.N., Do, T.L., Duong, A.D., He, X., Le, T.K., Li, W., Liu, A., Liu, X., Nguyen, K.T., Nguyen, V.T., Nie, W., Ninh, V.T., Su, Y., Ton-That, V., Tran, M.T., Xiang, S., Zhou, H., Zhou, Y., Zhou, Z. (2018). 2D Image-based 3D Scene Retrieval. In: *Eurographics Workshop on 3D Object Retrieval*.

Abdul-Rashid, H., Yuan, J., Li, B., Lu, Y., Schreck, T., Bui, N.M., Do, T.L., Holenderski, M., Jarnikov, D., Le, K.T., Menkovski, V., Nguyen, K.T., Nguyen, T.A., Nguyen, V.T., Ninh, T.V., Rey, P., Tran, M.T., Wang, T. (2019). Extended 2D Scene Image-Based 3D Scene Retrieval. In: S. Biasotti, G. Lavoué, R. Veltkamp (eds.) *Eurographics Workshop on 3D Object Retrieval*.

Anoosheh, A., Sattler, T., Timofte, R., Pollefeys, M., Gool, L.V. (2019). Night-to-Day Image Translation for Retrieval-based Localization. In: *International Conference on Robotics and Automation*. IEEE.

Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J. (2016). NetVLAD: CNN architecture for weakly supervised place recognition. In: *Conference on Computer Vision and Pattern Recognition*. IEEE.

Armeni, I., He, Z.Y., Gwak, J., Zamir, A.R., Fischer, M., Malik, J., Savarese, S. (2019). 3D Scene Graph: A Structure for Unified Semantics, 3D Space, and Camera. In: *International Conference on Computer Vision*.

Arthur, D., Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. In: *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics*, pp. 1027–1035.

Ashual, O., Wolf, L. (2019). Specifying object attributes and relations in interactive scene generation. *International Conference on Computer Vision*.

Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A.X., Nießner, M. (2019). Scan2CAD: Learning CAD model alignment in RGB-D scans. In: *Conference on Computer Vision and Pattern Recognition*.

Avetisyan, A., Khanova, T., Choy, C., Dash, D., Dai, A., Nießner, M. (2020). SceneCAD: Predicting object alignments and layouts in RGB-D scans. In: *European Conference on Computer Vision*.

Choy, C., Gwak, J., Savarese, S. (2019). 4D spatio-temporal convNets: Minkowski convolutional neural networks. In: *Conference on Computer Vision and Pattern Recognition*, pp. 3075–3084.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S., Brox, T., Ronneberger, O. (2016). 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In: *Conference on Computer Vision and Pattern Recognition*.

Dahnert, M., Dai, A., Guibas, L., Nießner, M. (2019) Joint embedding of 3D scan and CAD objects. In: *International Conference on Computer Vision*.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M. (2017). ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: *Conference on Computer Vision and Pattern Recognition*.

Dai, A., Nießner, M. (2018). 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In: *European Conference on Computer Vision*

Deng, L., Chen, Z., Chen, B., Duan, Y., & Zhou, J. (2016). Incremental image set querying based localization. *Neurocomputing*

Dhamo, H., Farshad, A., Laina, I., Navab, N., Hager, G.D., Tombari, F., Rupprecht, C. (2020). Semantic image manipulation using scene graphs. In: *Conference on Computer Vision and Pattern Recognition*.

Engelmann, F., Bokeloh, M., Fathi, A., Leibe, B., Nießner, M. (2020). 3D-MPA: Multi proposal aggregation for 3D semantic instance segmentation. In: *Conference on Computer Vision and Pattern Recognition*.

Engelmann, F., Kontogianni, T., Hermans, A., Leibe, B. (2017). Exploring spatial context for 3D semantic segmentation of point clouds. In: *International Conference on Computer Vision*

Fellbaum, C. (Ed.). (1998). *WordNet: An electronic lexical database*. Cambridge: MIT Press.

Fisher, M., Savva, M., Hanrahan, P. (2011). Characterizing structural relationships in scenes using graph kernels. *Transactions on Graphics*.

Gálvez-López, D., Tardós, J.D. (2011). Real-time loop detection with bags of binary words. In: *International Conference on Intelligent Robots and Systems*, pp. 51–58. IEEE.

Gay, P., Stuart, J., Del Bue, A. (2018). Visual graphs from motion (VGfM): Scene understanding with object geometry reasoning. In: *Asian Conference on Computer Vision*, pp. 330–346. Springer.

Gibson, J. J. (1979). *The ecological approach to visual perception*. Boston: Houghton Mifflin.

Glocker, B., Shotton, J., Criminisi, A., & Izadi, S. (2015). Real-time RGB-D camera relocalization via randomized ferns for keyframe encoding. *Transactions on Visualization and Computer Graphics,21*(5)

Graham, B., Engelcke, M., van der Maaten, L. (2018). 3D semantic segmentation with submanifold sparse convolutional networks. *Conference on Computer Vision and Pattern Recognition*.

Han, L., Zheng, T., Xu, L., Fang, L. (2020). OccuSeg: Occupancy-aware 3D instance segmentation. In: *Conference on Computer Vision and Pattern Recognition*.

He, K., Gkioxari, G., Dollár, P., Girshick, R. (2017). Mask R-CNN. In: *International Conference on Computer Vision*.

Herzig, R., Raboh, M., Chechik, G., Berant, J., Globerson, A. (2018). Mapping images to scene graphs with permutation-invariant structured prediction. In: *International Conference on Neural Information Processing Systems*.

Hou, J., Dai, A., Nießner, M. (2019). 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In: *Conference on Computer Vision and Pattern Recognition*.

Huang, J., Zhang, H., Yi, L., Funkhouser, T., Nießner, M., Guibas, L.J. (2019). TextureNet: Consistent local parametrizations for learning from high-resolution signals on meshes. In: *Conference on Computer Vision and Pattern Recognition*.

Huang, S., Qi, S., Xiao, Y., Zhu, Y., Wu, Y.N., Zhu, S.C. (2018). Cooperative holistic scene understanding: unifying 3D object, layout, and camera pose estimation. In: *International Conference on Neural Information Processing Systems*.

Izadinia, H., Shan, Q., Seitz, S.M. (2017). IM2CAD. In: *Conference on Computer Vision and Pattern Recognition*.

Jiang, C., Qi, S., Zhu, Y., Huang, S., Lin, J., Yu, L., Terzopoulos, D., Zhu, S. (2018). Configurable 3D scene synthesis and 2D image rendering with per-pixel ground truth using stochastic grammars. *International Journal of Computer Vision (IJCV)*.

Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.W., Jia, J. (2020) PointGroup: Dual-set point grouping for 3D instance segmentation. In: *Conference on Computer Vision and Pattern Recognition*.

Johnson, J., Gupta, A., Fei-Fei, L. (2018). Image generation from scene graphs. In: *Conference on Computer Vision and Pattern Recognition*.

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

Kirillov, A., He, K., Girshick, R., Rother, C., Dollar, P. (2019). Panoptic segmentation. In: *Conference on Computer Vision and Pattern Recognition*.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalanditis, Y., Li, L.J., Shamma, D.A., Bernstein, M., Fei-Fei, L. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision (IJCV)*.

Kulkarni, N., Misra, I., Tulsiani, S., Gupta, A. (2019). 3D-RelNet: Joint object and relational network for 3D prediction. *International Conference on Computer Vision*.

Kundu, A., Yin, X., Fathi, A., Ross, D., Brewington, B., Funkhouser, T., Pantofaru, C. (2020). Virtual multi-view fusion for 3D semantic segmentation. In: *European Conference on Computer Vision*.

Lahoud, J., Ghanem, B., Pollefeys, M., Oswald, M.R. (2019). 3D instance segmentation via multi-task metric learning. In: *International Conference on Computer Vision*.

Li, M., Gadi Patil, A., Xu, K., Chaudhuri, S., Khan, O., Shamir, A., Tu, C., Chen, B., Cohen-Or, D., Zhang, H. (2018a). GRAINS: Generative recursive autoencoders for indoor scenes. *Transactions on Graphics*.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B. (2018b). PointCNN: Convolution on X-transformed points. In: *Advances in Neural Information Processing Systems*.

Li, Y., Ouyang, W., Wang, X., Tang, X. (2017). ViP-CNN: Visual phrase guided convolutional neural network. In: *Conference on Computer Vision and Pattern Recognition*.

Li, Y., Ouyang, W., Zhou, B., Shi, J., Zhang, C., Wang, X. (2018c). Factorizable net: An efficient subgraph-based framework for scene graph generation. In: *European Conference on Computer Vision*.

Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P. (2014). Microsoft COCO: Common objects in context. In: *European Conference on Computer Vision*.

Liu, T., Chaudhuri, S., Kim, V., Huang, Q., Mitra, N., Funkhouser, T. (2014). Creating consistent scene graphs using a probabilistic grammar. *Transactions on Graphics*.

Liu, Y., Yi, L., Zhang, S., Fan, Q., Funkhouser, T.A., Dong, H. (2020). P4Contrast: Contrastive learning with pairs of point-pixel pairs for RGB-D scene understanding. *CoRR* **abs/2012.13089**.

Liu, Y., Zhang, D., Lu, G., & Ma, W. Y. (2007). A survey of content-based image retrieval with high-level semantics. *Pattern Recognition, 40*(1), 262–282.

Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L. (2016). Visual relationship detection with language priors. In: *European Conference on Computer Vision*.

Lu, G., Yan, Y., Kolagunda, A., Kambhamettu, C. (2016). A fast 3D indoor-localization approach based on video queries. In: *Multi-Media Modeling*, pp. 218–230.

Ma, R., Patil, A.G., Fisher, M., Li, M., Pirk, S., Hua, B.S., Yeung, S.K., Tong, X., Guibas, L., Zhang, H. (2018). Language-Driven Synthesis of 3D Scenes from Scene Databases. In: *SIGGRAPH Asia, Technical Papers*.

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research, 9,* 2579–2605.

Mittal, G., Agrawal, S., Agarwal, A., Mehta, S., Marwah, T. (2019). Interactive Image Generation Using Scene Graphs. *ICLR Deep Generative Models for Highly Structured Data Workshop*.

Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N., Guibas, L. (2019). StructureNet: Hierarchical Graph Networks for 3D Shape Generation. *Transactions on Graphics*.

Najibi, M., Lai, G., Kundu, A., Lu, Z., Rathod, V., Funkhouser, T., Pantofaru, C., Ross, D., Davis, L.S., Fathi, A. (2020). DOPS: Learning to Detect 3D Objects and Predict Their 3D Shapes. In: *Conference on Computer Vision and Pattern Recognition*.

Narita, G., Seno, T., Ishikawa, T., Kaji, Y. (2019). PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things. In: *International Conference on Intelligent Robots and Systems*, pp. 4205–4212. IEEE.

Nathan Silberman Derek Hoiem, P.K., Fergus, R. (2012). Indoor segmentation and support inference from RGBD images. In: *European Conference on Computer Vision*.

Newell, A., Deng, J. (2017). Pixels to graphs by associative embedding. In: *International Conference on Neural Information Processing Systems*.

Nie, Y., Han, X., Guo, S., Zheng, Y., Chang, J., Zhang, J.J. (2020). Total3DUnderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In: *Conference on Computer Vision and Pattern Recognition*.

Peyre, J., Laptev, I., Schmid, C., Sivic, J. (2017). Weakly-supervised learning of visual relations. In: *International Conference on Computer Vision*.

Qi, C.R., Litany, O., He, K., Guibas, L.J. (2019). Deep hough voting for 3D object detection in point clouds. In: *International Conference on Computer Vision*.

Qi, C.R., Su, H., Mo, K., Guibas, L. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. In: *Conference on Computer Vision and Pattern Recognition*.

Qi, C.R., Yi, L., Su, H., Guibas, L.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: *International Conference on Neural Information Processing Systems*.

Qi, M., Li, W., Yang, Z., Wang, Y., Luo, J. (2019). Attentive relational networks for mapping images to scene graphs. In: *Conference on Computer Vision and Pattern Recognition*.

Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In: *International Conference on Neural Information Processing Systems*.

Google Research, G. (2021). tf3d. https://github.com/google-research/google-research.

Rethage, D., Wald, J., Sturm, J., Navab, N., Tombari, F. (2018). Fully-convolutional point networks for large-scale point clouds. In: *European Conference on Computer Vision*.

Ronneberger, O., P.Fischer, Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, *LNCS*, vol. 9351, pp. 234–241. Springer.

Rosinol, A., Abate, M., Chang, Y., Carlone, L. (2020). Kimera: An open-source library for real-time metric-semantic localization and mapping. In: *International Conference on Robotics and Automation*, pp. 1689–1696.

Rosinol, A., Gupta, A., Abate, M., Shi, J., Carlone, L. (2020). 3D Dynamic scene graphs: Actionable spatial perception with places, objects, and humans. In: *Robotics: Science and Systems*.

Shi, Y., Chang, A.X., , Wu, Z., Savva, M., Xu, K. (2019). Hierarchy denoising recursive autoencoders for 3D scene layout prediction. In: *Conference on Computer Vision and Pattern Recognition*.

Sohn, K. (2016). Improved deep metric learning with multi-class N-pair loss objective. In: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (eds.) *International Conference on Neural Information Processing Systems*, vol. 29, pp. 1857–1865. Curran Associates, Inc.

Song, S., Lichtenberg, S., Xiao, J. (2015) SUN RGB-D: A RGB-D scene understanding benchmark suite. In: *Conference on Computer Vision and Pattern Recognition*.

Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.G. (2015). Multi-view convolutional neural networks for 3D shape recognition. In: *International Conference on Computer Vision*.

Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T. (2018). Pix3D: Dataset and methods for single-image 3D shape modeling. In: *Conference on Computer Vision and Pattern Recognition*.

Te, G., Hu, W., Zheng, A., Guo, Z (2018). RGCNN: Regularized graph CNN for point cloud segmentation. In: *International Conference on Multimedia*.

Teney, D., Liu, L., Van Den Hengel, A. (2017). Graph-structured representations for visual question answering. In: *Conference on Computer Vision and Pattern Recognition*.

Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J. (2019). KPConv: Flexible and deformable convolution for point clouds. In: *International Conference on Computer Vision*.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., et al. (2016). YFCC100M: The new data in multimedia research. *Communications of the ACM, 59*(2), 64–73.

Torii, A., Arandjelović, R., Sivic, J., Okutomi, M., Pajdla, T. (2015). 24/7 Place recognition by view synthesis. In: *Conference on Computer Vision and Pattern Recognition*. IEEE.

Wald, J., Avetisyan, A., Navab, N., Tombari, F., Nießner, M. (2019). RIO: 3D object instance re-localization in changing indoor environments. In: *International Conference on Computer Vision*.

Wald, J., Dhamo, H., Navab, N., Tombari, F. (2020a). Learning 3D semantic scene graphs from 3D indoor reconstructions. In: *Conference on Computer Vision and Pattern Recognition*.

Wald, J., Sattler, T., Golodetz, S., Cavallari, T., Tombari, F. (2020b). Beyond controlled environments: 3D camera re-localization in changing indoor scenes. In: *European Conference on Computer Vision*.

Wang, M., Lai, Y.K., Liang, Y., Martin, R.R., Hu, S.M. (2014). BiggerPicture: Data-driven image extrapolation using graph matching. *Transactions on Graphics*.

Xia, F., R. Zamir, A., He, Z.Y., Sax, A., Malik, J., Savarese, S. (2018). Gibson Env: Real-world perception for embodied agents. In: *Conference on Computer Vision and Pattern Recognition*.

Xu, D., Zhu, Y., Choy, C., Fei-Fei, L. (2017). Scene graph generation by iterative message passing. In: *Conference on Computer Vision and Pattern Recognition*.

Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D. (2018). Graph R-CNN for scene graph generation. In: *European Conference on Computer Vision*.

Yang, X., Tang, K., Zhang, H., Cai, J. (2019). Auto-encoding scene graphs for image captioning. In: *Conference on Computer Vision and Pattern Recognition*.

Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.J.: GSPN: Generative Shape Proposal Network for 3D Instance Segmentation in Point

Cloud. In: Conference on Computer Vision and Pattern Recognition (2019)
Zareian, A., Karaman, S., Chang, S.F. (2020). Bridging knowledge graphs to generate scene graphs. In: *European Conference on Computer Vision*.
Zellers, R., Yatskar, M., Thomson, S., Choi, Y. (2018). Neural motifs: Scene graph parsing with global context. In: *Conference on Computer Vision and Pattern Recognition*.
Zhang, J., Shih, K.J., Elgammal, A., Tao, A., Catanzaro, B. (2019). Graphical contrastive losses for scene graph parsing. In: *Conference on Computer Vision and Pattern Recognition*.
Zhao, Y., Birdal, T., Deng, H., Tombari, F. (2019). 3D Point capsule networks. In: *Conference on Computer Vision and Pattern Recognition*.
Zhao, Y., chun Zhu, S. (2011). Image parsing with stochastic scene grammar. In: *International Conference on Neural Information Processing Systems*.
Zhou, Y., Tuzel, O. (2017). VoxelNet: End-to-end learning for point cloud based 3D object detection.*Conference on Computer Vision and Pattern Recognition*.