Open3DSG: Open-Vocabulary 3D Scene Graphs from Point Clouds with Queryable Objects and Open-Set Relationships

Sebastian Koch[1,2,3] Narunas Vaskevicius[1,2] Mirco Colosi[2]
Pedro Hermosilla[4] Timo Ropinski[3]

[1]Bosch Center for Artificial Intelligence [2]Robert Bosch Corporate Research [3]University of Ulm [4] TU Vienna

kochsebastian.com/open3dsg

Abstract

Current approaches for 3D scene graph prediction rely on labeled datasets to train models for a fixed set of known object classes and relationship categories. We present Open3DSG, an alternative approach to learn 3D scene graph prediction in an open world without requiring labeled scene graph data. We co-embed the features from a 3D scene graph prediction backbone with the feature space of powerful open world 2D vision language foundation models. This enables us to predict 3D scene graphs from 3D point clouds in a zero-shot manner by querying object classes from an open vocabulary and predicting the inter-object relationships from a grounded LLM with scene graph features and queried object classes as context. Open3DSG is the first 3D point cloud method to predict not only explicit open-vocabulary object classes, but also open-set relationships that are not limited to a predefined label set, making it possible to express rare as well as specific objects and relationships in the predicted 3D scene graph. Our experiments show that Open3DSG is effective at predicting arbitrary object classes as well as their complex inter-object relationships describing spatial, supportive, semantic and comparative relationships.

1. Introduction

3D scene graphs are an emergent graph-based representation facilitating various 3D scene understanding tasks. In contrast to other more object-centric 3D scene representations, the key advantage of 3D scene graphs is the ability to also represent relationships between scene entities, such as for instance objects in a room. These relationships can be useful for a variety of different downstream tasks in computer vision or robotics, such as place recognition, change detection, task planning and more [1, 26, 34, 44, 53]. However, the exploitation of 3D scene graphs is limited by their availability.

Given their complexity and high-level abstraction, 3D scene graphs are hard to predict by learned models. The state-of-the-art (SOTA) methods for 3D scene graph prediction are limited to a fixed set of object and relationship labels provided by small-scale datasets. This reduces their effectiveness in downstream applications, which often require semantic reasoning on concepts extending beyond a rather narrow scope of training data. Furthermore, one of the most useful properties of scene graphs is their ability to represent relationships between scene entities. There are multiple ways of describing a relationship between two objects, e.g. spatial, comparative, semantic, etc. The relevance of the type of relationship is dictated by the downstream task. However, in a closed-set supervised training setting this choice is made and fixed in advance.

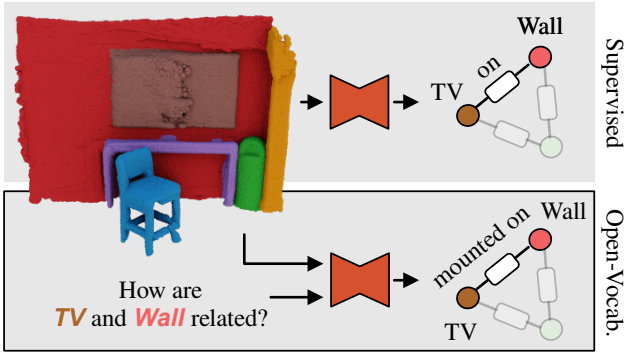


Figure 1. **Open3DSG.** We present Open3DSG the first approach for learning to predict open-vocabulary 3D scene graphs from 3D point clouds. The advantage of our method is that it can be queried and prompted for any instance in the scene, such as the *TV* and *Wall*, to predict fine-grained semantic descriptions of objects and relationships. By considering all instance pairs in the scene, we can reconstruct a complete explicit open-vocabulary 3D scene graph.

Open-vocabulary 3D scene understanding methods propose a solution towards these challenges by training a model not on a fixed label set but rather aligning the 3D model with 2D foundation models [14, 15, 18, 20, 31, 41]. By doing so, e.g. with foundation models such as CLIP [33], the 3D model can express nearly the same broad vocabulary

that these vision language models (VLMs) were trained on. However, while these 2D models are very capable of predicting single objects or higher-level concepts, they do not perform well in modeling compositional structures such as relationships or attributes [50, 52]. This limitation makes it challenging to adopt 2D VLMs for scene graph predictions where compositional relationships are the core part.

In this paper, we demonstrate that intuitive CLIP-like approaches are ill-suited for open-vocabulary relationship prediction. To this end, our key idea is to combine the advantages of VLMs with large language models (LLMs), that have proven to be better at understanding compositional concepts [16], to predict open-vocabulary 3D scene graphs. We highlight the following three contributions:

- We are the first to present a method to create an interactive graph representation of a scene from a 3D point cloud, which can be queried for objects and prompted for relationships during inference time.
- We show how such a representation can be converted into an explicit open-vocabulary 3D scene graph. Thus effectively proposing the first open-vocabulary scene graph prediction approach from 3D point cloud data.
- Our proposed approach shows promising results on the closed-set benchmark 3DSSG [44], proving success in modeling compositional concepts in an open-vocabulary manner.

2. Related Work

3D scene graph prediction. 3D scene graphs were first proposed by Armeni et al. [2] as a hierarchical structure to combine entities such as buildings, rooms, objects and cameras into a unified structure. Following their inception, subsequent works improved upon the estimation of such hierarchical 3D scene graphs for large-scale environments [17, 36, 37]. Other 3D scene graph approaches rather focus on predicting local semantic inter-object relationships and building a graph of objects [21, 44–48, 55, 58]. The applications of these 3D scene graphs are plentiful, with uses in aligning 3D scans [38], reconstructing and generating 3D scenes [9, 22], forecasting scene change [26], or even task planning over 3D scene graphs [1, 34]. However, none of these approaches consider the topic of open vocabulary in the context of 3D scene graphs. Cheng et al. are the first to model an implicit scene graph representation for planning in navigation tasks which they call OVSG [4] – an open-vocabulary 3D scene graph model – however they do not predict any open-vocabulary relationships from sensor data and are reliant on human descriptions which are encoded in the scene graph using a language model for open-vocabulary lookup and matching. Another approach to explore open-vocabulary 3D scene graphs is ConceptGraphs [13] which is concurrent work to ours. ConceptGraphs utilizes 2D VLMs and captioning models to predict scene graphs with queryable nodes and stored summarized image captions for edges. However, they do not provide extensive evaluations for their predicted scene graphs, limiting themselves to a qualitative evaluation of spatial relationships with Amazon Turk. We identify that the core difference of our approach to ConceptGraphs and OVSG, is that we learn to predict 3D scene graphs directly from raw point clouds, which brings numerous advantages such as being able to predict 3D scene graphs at test time without requiring inference from computationally expensive VLMs and when only 3D scans are available. We also predict explicit semantic relationships as part of our method and do not have to store multiple captions per edge that describe the relationship.

Open-vocabulary 3D scene understanding. The recent success of 2D vision language models as open-vocabulary methods such as CLIP [33], ALIGN [19], or ImageBind [12] have motivated the process of adapting these foundation models for 3D scene understanding tasks such as semantic/instance segmentation or 3D open-vocabulary detection. One of the earliest lines of approaches [14, 15, 57, 59] and also ConceptGraphs [13] explore annotation-free 3D recognition by combining CLIP with a 3D detection head using available RGB-D images with known poses. However, these approaches can suffer from inaccurate 2D-3D projections and occlusion artifacts. Furthermore, RGB-D images with known poses are not always available. Therefore, more recently approaches such as OpenScene [31], LERF [20] and others [18, 28, 41, 56] aim to distill the knowledge of those 2D vision language models into a 3D architecture with the advantage that these approaches do not rely on available 2D images when performing inference on 3D data. After the distillation, these approaches demonstrate impressive open-vocabulary results and unique abilities such as localizing rare objects in large 3D scans. However, their accuracy on closed vocabulary benchmarks still falls short of fully-supervised methods that are specifically trained on one dataset.

However, in contrast to our goal, none of these 3D scene understanding approaches has attempted to model 3D relationships which are hard to learn and distill based on their compositional nature.

Compositionality in vision-language models. While vision-language models show impressive performances in zero-shot image retrieval or image classification [12, 19, 33, 43, 54], they lack complex compositional understanding. Yuksekgonu et al. [52] and Yamada et al. [50] identified that contrastive vision-language pre-trained models such as CLIP [33] tend to collapse to a *bag-of-words* representation, which cannot disentangle multi-object concepts. To this end, a number of benchmarks have surfaced to examine the compositional reasoning capabilities of current vision language models [16, 29, 42, 52]. Yet, attempts to improve compositional understanding of contrastive vision-language pre-

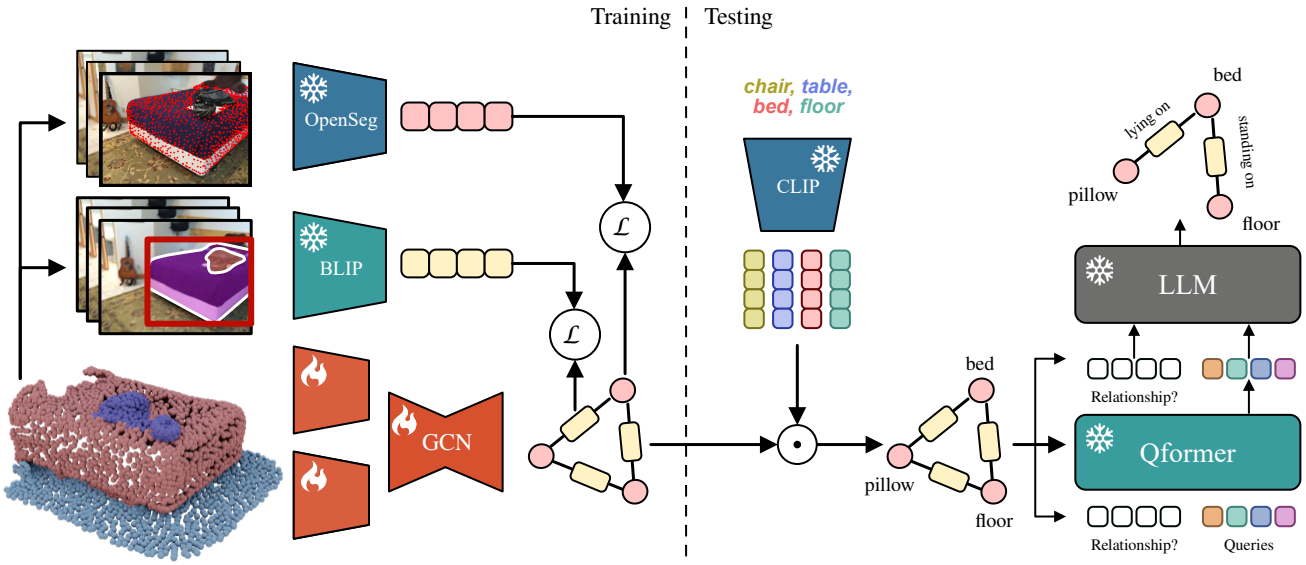


Figure 2. **Open3DSG overview.** Given a point cloud and RGB-D images with their poses, we distill the knowledge of two vision-language models into our GNN. The nodes are supervised by the embedding of OpenSeg [11] and the edges are supervised by the embedding of the InstructBLIP [7] vision encoder. At inference time, we first compute the cosine similarity between object queries encoded by CLIP [33] and our distilled 3D node features to infer the object classes. Then we use the edge embedding as well as the inferred object classes to predict relationships for pairs of objects using a Qformer & LLM from InstructBLIP.

trained models by utilizing additional data, prompting, models, losses and/or hard negatives [3, 10, 30, 35, 40, 52] yield only marginal improvements on these benchmarks. Furthermore, it is unclear whether these models achieve these improvements by actually acquiring compositional understanding or by exploiting biases in these benchmarks as indicated in [16].

Predicting relationships in a scene graph requires compositional understanding. In this paper, we approach this problem by shifting from a discriminative zero-shot approach to a generative approach using an LLM.

3. Method

The overall goal of our approach is to distill the knowledge of 2D vision-language models into a 3D graph neural network (GNN) to predict open-vocabulary 3D scene graphs in a 2-step process. We first construct an initial graph representation (Sec. 3.1), and in parallel, we extract vision-language features from aligned 2D images (Sec. 3.2). These features are then aligned to the ones extracted via the 3D GNN (Sec. 3.3), so that we can predict the same language-aligned features from 3D data only. At inference time, we perform a two-step prediction for objects and relationships. First, we predict object classes via a cosine similarity between the distilled features and open-vocabulary queries encoded by CLIP [33]. Then, we predict inter-object relationships by providing the learned relationship feature vector and the predicted object classes as context for a LLM (Sec. 3.4). An overview of our method is shown in Fig. 2.

3.1. Scene graph construction

Given a point cloud $\mathcal{P}$ of a scene with class-agnostic instance segmentation $\mathcal{M}$ provided by an off-the-shelf instance segmentation method such as Mask3D [39] or the dataset itself, we extract each object point cloud $\mathcal{P}_i$ containing instance i using the mask $\mathcal{M}_i$. Further, we extract point clouds $\mathcal{P}_{ij}$ of the instance pair $\langle i, j \rangle \in |\mathcal{M}| \times |\mathcal{M}|$, by selecting all points falling within the union of their respective bounding boxes $\mathcal{B}_{ij} = \mathcal{B}_i \cup \mathcal{B}_j$.

We construct an initial graph with node features ϕ_n and edge features ϕ_e. Each point set $\mathcal{P}_i$ is fed into a shared PointNet [32] to extract features for object nodes. Every point set $\mathcal{P}_{ij}$ is concatenated with a mask which is equal to 1 if the point corresponds to object i, 2 if the object corresponds to object j, and 0 otherwise. The concatenated feature vector is then fed into another shared PointNet to extract features for predicate edges.

The extracted node and edge features are then arranged as triplets $t_{ij} = \langle \phi_{n,i}, \phi_{e,ij}, \phi_{n,j} \rangle$ in a graph structure. This initial feature graph is passed into a GNN that processes the triplets t_{ij} and propagates the information through the graph

$$\phi_{n,i}^{(k)}, \phi_{e,ij}^{(k)}, \phi_{n,j}^{(k)} = \mathrm{G}(\phi_{n,i}, \phi_{e,ij}, \phi_{n,j}) \quad (1)$$

where $\mathrm{G}(\cdot)$ is a GNN and $\phi_{n,i}^{(k)}, \phi_{e,ij}^{(k)}, \phi_{n,j}^{(k)}$ are the refined features after k iterations of the GNN.

3.2. 2D feature extraction

Frame selection. The first step for aligning our 3D GNN with the 2D vision-language models is to extract 2D fea-

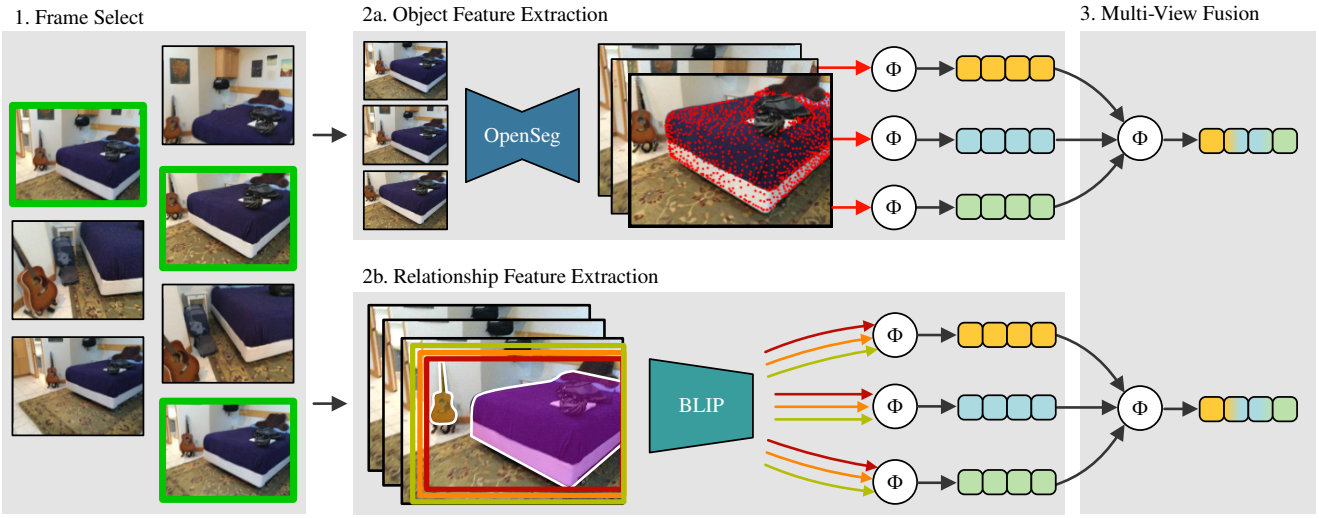


Figure 3. **Supervision feature extraction.** For each instance in the 3D point cloud, we select the top k frames for object and predicate supervision. For objects, we encode the frames using OpenSeg [11] and aggregate the computed features over the projected points. For predicates, we identify object pairs in the frame, crop the image at multiple scales and compute the image feature with the BLIP [7] image encoder. The features are aggregated over all crops. Finally, both object and predicate features are fused across the multiple views.

tures from the available 2D images and project them onto the constructed 3D feature graph. Selecting high-quality frames where the desired objects are visible is crucial to obtain robust and high-quality features. To achieve this, we utilize the same class-agnostic instance mask already used before to select a small subset of frames containing each pair of objects to be aligned with nodes and edges in the produced graph.

To estimate whether an object of instance i is visible in frame k, we use the camera's intrinsic $\mathbf{I}$ and extrinsics $(\mathbf{R}_k|\mathbf{t}_k)$ to project all points $\mathcal{P}_i$ onto the image plane of frame k. We define the projection of a single point p_i belonging to instance i projected into frame k with $\mathbf{p}_{ik} = (u, v, w)^T = \text{proj}_k(p_i) = \mathbf{I} \cdot (\mathbf{R}_k|\mathbf{t}_k) \cdot p_i$ where we represent p_i in homogeneous coordinates. We consider a point falling into the image plane if u/w falls in the interval $[0, W-1]$ and likewise if v/w falls in the interval $[0, H-1]$, where W and H are the image width and height dimension respectively. Furthermore, we discard each point $\mathbf{p}_{ik}$ that is occluded from the point of view of frame k, for which the inequality $w - d_k > t_{\text{occ}}$ is satisfied, where d_k is the measured depth for pixel (u, v), w is the estimated depth of the object instance for the same pixel, and t_{occ} is a fixed threshold hyperparameter. We denote the set of projected points passing the validity checks as $\mathbf{P}_{ik}$. Subsequently, we compute a visibility percentage as

$$\text{vis}(i, k) = \frac{|\mathbf{P}_{ik}|}{|\mathcal{P}_i|} \tag{2}$$

expressing the ratio of object points that are successfully projected onto the image frame. From the projected points, we can estimate their bounding box in the image as $box_{ik} = [\min_x(\mathbf{P}_{ik}), \min_y(\mathbf{P}_{ik}), \max_x(\mathbf{P}_{ik}), \max_y(\mathbf{P}_{ik})]$. Following this projection routine, each object instance i can be projected onto multiple frames. To ensure high-quality visual features, we choose a subset of high-quality frames by rejecting low-quality ones based on the condition

$$\text{vis}(i, k) > t_{\text{vis}} \vee \text{A}(box_{ik}) > t_{box} \tag{3}$$

where t_{vis} and t_{box} are hyperparameters, and $\text{A}(\cdot)$ computes the area of the given bounding box. We consider the bounding box area as an additional condition since large objects, such as *floor* or *wall*, might cover a huge portion of the scene, leading to a low visibility percentage for the current frame. In the end, we choose the top-k frames with the highest quality. For relationship frame selection, the process is similar, but we consider two object instances $\mathcal{P}_i$ and $\mathcal{P}_j$ simultaneously and a candidate frame has to satisfy Eq. (3) for both objects. The process of selecting both object and relationship frames is shown in Fig. 3 box 1.

Object feature computation. In order to achieve a coherent language-aligned object feature, we decide to leverage a VLM and collect the extracted features in a single representation. We choose OpenSeg [11] over CLIP [33], since the latter returns a global feature vector for the entire image or provided crop. This might also include extracted features regarding other parts of the image that are not relevant, while OpenSeg outputs pixel-wise embeddings. We provide an ablation for the advantages of using OpenSeg in Tab. 3. Thus, limiting the collected features to the ones related to the object improves our results. Consequently, from the selected top-k images we use OpenSeg to compute pixel-wise language-aligned features for object i and we compute a

global language-aligned embedding for the object by aggregating pixel-embeddings of the projected pixels $\mathbf{P}_{ik}$ using average-pooling. This step can be observed in box 2a. in Fig. 3.

Relationship feature computation. Similar to extracting per-object features, we also want to extract a global language-aligned feature embedding for relationships between two objects. Again we make use of VLM and decide to use InstructBLIP [7] since we identify in Sec. 2 that CLIP-like models are ill-suited to express compositional knowledge. Thus we use a BLIP-like model which visual feature embedding can be grounded with language to attend to the desired subjects. Given the top-k images where both object instance i and j are visible, we crop the image to the union of their respective bounding boxes $box_{ij}^{k} = box_{ik} \cup box_{jk}$. Then, we encode the crop at multiple scales using the BLIP image encoder from InstructBLIP to align the features with the InstructBLIP language model. Providing multiple scales of the same crop has been shown beneficial to provide important context information in [20] and [41]. The embedded crops are then aggregated using average-pooling. This step can be observed in Fig. 3 box 2b.

Feature aggregation. To provide a more robust and view-independent visual feature for objects and relationships, we average-pool all the global object features, and all the global relationship features previously extracted from each of the top-k frames. This results in two new global robust visual features: $\mathbf{f}_{o,i}^{\text{2D}}$ for object i, and $\mathbf{f}_{r,ij}^{\text{2D}}$ for relationship between objects i and j. The set of all the object features and relationship features are denoted by $\mathbf{F}_{o}^{\text{2D}} = \{\mathbf{f}_{o,1}^{\text{2D}}, ..., \mathbf{f}_{o,N}^{\text{2D}}\}$ and $\mathbf{F}_{r}^{\text{2D}} = \{\mathbf{f}_{r,1}^{\text{2D}}, ..., \mathbf{f}_{r,M}^{\text{2D}}\}$ respectively.

3.3. Graph distillation

The projected 2D object $\mathbf{F}_{o}^{\text{2D}}$ and predicate $\mathbf{F}_{r}^{\text{2D}}$ features can be directly used to predict 3D scene graphs if camera pose, depth and color images are available. However, in some circumstances, only 3D meshes or point clouds are provided. Furthermore, the fused 2D features can suffer from occlusions or prediction inconsistencies, resulting in noisy features. Therefore, we choose to distill the knowledge of the 2D vision-language models into a 3D network that operates on point clouds. To the best of our knowledge, the most suitable way to predict scene graph from 3D data is to leverage a GNN architecture.

Specifically, given a point cloud $\mathcal{P}$, we construct a graph $\mathcal{G}$ as defined in Sec. 3.1. We use the GNN architecture with message passing as proposed in [44] to output vision-language-aligned object node features as $\mathbf{F}_{o}^{\text{3D}} = \{\mathbf{f}_{o,1}^{\text{3D}}, ..., \mathbf{f}_{o,N}^{\text{3D}}\}$ with N being the number of nodes, and relationship edge encoding features as $\mathbf{F}_{r}^{\text{3D}} = \{\mathbf{f}_{r,1}^{\text{3D}}, ..., \mathbf{f}_{r,M}^{\text{3D}}\}$ with M being the number of edges.

To enforce the vision-language alignment for our 3D graph features, we define a training objective using a cosine similarity loss between the 2D vision-language features and the 3D features for nodes and edges

$$\mathcal{L} = 1 - \cos(\mathbf{F}_{o}^{\text{2D}}, \mathbf{F}_{o}^{\text{3D}}) + 1 - \cos(\mathbf{F}_{r}^{\text{2D}}, \mathbf{F}_{r}^{\text{3D}}). \quad (4)$$

Using this training objective, we distill the broad knowledge from the 2D vision-language foundation models into our 3D GNN. The process is depicted in Fig. 2.

After the distillation, the 3D graph features live in the same embedding space of the 2D vision-language foundation models.

3.4. Prediction and filtering

2D-3D Feature fusion. At inference time, we can perform open-vocabulary 3D scene graph prediction using only the distilled 3D features. However, if 2D images are available, we choose to fuse the 2D and 3D features in $\mathbf{f}_{o,i}^{\text{2D3D}}$ and $\mathbf{f}_{r,ij}^{\text{2D3D}}$ by average pooling the two for each feature pair 2D-3D. This is inspired by Peng et al. who observed in [31] that 2D features are beneficial to predict small objects, while 3D features yield good predictions for large objects with distinctive shapes. From this 2D-3D ensemble, we can infer node object classes and inter-object relationships in a two-step manner. First, we predict the object class of each node, and then using the inferred object classes we predict the relationship label on the edge between the classes.

Node prediction. As the first step to predict full open-vocabulary 3D scene graphs, we infer the object class label of each node from an open-vocabulary of arbitrary text prompts. These text prompts are encoded using the CLIP [33] text encoder to get the text features $\mathcal{T} = \{\mathbf{t}_1, ..., \mathbf{t}_N\}$, which are aligned with the OpenSeg [11] vision model and where N is the number of candidate classes. To classify the object class, we compute the cosine similarity between the candidate text prompts and the 2D-3D ensemble graph embedding and choose the class with the highest similarity score to the node feature:

$$\text{argmax}_n \cos(\mathbf{f}_{o,i}^{\text{2D3D}}, \mathbf{t}_n). \quad (5)$$

Relationship prediction. Following the prediction of the node classes in an open-vocabulary manner, the second step predicts relationships informed by the object predictions from the first step.

Contrastive vision-language models such as CLIP [33] have been shown to have a poor compositional understanding of the world [16, 29, 50, 52] resulting in limited accuracy when used for tasks such as relationship prediction. Thus, querying predicates for the scene graph edges in a similar manner as we have done for our node prediction will yield poor results. We provide experimental results to this hypothesis in the Tab. 1.

To solve this issue we exploit generative VLMs, which are grounded via a specific task. These models usually produce outputs that perform better on VQA benchmarks or benchmarks where it is required to have compositional reasoning [7, 23, 24]. However, a big drawback of deploying a generative approach is that restricting the output to a desired answer is not straightforward. To this end, InstructBLIP [7] is uniquely designed to give more output control using prompting. The InstructBLIP model consists of a vision transformer (ViT) encoder followed by a *Qformer*, which receives context from learnable tokens, a user prompt and the output of the ViT. The *Qformer* fuses and projects this information to the token space of a pre-trained LLM which is again conditioned on a user prompt.

We change the input to the *Qformer* such that instead of receiving the vision features from the ViT, we provide our 2D-3D distilled ensemble features from Sec. 3.3 coming from our graph neural network. To infer an accurate relationship grounded for a specific subject-object relation, we use the object class predictions from the first step to refine a template query to only output a relationship description for these two objects.

The output of the scene-conditioned *Qformer* is fed into the LLM which is prompted to output a relationship description for the subject-object pair in the graph, given the same conditioned query. This process is done in parallel for all edges in the scene graph to predict relationships for all subject-object pairs. The final result is an open-vocabulary 3D scene graph with open-vocabulary objects as well as open-vocabulary relationships.

4. Experiments

4.1. Experimental Setup

Datasets. The choice of training data is generally fixed for other 3D scene graph methods. The 3DSSG dataset [44] is at the time of writing this paper, the only dataset that provides semantic scene graph labels aligned with a 3D scene. This forces other methods [21, 44, 47, 55] to train and test on this rather small 3D dataset. In contrast, our method can be trained independently from scene graph labels on a 3D dataset that provides a 3D representation with posed 2D images, including their depth. While 3DSSG provides high-quality 3D point clouds and scene graphs, the provided portrait images have a low FOV, leading to a suboptimal 2D feature extraction. Therefore, we choose ScanNet [6], a similar indoor dataset, which provides image frames with acceptable FOVs and high-quality point clouds. However, since 3DSSG is the only dataset to provide ground truth scene graph labels, we evaluate our distilled model quantitatively on it.

Baseline methods. Given the challenging nature of open-vocabulary 3D scene graph prediction, our method is the first true open-vocabulary 3D scene graph method, that not only models open-vocabulary objects, but also open-vocabulary relationships from 3D point clouds. Therefore, no comparable method exists. As the first open-vocabulary 3D scene graph prediction method we compare against the first closed-vocabulary semantic 3D scene graph estimation method 3DSSG [44]. Further we compare against the current state-of-the-art [22, 47]. Additionally, we devise some open-vocabulary baseline methods for a fair comparison of our method. The first baseline is a naive CLIP-based approach, where we try to predict relationships directly with CLIP [33]. The second baseline we propose is a CLIP-based alternative to our method, where we predict objects and predicates in a 2-step manner directly from 2D images, querying first objects and then relationships using CLIP. This baseline is meant to highlight the advantage of using InstructBLIP for relationship prediction. We also evaluate the performance of NegCLIP [52] which is supposed to have improved compositional understanding. The third open-vocabulary baseline is similar to the concurrent work ConceptGraphs [13] and utilizes a caption-based approach directly from 2D images. We use OpenSeg [11] and BLIPv2 [24] to predict objects and their image captions, from which we extract objects and relationships for evaluation.

For further insights into our devised baselines, the reader is referred to our supplementary work.

Metrics. Designing metrics to quantitatively evaluate the capabilities of open-vocabulary methods is a current problem. So far, the best approach remains evaluating an open-vocabulary method on closed-vocabulary metrics. In our case, we choose the commonly used top-k recall metric (R@k) [27] for scene graphs. Following [44, 45, 49, 51, 55], we evaluate objects and predicates individually and relationships as subject-predicate-object triplets. Additionally, we provide a class-wise evaluation using the stricter mean recall metric (mR@k) [5].

Label mapping. To evaluate our method on a fixed-vocabulary benchmark, we provide object text queries from the class label set of 3DSSG, which comprises 160 classes. We compute the cosine similarity and choose the top-k predictions based on their cosine similarities. However, since we predict relationships in a generative manner, we cannot provide fixed queries for our relationship prediction. The LLM will output the most likely and best descriptive relationship given the context as well as subject and object. To map this to the fixed label set, we employ BERT [8], a small language model with well-structured word embeddings. It encodes the output of the LLM and the target relationship labels set and computes the cosine similarity from which we select the top-k most likely candidates. We reason that BERT has a well-structured word embedding space and is a good look-up approach to finding the most fitting syn-

onyms from the 27 relationship classes from the 3DSSG [44] dataset, which contains spatial, supportive, semantic, and comparative relationships labels.

4.2. Closed-set 3D scene graph prediction

Comparisons with fully-supervised and zero-shot methods. In Tab. 1 we compare our new zero-shot open-vocabulary 3D scene graph prediction approach with both fully-supervised as well as other zero-shot baselines on the 3DSSG [44] dataset. We outperform all our supervised baselines on object, predicate and relationship prediction. We demonstrate that a naive CLIP-based approach is ill-suited for relationship prediction, but also a two-step approach similar to our method by combining OpenSeg [11] and CLIP [33] or even NegCLIP [52] does not yield significant improvements. The caption-based approach also achieves considerably lower performances compared to our method. This is likely due to the poor quality of the 2D frames within the 3DSSG dataset, which negatively affects the caption-based approach which only uses 2D information for inference. In contrast, our approach uses a 2D-3D ensemble, where the distilled 3D features can compensate for the poor or missing 2D features.

Similar to other open-vocabulary approaches [31, 41], there is a noticeable gap to the state-of-the-art fully-supervised approaches. However, our zero-shot open-vocabulary approach is surprisingly competitive with the fully-supervised approach from a few years ago [44].

Impact of class occurrence. Fully-supervised methods are heavily biased by what they observe during training. Training samples of classes that are observed in a higher frequency are generally learned more effectively than rarer classes. In literature, there are multiple ways to alleviate this problem. Most scene graph methods [22, 44, 46] for instance, uses a focal loss [25] to solve the problem of class imbalance in the training set. As a zero-shot approach, our method is less susceptible to class imbalance. To evaluate this, we compare in Tab. 2 the mR@k recall of our first open-vocabulary method with recent 3D scene graph methods on the most common head classes, moderately common body classes and rare tail classes. We observe that while fully supervised methods demonstrate impressive accuracy on common object and predicate classes, their recall drops drastically for rare tail classes. In contrast, our zero-shot method reports consistent results across all classes, achieving on-par results with current fully supervised methods for all object and predicate classes averaged and outperforming the fully supervised methods on tail-end object classes by a considerable margin. This demonstrates the core advantage of our zero-shot open-vocabulary approach that it performs robustly on a wide variety of objects and predicates.

	Object		Predicate		Relationship	
Method	R@5	R@10	R@3	R@5	R@50	R@100
Fully-supervised						
3DSSG [44]	0.68	0.78	0.89	0.93	0.40	0.66
SGFN [47]	0.70	0.80	0.97	**0.99**	0.85	0.87
SGRec3D [22]	**0.80**	**0.87**	0.97	**0.99**	0.89	0.91
VL-SAT [46]	0.78	0.86	**0.98**	**0.99**	**0.90**	**0.93**
Zero-shot open-vocabulary						
CLIP (naive) [33]	0.35	0.42	0.09	0.19	0.02	0.04
OpenSeg [11] + CLIP [33]	0.38	0.45	0.10	0.23	0.05	0.07
OpenSeg [11] + NCLIP [52]	0.38	0.45	0.10	0.20	0.05	0.08
OpenSeg [11] + Cap. [24]	0.38	0.45	0.50	0.58	0.30	0.32
Open3DSG (Ours)	**0.57**	**0.68**	**0.63**	**0.70**	**0.64**	**0.66**

Table 1. **Closed-vocabulary evaluation on 3DSSG.** We compare our method with both zero-shot and fully-supervised baselines for 3D scene graph prediction. Overall, the zero-shot approaches perform worse than the fully-supervised methods. However, Open3DSG achieves comparable results to the first supervised 3D scene graph prediction method 3DSSG.

		Labels	Head	Body	Tail	All
Objects R@5	3DSSG [44]	10^5	0.88	0.45	0.06	0.30
	SGRec3D [22]	10^5	**0.92**	**0.78**	0.24	0.45
	VL-SAT [46]	10^5	**0.92**	0.73	0.31	**0.46**
	Open3DSG	0	0.60	0.50	**0.42**	0.45
Predicates R@3	3DSSG [44]	10^5	0.94	0.83	0.41	0.57
	SGRec3D [22]	10^5	0.97	**0.96**	**0.65**	0.69
	VL-SAT [46]	10^5	**0.99**	0.94	0.58	**0.75**
	Open3DSG	0	0.38	0.29	0.57	0.37

Table 2. **Frequency based class evaluation.** Here we compare the prediction performances for objects and predicates based on their frequency in the training set. Even though the fully-supervised approaches are trained specifically on this dataset, we can handle the less-common / long-tail classes much better.

4.3. Ablation studies

Is our knowledge distillation effective? In the top part of Tab. 3 we ablate the effectiveness of the feature distillation from the VLMs to our graph neural network. We compare results on 3DSSG [44] for our distilled 2D-3D ensemble method with a distilled 3D only method when posed images are not available and with a 2D only method where we directly use the 2D VLM features for 3D scene graph prediction. While the 2D method already shows good results, only when combining 2D and 3D features we reach the best performance of object and predicate prediction.

What if we have ground truth objects? Our relationship prediction using the LLM from InstructBLIP is conditioned on the queried objects from the OpenSeg embedding. Therefore, the correctness of the relationship prediction is influ-

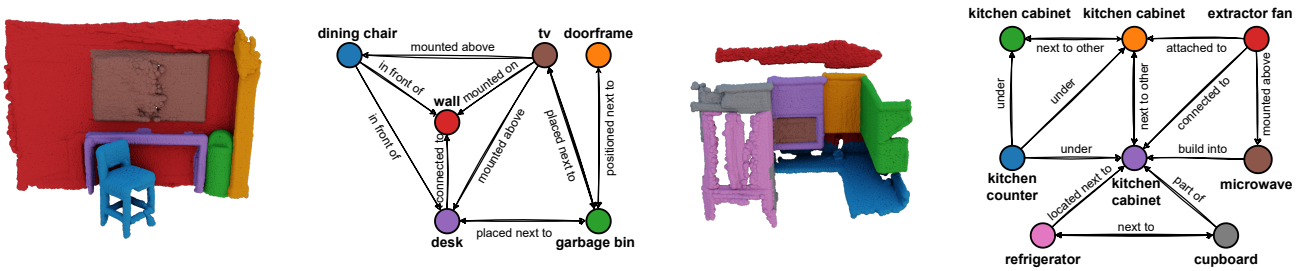


Figure 4. **Qualitative open-vocabulary 3D scene graph predictions.** We show the top-1 predictions on ScanNet [6] from Open3DSG. The nodes are queried using the 3DSSG [44] 160 class label set, while the edges are generated directly from the graph-conditioned LLM.

enced by the accuracy of the object querying. To evaluate both modules decoupled from each other, we provide the ground truth labels to InstructBLIP from which the LLM predicts the relationship. In the bottom part of Tab. 3, we observe that this has only a minimal impact, indicating that our method is robust towards slightly incorrectly predicted object nodes.

What if we use CLIP instead of OpenSeg? We choose OpenSeg [11] as our 2D object feature extractor. A popular alternative is CLIP. In the bottom part of Tab. 3 we show experimentally that using OpenSeg as the 2D object feature extractor yields better results compared to CLIP.

What if we learn predicates supervised? While the 3DSSG [44] contains over 160 annotated object classes, the number of categorized predicates is below 50 and most related works only evaluate on 27 or fewer distinct predicates [22, 44, 47, 55]. Therefore, given the comparably small vocabulary of predicates, we choose to fine-tune our model on 27 fixed predicate classes with only a few labels per class (~100). In the bottom part of Tab. 3, we observe that fine-tuning on 3DSSG improves predicate prediction with our model. Additionally, we observe synergy effects for object prediction. Hence, our VLM distillation training can also be an effective pre-training strategy when labels are scarce.

4.4. Qualitative Results

In Fig. 4, we provide qualitative results from our open-vocabulary 3D scene graph prediction approach for two different scenes from ScanNet [6]. We show the top-1 prediction for nodes and edges but filter edges where objects are further apart than 0.5m. The predicted object class labels are overall predicted correct and very specific, such as *microwave* or *dining chair*. The relationships between objects are generally correct as well with a diverse set of predicates such as *next to, attached to, under, above*. The advantages of our open-vocabulary prediction are especially good to see for the predictions such as "tv *mounted on* wall" or "microwave *build into* kitchen cabinet".

4.5. Limitations

The experiments conducted in this paper demonstrate the potential and advantages of open-vocabulary 3D scene graph methods. We observe that while predicting open-vocabulary objects shows great potential, predicting open-vocabulary relationships remains a challenging problem.

Furthermore, the evaluation setup for systematically evaluating open-vocabulary 3D scene graph methods still remains an open problem. While closed-vocabulary evaluations are valuable, they cannot highlight the huge potential of open-vocabulary methods such as ours.

	Object		Predicate	
	R@5	mR@5	R@3	mR@3
Open3DSG 2D	0.37	0.37	0.67	0.19
Open3DSG 3D	0.46	0.25	0.60	0.33
Open3DSG 2D-3D	0.57	0.45	0.63	0.37
Open3DSG 2D-3D w/ CLIP	0.48	0.32	0.59	0.32
Open3DSG 2D-3D + GT Objs	1.00	1.00	0.64	0.38
Open3DSG 2D-3D + Supv. Rels.	0.59	0.46	0.76	0.44

Table 3. **Ablation study.** 3D scene graph prediction with different input modalities, object VLM, privileged ground-truth information and supervised fine-tuning.

5. Conclusion

This paper introduces a new approach to learning semantic 3D scene graphs in an open-vocabulary manner from 3D point cloud data. Our method distills 2D VLMs into a 3D graph neural network thus creating a graph-based and language-aligned scene representation which can be queried and prompted to create an explicit open-vocabulary scene graph. To tackle the problem of lacking compositional knowledge in traditional VLMs, we split the relationship prediction into two steps, where we first query objects in a scene using CLIP and prompt relationships in a second step from the inferred objects using an LLM decoder. Our proposed approach shows promising results when evaluated on a closed-set benchmark and qualitative results confirm the open-vocabulary nature of our method. In future work, we see potential in improving relationship prediction even further to achieve even better and more reliable open-vocabulary 3D scene graph predictions that can be useful for many downstream tasks.

Acknowledgement This work was partly supported by the EU Horizon 2020 research and innovation program under grant agreement No. 101017274 (DARKO).

References

[1] Christopher Agia, Krishna Murthy Jatavallabhula, Mohamed Khodeir, Ondrej Miksik, Vibhav Vineet, Mustafa Mukadam, Liam Paull, and Florian Shkurti. Taskography: Evaluating robot task planning over large 3d scene graphs. In *Proceedings of the 5th Conference on Robot Learning*, pages 46–58. PMLR, 2022. 1, 2

[2] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[3] Paola Cascante-Bonilla, Khaled Shehada, James Seale Smith, Sivan Doveh, Donghyun Kim, Rameswar Panda, Gul Varol, Aude Oliva, Vicente Ordonez, Rogerio Feris, and Leonid Karlinsky. Going beyond nouns with vision & language models using synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20155–20165, 2023. 3

[4] Haonan Chang, Kowndinya Boyalakuntla, Shiyang Lu, Siwei Cai, Eric Pu Jing, Shreesh Keskar, Shijie Geng, Adeeb Abbas, Lifeng Zhou, Kostas Bekris, et al. Context-aware entity grounding with open-vocabulary 3d scene graphs. In *7th Annual Conference on Robot Learning*, 2023. 2

[5] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5828–5839, 2017. 6, 8

[7] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 3, 4, 5, 6

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 6

[9] Helisa Dhamo, Fabian Manhardt, Nassir Navab, and Federico Tombari. Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16352–16361, 2021. 2

[10] Sivan Doveh, Assaf Arbelle, Sivan Harary, Eli Schwartz, Roei Herzig, Raja Giryes, Rogerio Feris, Rameswar Panda, Shimon Ullman, and Leonid Karlinsky. Teaching structured vision & language concepts to vision & language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2657–2668, 2023. 3

[11] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022. 3, 4, 5, 6, 7, 8

[12] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15180–15190, 2023. 2

[13] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *arXiv*, 2023. 2, 6

[14] Huy Ha and Shuran Song. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. In *6th Annual Conference on Robot Learning*, 2022. 1, 2

[15] Deepti Hegde, Jeya Maria Jose Valanarasu, and Vishal Patel. Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2028–2038, 2023. 1, 2

[16] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *arXiv preprint arXiv:2306.14610*, 2023. 2, 3, 5

[17] N. Hughes, Y. Chang, and L. Carlone. Hydra: A real-time spatial perception system for 3D scene graph construction and optimization. In *Robotics: Science and Systems (RSS)*, 2022. 2

[18] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso Miguel de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. Conceptfusion: Open-set multimodal 3d mapping. *arXiv*, 2023. 1, 2

[19] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 2

[20] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. *arXiv preprint arXiv:2303.09553*, 2023. 1, 2, 5

[21] Sebastian Koch, Pedro Hermosilla, Narunas Vaskevicius, Mirco Colosi, and Timo Ropinski. Lang3dsg: Language-based contrastive pre-training for 3d scene graph prediction. *arXiv preprint arXiv:2310.16494*, 2023. 2, 6

[22] Sebastian Koch, Pedro Hermosilla, Narunas Vaskevicius, Mirco Colosi, and Timo Ropinski. Sgrec3d: Self-supervised

3d scene graph learning via object-level scene reconstruction. *arXiv preprint arXiv:2309.15702*, 2023. 2, 6, 7, 8

[23] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 6

[24] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 6, 7

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 7

[26] Samuel Looper, Javier Rodriguez-Puigvert, Roland Siegwart, Cesar Cadena, and Lukas Schmid. 3d vsg: Long-term semantic scene change prediction through 3d variable scene graphs. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8179–8186. IEEE, 2023. 1, 2

[27] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Computer Vision – ECCV 2016*, pages 852–869, Cham, 2016. Springer International Publishing. 6

[28] Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas Bekris. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In *7th Annual Conference on Robot Learning*, 2023. 2

[29] Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. Crepe: Can vision-language foundation models reason compositionally? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10910–10921, 2023. 2, 5

[30] Nihal V Nayak, Peilin Yu, and Stephen H Bach. Learning to compose soft prompts for compositional zero-shot learning. *arXiv preprint arXiv:2204.03574*, 2022. 3

[31] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 5, 7, 15

[32] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 4, 5, 6, 7

[34] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. *arXiv preprint arXiv:2307.06135*, 2023. 1, 2

[35] Arijit Ray, Filip Radenovic, Abhimanyu Dubey, Bryan A Plummer, Ranjay Krishna, and Kate Saenko. Cola: How to adapt vision-language models to compose objects localized with attributes? *arXiv preprint arXiv:2305.03689*, 2023. 3

[36] Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans. In *Robotics: Science and Systems (RSS)*, 2020. 2

[37] Antoni Rosinol, Andrew Violette, Marcus Abate, Nathan Hughes, Yun Chang, Jingnan Shi, Arjun Gupta, and Luca Carlone. Kimera: From slam to spatial perception with 3d dynamic scene graphs. *The International Journal of Robotics Research*, 40(12-14):1510–1546, 2021. 2

[38] Sayan Deb Sarkar, Ondrej Miksik, Marc Pollefeys, Daniel Barath, and Iro Armeni. Sgaligner : 3d scene alignment with scene graphs. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 2

[39] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. In *International Conference on Robotics and Automation (ICRA)*, 2023. 3

[40] Harman Singh, Pengchuan Zhang, Qifan Wang, Mengjiao Wang, Wenhan Xiong, Jingfei Du, and Yu Chen. Coarse-to-fine contrastive learning in image-text-graph space for improved vision-language compositionality. *arXiv preprint arXiv:2305.13812*, 2023. 3

[41] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 2, 5, 7

[42] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5238–5248, 2022. 2

[43] Michael Tschannen, Manoj Kumar, Andreas Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. Image captioners are scalable vision learners too. *arXiv preprint arXiv:2306.07915*, 2023. 2

[44] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 5, 6, 7, 8

[45] Johanna Wald, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs with instance embeddings. *International Journal of Computer Vision*, 130(3):630–651, 2022. 6

[46] Ziqin Wang, Bowen Cheng, Lichen Zhao, Dong Xu, Yang Tang, and Lu Sheng. Vl-sat: Visual-linguistic semantics assisted training for 3d semantic scene graph prediction in point cloud. *arXiv preprint arXiv:2303.14408*, 2023. 7

[47] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision*

and Pattern Recognition (CVPR), pages 7515–7525, 2021. 6, 7, 8

[48] Shun-Cheng Wu, Keisuke Tateno, Nassir Navab, and Federico Tombari. Incremental 3d semantic scene graph prediction from rgb sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5064–5074, 2023. 2

[49] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[50] Yutaro Yamada, Yingtian Tang, and Ilker Yildirim. When are lemons purple? the concept association bias of clip. *arXiv preprint arXiv:2212.12043*, 2022. 2, 5

[51] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 6

[52] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations*, 2023. 2, 3, 5, 6, 7

[53] Guangyao Zhai, Xiaoni Cai, Dianye Huang, Yan Di, Fabian Manhardt, Federico Tombari, Nassir Navab, and Benjamin Busam. Sg-bot: Object rearrangement via coarse-to-fine robotic imagination on scene graphs. *arXiv preprint arXiv:2309.12188*, 2023. 1

[54] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*, 2023. 2

[55] Chaoyi Zhang, Jianhui Yu, Yang Song, and Weidong Cai. Exploiting edge-oriented reasoning for 3d point-based scene graph analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9705–9715, 2021. 2, 6, 8

[56] Junbo Zhang, Runpei Dong, and Kaisheng Ma. Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip. *arXiv preprint arXiv:2303.04748*, 2023. 2

[57] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8552–8562, 2022. 2

[58] Shoulong Zhang, Shuai Li, Aimin Hao, and Hong Qin. Knowledge-inspired 3d scene graph prediction in point cloud. In *Advances in Neural Information Processing Systems*, pages 18620–18632. Curran Associates, Inc., 2021. 2

[59] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2639–2650, 2023. 2

Open3DSG: Open-Vocabulary 3D Scene Graphs from Point Clouds with Queryable Objects and Open-Set Relationships

Supplementary Material

In this **supplementary material**, we first provide additional implementation details in Sec. A. Next, we detail our design choices for our open-vocabulary 3D scene graph approach in Sec. B. In Sec. C we provide additional details on our proposed baselines. Next, we highlight the improved semantic understanding of our open-vocabulary method compared to fully-supervised methods in Sec. D and demonstrate the advantages with long-distance relationships compared to 2D-only open-vocabulary methods in Sec. E. We show unique applications of how our open-vocabulary 3D scene graphs can be used in Sec. F. Finally, we provide more qualitative results in Sec. G.

A. Implementation details

For our 3D graph backbone, we extract features from the point cloud using two PointNets that compute an initial 1024-dimensional feature vector for each node and edge. The graph features are refined using five layers of graph convolutions with message passing inspired by [44] and a hidden dimension of 2048. Finally, the node features are projected into the 768-dimensional CLIP space using a 5-layer MLP with ReLU activations and batch norm. The edge features are concatenated with the positional encoding from the BLIP-ViT and projected into the 1408-dimensional BLIP feature space using a 5-layer transformer architecture. The model is trained for 50 epochs using the Adam optimizer with weight decay, a learning rate of 5e-4, and a cyclic cosine-annealing learning rate scheduler. We use a batch size of 6 on a single Nvidia A100 GPU with mixed-precision.

During inference time, we use the pre-trained CLIP ViT-L/14@336 text encoder to encode the object queries and a pre-trained Vicuna 7B LLM model from Hugging Face [1] for predicate prediction. To query the CLIP text encoder we use object classes from the 160 class label set from 3DSSG [44], but we are not limited to those and can also query other arbitrary object classes or even concepts rather than discrete classes. To prompt the LLM we design an open-ended prompt to get the most open-vocabulary response: "Describe the relationship between [*object1*] and [*object2*]?". Here *object1* and *object2* are the object classes queried in the first step by CLIP. It is also possible to ask whether a specific relationship exists. However, we observe that providing more than five options confuses the LLM. To map the LLM predictions to the closed-vocabulary benchmark label set, we use the bert-base-uncased model from Hugging Face [2] with 768-dimensional feature embeddings.

[1] https://huggingface.co/Salesforce/instructblip-vicuna-7b

[2] https://huggingface.co/bert-base-uncased

B. Design choices

To succeed with distilling an open-vocabulary 3D scene graph method from 2D foundation models, we first study which model and which dataset is best suited for the distillation.

Compositionality pilot-study. Our approach highly depends on the knowledge encoded in the 2D vision-language model. However, Yuksekgonul et al. [52] and others [50] have demonstrated that current contrastive pre-trained vision-language models behave like bag-of-words models and have little understanding of compositionality. To evaluate whether a contrastively pre-trained VLM is suited for the distillation into our 3D scene graph model, we perform a pilot-study on a subset of the VL-Checklist Relation [52] benchmark. Differently from the evaluations conducted in [52], we do not evaluate whether the VLM can differentiate between the correct and incorrect relationship description but provide a set of queries where the VLM has to choose the most likely. This makes the task much harder for the VLM as the likelihood that the VLM picks the correct caption among the incorrect captions by random chance is much smaller. In the evaluation, we query the VLM using the query template "A relationship of a [*subject*] is [*predicate*] a [*object*]", where *subject* and *object* are fixed to the ground truth to solely evaluate the relationship understanding of the VLM. We report the top-1, top-2, and top-5 recall scores denoting whether the correct predicate was in the top-k highest similarity scores.

	top-1	top-3	top-5
Random chance	0.04	0.12	0.19
CLIP (ViT-L/14)	0.12	0.30	0.42
NegCLIP	0.14	0.35	0.48
SigLIP	0.11	0.27	0.37

Table A. **VL-Checklist Relation.** We evaluate the embedded relationship knowledge of the current state of contrastively pre-trained VLMs on an adapted benchmark from [52]. Results are reported for whether the VLM scores the correct predicate in the top-1, top-3, or top-5.

As expected, while CLIP [33], NegCLIP [52], and SigLIP [54] are exceptional zero-shot classifiers of objects, they cannot model inter-object relationships. The experimental evidence on a small controlled evaluation benchmark indi-

3RScan / 3DSSG

ScanNet

Figure A. **ScanNet vs. 3RScan.** We choose ScanNet over 3RScan / 3DSSG as a distillation dataset since the FOV of each frame is generally higher and more objects are visible in one frame.

cates that CLIP-like contrastively pre-trained VLMs do not have enough compositional knowledge about relationships that can be distilled into a 3D network. Therefore, in this paper, we choose to go beyond CLIP-like VLMs for relationship prediction and leverage a BLIP [7] vision encoder that can be projected into the token space of an LLM via a Qformer to predict relationships.

Distillation dataset. We choose to distill features on ScanNet [6] rather than 3RScan / 3DSSG [44], which we evaluate on. The reason for this is highlighted in Fig. A. Both datasets are indoor datasets depicting similar scenes. While ScanNet was recorded with an iPad with an attached depth sensor in landscape mode, 3RScan / 3DSSG was recorded with a Google Tango in portrait mode. The different recording setups result in entirely different vertical and horizontal field-of-views. We reason that to extract meaningful visual features representing the relationships between two objects, it is necessary that two objects are nearly fully visible in the same frame. This is rarely the case in 3RScan with its portrait setup. Therefore, we choose to use ScanNet for distillation as more of its frames depict more than one object.

C. Baselines

In addition to proposing a novel open-vocabulary 3D scene graph prediction method, we also propose several baselines. Here we provide further details on these baselines.

CLIP (naive). The most naive approach is to predict objects and predicates independently from each other directly using CLIP [33]. We select images for each object instance as well as images where a pair of objects is shown similar to the process in Sec. 3.2 and encode them using the CLIP image encoder. Then we build a fully-connected graph from the encoded features and query the nodes with object class labels and the edges with predicate class labels.

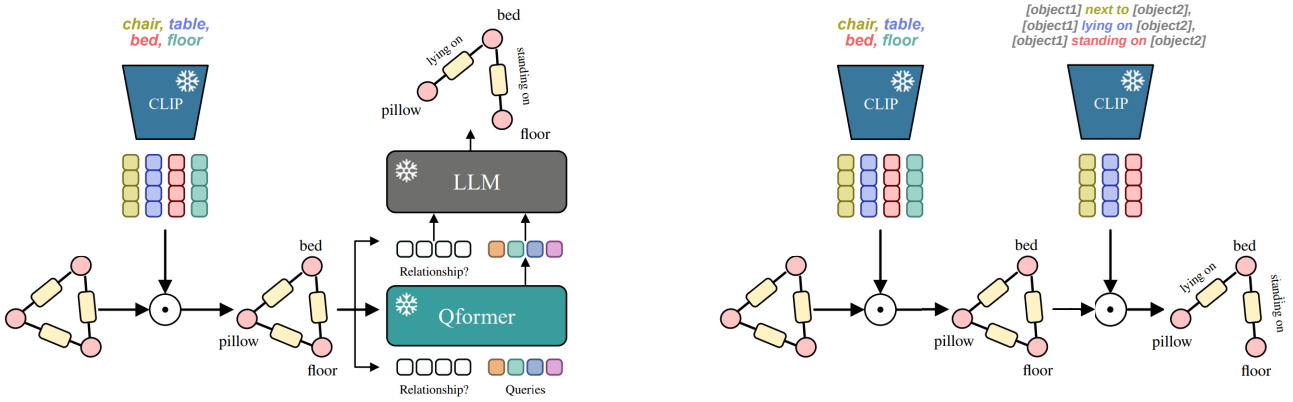
CLIP & NegCLIP. A more sophisticated approach using CLIP [33] or NegCLIP [52] is more similar to our two-step approach. The difference is shown in Fig. B. Here we also first build a fully-connected feature graph and predict object classes by querying the class of each node. Then we use the predicted objects as context to query full relationships in a second step using CLIP. Using the predicted objects as context improves results compared to the naive approach, nevertheless, the results fall short of our LLM approach due to the limited compositional knowledge of both CLIP and NegCLIP.

D. Improved semantics

While Tab. 1 in the main paper shows that our proposed open-vocabulary 3D scene graph method achieves overall worse performance compared to the current SOTA fully-supervised methods, Tab. 2 demonstrates the advantages of an open-vocabulary method, where we outperform the fully-supervised baselines on long-tail distribution classes. To give further insights into the benefits of our proposed open-vocabulary method, we provide scores on selected object and predicate classes in Tab. B. It shows that our open-

	3DSSG	SGRec3D	Open3DSG
Objects R@5			
cabinet / kitchen cabinet	0.39 / 0.33	0.67 / 0.87	0.39 / 0.94
chair / dining chair	0.98 / 0.00	0.94 / 0.00	0.48 / 1.00
table / bedside table	0.60 / 0.00	0.90 / 0.25	0.37 / 1.00
Predicates R@3			
standing on	0.73	0.95	0.86
covering	0.00	0.00	0.24
belonging to	0.48	0.65	0.91

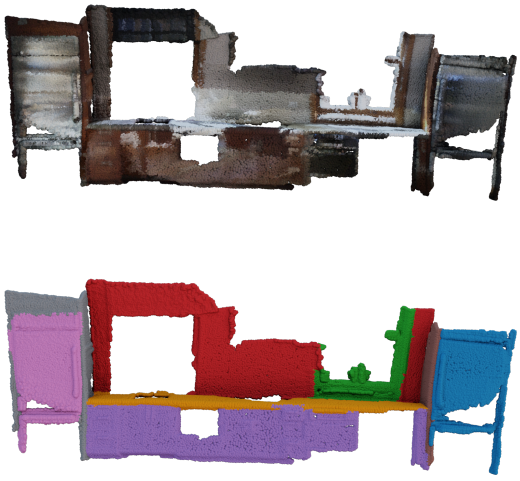
Table B. **Semantic awareness.** While fully-supervised methods such as 3DSSG [44] and SGRec3D [22] produce overall good results, their performance on difficult, rare, and semantically descriptive classes remains low. In contrast our open-vocabulary approach excels at semantically descriptive classes.



(a) Generative: LLM-based Relationships (Ours) **(b) Discriminative: CLIP-based Relationships (Baseline)**

Figure B. **Relationship prediction comparison.** We compare our generative relationship prediction approach using a prompted LLM (a), with a CLIP-based querying baseline (b) from Tab. 1. Due to the limited compositional knowledge of CLIP-like models, a discriminative approach where predicates can be directly queried performs much worse than a generative LLM-based approach.

vocabulary method outperforms the fully-supervised methods on very specific and semantically descriptive classes. For instance, for objects our network is better at differentiating a *chair* from a *dining chair* or a *table* from a *bedside table*. At the same time, fully-supervised methods, likely due to class imbalance during training, often only predict a generic class rather than the most specific class possible. This is similar for predicates. While the fully-supervised methods generally perform well on all predicates, highly semantic and specific predicates such as *covering* or *belonging to* are predicted less accurately. In contrast, our open-vocabulary method performs particularly well on semantic predicates such as *standing on*, *covering* or *belonging to*.



E. Long distance relationships

In Tab. 3, we provide an ablation for 3D scene graph prediction solely with 2D vision-language models. Only using 2D data performs worse than our learned 2D-3D ensemble approach.

While a prediction using 2D images is possible, a significant disadvantage of relying only on 2D data is that to predict a relationship between two objects, those two objects must be visible together in at least one frame. In contrast, our method does not have this limitation since it processes the 3D point cloud and can predict a relationship between two objects of arbitrary distance in a point cloud. Fig. C shows such two far-apart objects that are not close enough to appear in a shared frame, but still have a meaningful relationship detected by our method.

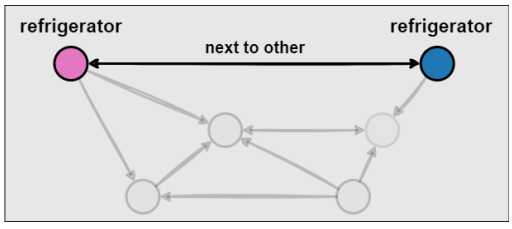


Figure C. **Long distance relationships.** In contrast to a 2D-only relationship prediction approach, which requires two objects to be visible in an image together, our 3D approach can predict relationships for two arbitrary far objects.

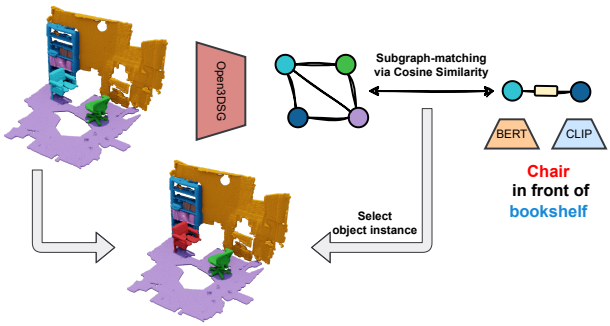


Figure D. **Application: Object localization via triplet description.** Using our open-vocabulary approach, we can localize object instances in the 3D point cloud given a relationship description of the object instance.

F. Applications

F.1. 3D Triplet localization

3D scene graphs are useful for various downstream computer vision or robotics tasks. In Fig. D we demonstrate one of those use cases uniquely suited to our language-aligned open-vocabulary 3D scene graphs. First, a 3D scene is encoded as an open-vocabulary 3D scene graph using our method. This representation is now queryable and promptable with an open vocabulary, making it a versatile tool for various scene understanding tasks. We demonstrate its usefulness for object localization in a 3D point cloud. Unlike other object localization methods [31], our goal is not to localize all objects of the same class but a specific instance that fits a relationship description. We encode a relationship description using the CLIP [33] and BERT [8] language encoders to generate a triplet feature representing the relationship. Then, we perform a subgraph-matching based on the cosine similarity of each triplet in the encoded scene graph with our target triplet feature. We select the triplet with the highest similarity score and reference it in the point cloud using the scene graph-point cloud alignment.

F.2. Material prediction

We present another application of zero-shot object attribute/material prediction, evaluated quantitatively in Fig. E. The material prediction can be performed without further training with the same querying strategy described in Sec. 3.4. Predicting attributes for each object further enriches the predicted 3D scene graph. We provide a top-1 accuracy metric comparison with OpenScene [31], a point cloud-based open-vocabulary method, on 3DSSG. Open3DSG outperforms OpenScene for most materials and also achieves a higher average accuracy for all classes. Note however that OpenScene predicts the material per point while we predict the material per instance.

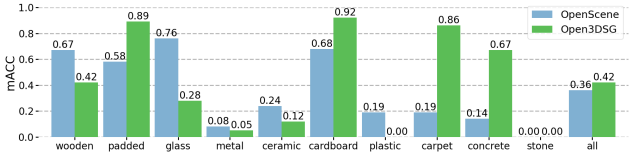


Figure E. **Application: Material prediction.** Using our open-vocabulary approach, we predict the material of objects without explicit training. We compare against OpenScene [31].

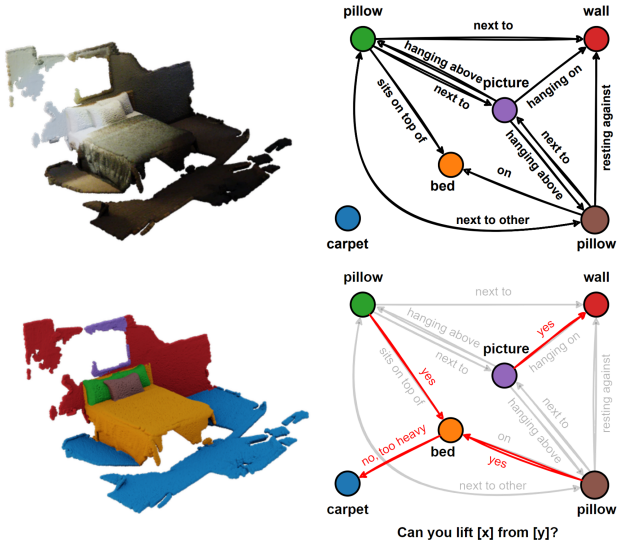


Figure F. **Application: Reasoning over object affordances.** Using our open-vocabulary approach, we reason about the affordances of objects by for instance prompting the LLM to output whether an object can be lifted from the other.

F.3. Reasoning over object affordances

A further application is the reasoning over scene-specific affordances using Open3DSG. Given the open-vocabulary representation computed by our method, we can prompt the LLM to predict affordances between objects. These affordances are grounded by the processed scene. In Fig. F, we demonstrate how Open3DSG can reason over which objects can be picked up by a human by prompting the LLM "Can you lift [x] from [y]". Our model correctly predicts that the pillows can be picked up from the bed while the bed would be too heavily to lift from the carpet.

G. Additional 3D scene graph predictions

In Fig. G, we provide additional 3D scene graph predictions on ScanNet [6]. Relationships for objects that are further apart than 0.5m are pruned for clarity in the visualization. Overall, the 3D scene graph predictions are correct and the advantages of an open-vocabulary method become especially apparent for rare and specific object classes such

as *computer desk* or precise relationship descriptions such as *tv mounted on wall*. But our open-vocabulary approach still has several limitations, such as overall low diversity in the predicted relationships. However, this limitation is not unique to our open-vocabulary method but also remains an issue with the current state of fully-supervised methods.

Nonetheless, our approach also has unique limitations, such as LLM-typical hallucinations like *computer desk (keyboard) connected via USB to monitor* or imperfect geometric understanding where two monitors are both predicted to be *to the left of* each other.

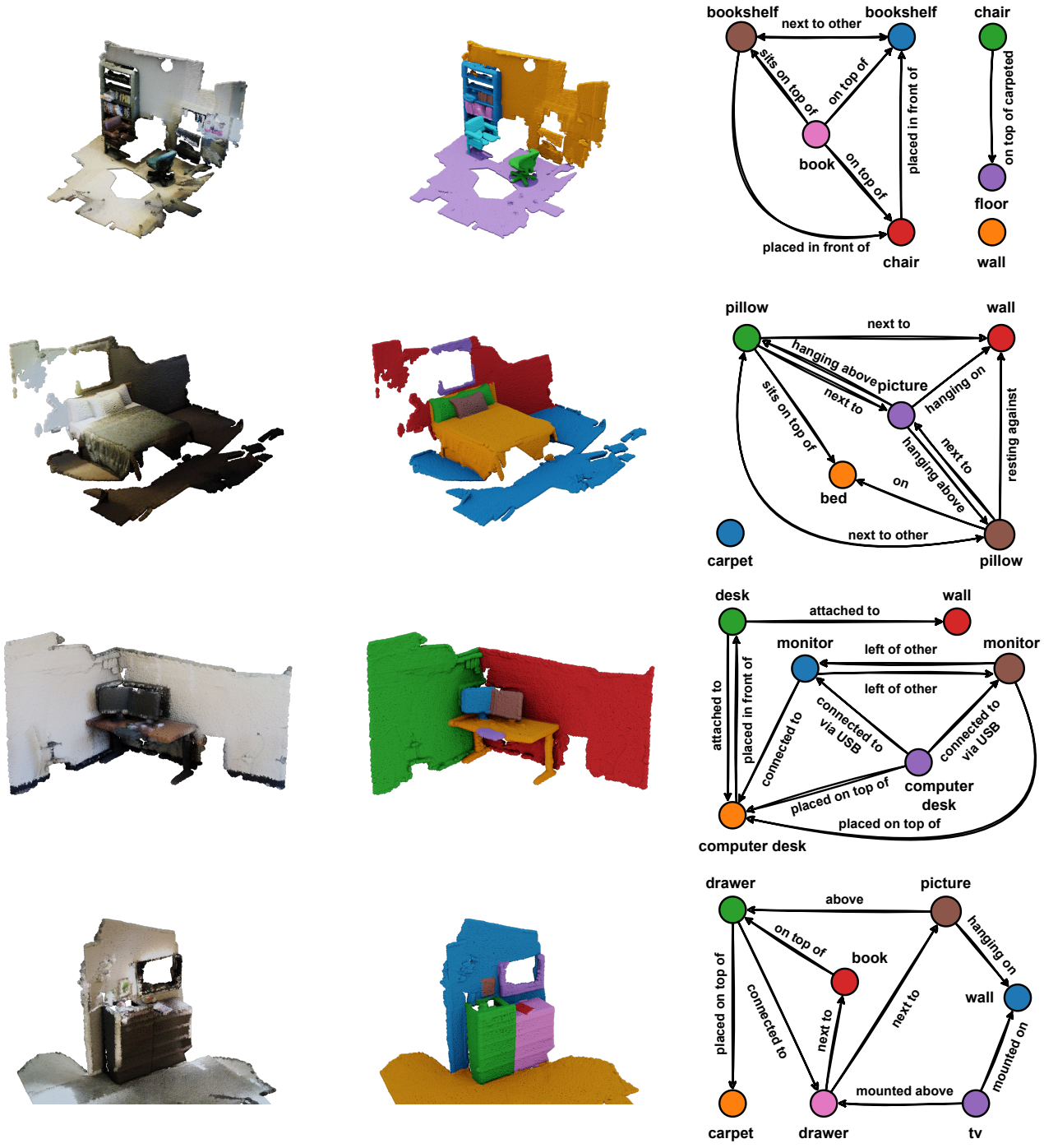


Figure G. **Qualitative open-vocabulary 3D scene graph predictions.** Left: Colored point cloud input; Middle: Class-agnostic mask; Right: Predicted open-vocabulary 3D scene graph.