

# Lang3DSG: Language-based contrastive pre-training for 3D Scene Graph prediction

Sebastian Koch<sup>1,2,3</sup>

Pedro Hermosilla<sup>4</sup>

Narunas Vaskevicius<sup>1,2</sup>

Mirco Colosi<sup>2</sup>

Timo Ropinski<sup>3</sup>

<sup>1</sup>Bosch Center for Artificial Intelligence    <sup>2</sup>Robert Bosch Corporate Research    <sup>3</sup>University of Ulm    <sup>4</sup>TU Vienna  
[kochsebastian.com/lang3dsg](http://kochsebastian.com/lang3dsg)

## Abstract

3D scene graphs are an emerging 3D scene representation, that models both the objects present in the scene as well as their relationships. However, learning 3D scene graphs is a challenging task because it requires not only object labels but also relationship annotations, which are very scarce in datasets. While it is widely accepted that pre-training is an effective approach to improve model performance in low data regimes, in this paper, we find that existing pre-training methods are ill-suited for 3D scene graphs. To solve this issue, we present the first language-based pre-training approach for 3D scene graphs, whereby we exploit the strong relationship between scene graphs and language. To this end, we leverage the language encoder of CLIP, a popular vision-language model, to distill its knowledge into our graph-based network. We formulate a contrastive pre-training, which aligns text embeddings of relationships (subject-predicate-object triplets) and predicted 3D graph features. Our method achieves state-of-the-art results on the main semantic 3D scene graph benchmark by showing improved effectiveness over pre-training baselines and outperforming all the existing fully supervised scene graph prediction methods by a significant margin. Furthermore, since our scene graph features are language-aligned, it allows us to query the language space of the features in a zero-shot manner. In this paper, we show an example of utilizing this property of the features to predict the room type of a scene without further training.

## 1. Introduction

In recent years, 3D scene graphs began to emerge as a new graph-based 3D scene representation that has seen a wide range of applications in computer vision and robotics [1, 11, 23, 36, 47, 53, 57, 58]. In this field, 3D scene graphs are powerful tools since they allow a compact and straightforward formulation to model both objects in the scene and their semantic relationship. In fact, they allow for

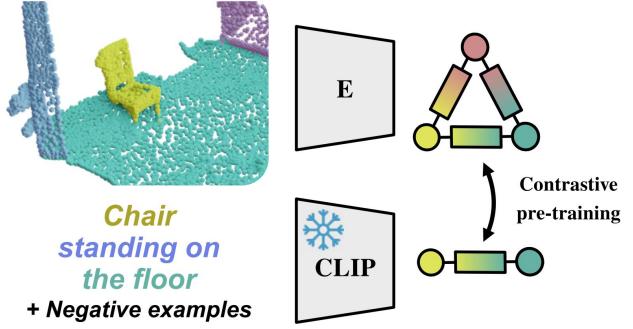


Figure 1. **Lang3DSG key idea.** Lang3DSG exploits the natural relatedness of language and 3D scene graphs by pre-training on contrastive language supervision.

a more high-level description of a 3D scene, as compared to conventional scene representations, such as 3D object detections or segmentation. Due to the encoded high-level information, 3D scene graphs can be used to solve various tasks, such as scene understanding or robot interaction, that conventional 3D models struggle with due to their limited understanding of scene semantics. However, predicting 3D scene graphs comes with several challenges, such as noisy and incomplete sensor data, as well as ambiguous object and relationship descriptions. Furthermore, while large training data sets are readily available for conventional 3D scene representations, training data for learning 3D scene graphs is much scarcer because relationships are harder to annotate.

A frequent approach to deal with the challenges of low data regimes is to facilitate pre-training methods, which allow utilizing the existing data more efficiently [6, 21]. While pre-training is popular in point cloud learning, our pre-training analysis for 3D scene graphs indicates that it is not sufficient for this particular case (see Sec. 3). While we could see an improvement in learning object predictions, pre-training did not lead to improved results when predicting object relations. We hypothesize that this lack of relationship understanding originates from the poor utiliza-

tion of context cues in the existing pre-training approaches. Such context information can be well represented and propagated using a graph neural network. Therefore, one of our key insights is to design a pre-training approach with a graph structure in mind.

Our second key insight is that scene graphs are inherently related to natural language. In language, a subject, a predicate and an object are the fundamental building blocks of a sentence and in scene graphs, the same triplet forms a relationship represented by two nodes and an edge. Recent advancements in large language models demonstrate their ability to abstract vast semantic knowledge in the embedding space. Aligning this embedding space with different modalities such as vision resulted in a paradigm shift in many scene understanding tasks [24, 41, 44]. Inspired by this progress, we propose to leverage the knowledge of the pre-trained language models for 3D scene graph prediction by formulating a language-based contrastive pre-training.

Thus, we present the following contributions:

- We propose a novel language-based contrastive pre-training for the downstream task of 3D scene graph prediction by exploiting the knowledge of language models.
- We show that our pre-training improves 3D scene graph prediction of a simple graph neural network to define a new SOTA by outperforming the existing fully-supervised methods.
- We demonstrate further capabilities of our approach, by exploiting the language-aligned scene graph features to predict room types in a zero-shot manner.

To the best of our knowledge, we are the first to investigate and propose an approach for 3D scene graph pre-training.

## 2. Related Work

**3D scene graph prediction.** A 3D scene graph models a scene as a graph by representing objects in the scene as nodes and relationships between objects as edges connecting two nodes. Scene graphs were first proposed in the 2D image domain by Johnson *et al.* [26] but have been adapted to 3D first by Armeni *et al.* [1], as a hierarchical structure to connect buildings, rooms and objects. Wald *et al.* [53] were the first to introduce a 3D semantic scene graph dataset focused on semantic relationships between objects. This dataset is built on top of the large-scale 3D dataset 3RScan [52] with over one thousand 3D scans. Based on this dataset, some subsequent works focused on expanding the common principles from 2D scene graph prediction to 3D [53, 67]. Other works focused on utilizing 3D scene graphs for image and scene retrieval [53], 3D scene reconstruction [28], generation and manipulation [11], alignment of 3D scene graphs as well as registration of scans [47] and change forecasting within the 3D scene [36]. Some approaches investigated the construction of 3D scene graphs during dynamic explorations of the scene with RGB-D [57]

or RGB cameras [58]. Finally, other works focused on the improvement of the 3D scene graph prediction with advanced message passing and graph convolutions [67], transformers [38] using pre-trained oracle models [56]. Instead, our approach focuses on a novel pre-training strategy leveraging the unique similarity between scene graphs and language.

**Pre-training for 3D scene understanding.** In the 2D domain, it is common practice to use backbone networks pre-trained on ImageNet [9], or pre-trained using other representation learning techniques [2, 18, 20, 40, 50, 60]. Inspired by the progress in the 2D scene understanding, recent works explore to adapt pre-training [5, 17, 46, 48, 55] on the 3D object-centric datasets such ShapeNet [3] and ModelNet [59]. However, Xie *et al.* [61] showed empirically that pre-training on datasets like ShapeNet [3] seem to be ineffective for scene-level 3D perception tasks such as 3D segmentation or 3D object detection. This motivated other works to investigate 3D representation learning based on self-supervised contrastive learning [6, 21, 22, 61, 69]. However, so far none of these works has considered 3D scene graph prediction as a downstream task for pre-training.

**Language-based 3D scene understanding.** In recent years natural language has become an important part in 2D scene understanding. The recent advances of large vision-language models such as CLIP [44], ALIGN [25] and follow-up works [30, 31, 66, 68] have made a paradigm shift in scene understanding from images enabling open-vocabulary object classification, detection and semantic segmentation [14, 16, 19, 29, 34]. This progress ignited interest in distilling the knowledge of 2D vision-language models into 3D representations [12, 27, 41, 63, 64]. Rozenberszki *et al.* [45] demonstrate the usefulness of semantically rich language features by grounding the 3D representation learning with language. They successfully use this idea for language-based pre-training to tackle the long-tail distribution problem in 3D semantic segmentation. Inspired by this we demonstrate how to leverage the vast semantic knowledge from language models in 3D scene graph pre-training.

## 3. 3D scene graph pre-training

### 3.1. Point cloud-based scene graphs pre-training

Latest improvements in pre-training methods focused on point clouds yield to positive results for various applications such as segmentation or object detection [6, 7, 21, 22, 45, 61, 69]. However, the domain of 3D scene graphs has received little attention in pre-training studies. To fill this gap, we conduct a pilot study presented in Tab. 1, exploring the effectiveness of point cloud pre-training for 3D scene graph prediction. In this study, we take two recent point

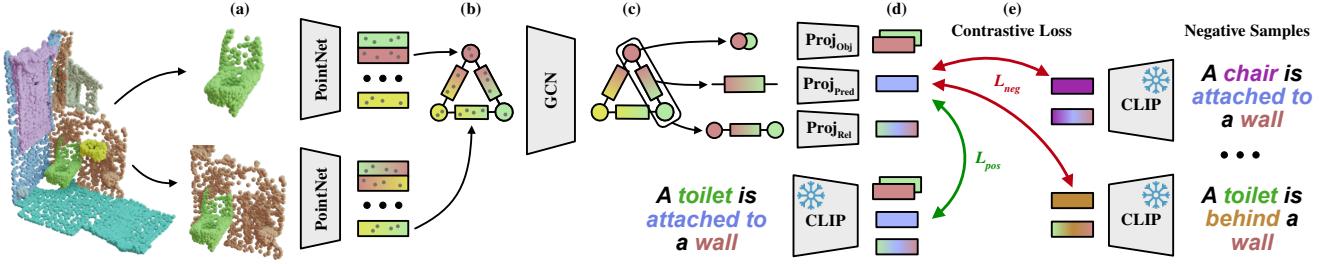


Figure 2. **Overview of our Lang3DSG pre-training framework.** Our method takes as input a class-agnostic segmented point cloud and extracts point sets of objects and pairs of objects (a). The point sets are passed into a PointNet backbone to construct an initial feature graph (b). Using a GCN, the features in the graph get refined (c) and node, edge and node-edge-node triplets are projected into the language feature space (d). Using a contrastive loss, we align the 3D graph features with the CLIP embeddings of the scene description (e).

cloud-based pre-training methods STRL [7] and DepthContrast [69] that open-sourced pre-trained PointNet++ [43] backbones trained on ScanNet [8] and fine-tune them for scene graph prediction by adding two prediction heads for objects and predicates. Additionally, we establish a simple graph-based baseline inspired by the work of Wald *et al.* [53] as a reference.

For evaluation, we employ top-k recall metrics for objects and predicates, where higher scores indicate better performance. Detailed information regarding the metrics used in this study can be found in Sec. 4.1. Our findings in Tab. 1 demonstrate that the pre-trained methods perform well on object prediction compared to the graph-based baseline. However, the predicate predictions do not show the same improvement with pre-training compared to the graph-based baseline. This indicates that having a graph-based backbone is essential for predicate prediction and that existing pre-training strategies are ineffective for scene graphs since they do not encode graph structures.

This result motivates us to design a pre-training tailored for scene graph prediction with a graph backbone in mind. In the following section, we will introduce our architectural setup as well as our 3D scene graph pre-training approach.

### 3.2. Language-based scene graph pre-training

Since a graph-based backbone seems important for scene graph prediction and the success of pre-training as concluded from our pilot study, we first define our graph-based backbone. We start by describing the feature extraction and graph construction methodology to embed a 3D scene into an initial feature graph. Then, we continue with a simple graph convolutional network (GCN). Furthermore, we specify the modules for contrastive self-supervised pre-training.

**Feature extraction and graph construction.** For the first step of our approach, we build an initial graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  from a generic scene  $s$ , where  $\mathcal{N}$  describes the set of objects and  $\mathcal{E}$  describes the relationships of the scene. This step involves generating a class-agnostic instance mask  $\mathcal{M}$  to extract instances  $i$  from the point cloud  $\mathcal{P}$  using the mask  $\mathcal{M}_i$

	Object		Predicate	
	R@5 $\uparrow$	mR@5 $\uparrow$	R@3 $\uparrow$	mR@3 $\uparrow$
Graph-baseline	0.63	0.30	0.94	0.57
STRL [7]	0.75 (+0.12)	0.35 (+0.05)	0.94 (-0.00)	0.50 (-0.07)
DepthContrast [69]	0.77 (+0.14)	0.36 (+0.06)	0.94 (-0.00)	0.51 (-0.06)

Table 1. **Comparison among point cloud-based 3D scene graphs pre-training.** We conduct a pilot study comparing existing point cloud-based pre-training studies with a graph baseline without pre-training.

from an off-the-shelf instance segmentation method such as Mask3D [49]. Those masks are used to extract from the scene point cloud  $\mathcal{P}$  a subset of points  $\mathcal{P}_i$  belonging to the object instance  $i$ , using the relative mask  $\mathcal{M}_i$ . From  $\mathcal{P}_i$  we also produce the object bounding box  $\mathcal{B}_i$  and discard any predicted class labels. Alternatively, when available, ground truth instance annotations can be used to extract the needed instances. Each point set  $\mathcal{P}_i$  including its color information is fed into a shared PointNet [42] to extract features  $\phi_n$  for each object node.

To generate edge features  $\phi_p$ , we use every instance pair  $\langle i, j \rangle \in \|\mathcal{M}\| \times \|\mathcal{M}\|$  to get the combined point set  $\mathcal{P}_{ij}$  belonging to the union of their respective bounding boxes  $\mathcal{B}_{ij} = \mathcal{B}_i \cup \mathcal{B}_j$ . Note that, we use the bounding box union to include also points  $p_k$  around both objects, which might introduce further contextual information. Before feeding every point set  $\mathcal{P}_{ij}$  and its color information into a second shared PointNet to extract the edge features  $\phi_p$ , we concatenate it with a point-wise mask equal to 1 if the point corresponds to object  $i$ , 2 if the object corresponds to object  $j$ , and 0 otherwise.

**Encoder.** From the extracted features  $\phi_n$  and  $\phi_p$ , we construct an initial feature graph where every node contains only local features describing the object and each edge contains only feature information about a pair of objects. However, this information lacks global scene context, which is necessary for predicting complex object relationships. To address this issue, we employ a graph convolutional network with message passing to propagate in-

formation through the graph such that each node and edge have contextual information about its nearest neighbors. To this purpose, we arrange the nodes and edges as triples  $t_{ij} = \langle \phi_{n,i}, \phi_{p_{ij}}, \phi_{n,j} \rangle$ . Every GCN layer  $l_g$  propagates information through the graph in three steps with a message passing procedure similar to [53]. First the triplet  $t_{ij}$  is fed into a MLP  $g_1(\cdot)$

$$\left( \psi_{n,i}^{(l_g)}, \phi_{p,ij}^{(l_g+1)}, \psi_{n,j}^{(l_g)} \right) = g_1 \left( \phi_{n,i}^{(l_g)}, \phi_{p,ij}^{(l_g)}, \phi_{n,j}^{(l_g)} \right) \quad (1)$$

where  $\phi_{p,ij}^{(l_g+1)}$  is the updated edge feature and  $\psi$  represents the incoming features for the nodes  $i$  and  $j$ . Using an aggregation function, the incoming node features are aggregated in a second step. We choose the average function as a suitable aggregation function

$$\rho_{n,i}^{(l_g)} = \frac{1}{N_i} \left( \sum_{k \in \mathcal{R}_i} \psi_{n,k}^{(l_g)} + \sum_{k \in \mathcal{R}_j} \psi_{n,k}^{(l_g)} \right) \quad (2)$$

where  $N_i$  denotes the number of edges connected to node  $i$ , and  $\mathcal{R}_i$  and  $\mathcal{R}_j$  are the set of nodes connected to node  $i$  and node  $j$  respectively.

Finally, the aggregated node features  $\rho_{n,i}^{(l_g)}$  are passed into a second MLP  $g_2(\cdot)$  adding a residual connection:

$$\phi_{n,i}^{(l_g+1)} = \phi_{n,i}^{(l_g)} + g_2 \left( \rho_{n,i}^{(l_g)} \right). \quad (3)$$

This process is repeated for  $k$  layers with which the receptive field of each node grows to finally get the refined features  $\phi_{n,i}^{(l_k)}, \phi_{p,ij}^{(l_k)}, \phi_{n,j}^{(l_k)}$  containing contextual information of their neighbors and beyond.

**Projection heads** Following Grill *et al.* [15], we propose three projector heads that project graph-based features into a high-dimensional feature space that matches the dimensionality  $D$  of our text encoder features.

The three 3-layers MLP project into the projection space the features generated after performing  $k$  iterations of graph convolutions. The first projector  $p_1(\cdot)$  projects only the node features, the second  $p_2(\cdot)$  only the edge features, while we feed the third projector  $p_3(\cdot)$  with the concatenated features for each triplet in the graph to get a singular triplet feature representing the entire relationship.

$$\begin{aligned} f_{n,i} &= p_1 \left( \phi_{n,i}^{(l_k)} \right), f_{p,ij} = p_2 \left( \phi_{p,ij}^{(l_k)} \right), \\ f_{triplet,ij} &= p_3 \left( \phi_{n,i}^{(l_k)} \oplus \phi_{p,ij}^{(l_k)} \oplus \phi_{n,j}^{(l_k)} \right). \end{aligned} \quad (4)$$

**Text Encoder** We leverage a pre-trained language model to map semantic relationship descriptions to text features. For this, we choose CLIP [44]. CLIP has been shown to have excellent visual understanding capabilities thanks to its vision-language pre-training. Note that our approach is agnostic to the choice of language model, but we found CLIP’s representations formed by its multi-modal training

well-suited to our relationship pre-training. During pre-training, we keep the text encoder frozen.

We provide three types of text prompts to our text encoder matching the three projected features from our 3D graph network. The first query contains only the object names, the second one provides only the predicate category to the text encoder, and the third query consists of the entire relationship in the form of “A scene of a [subject] is [predicate] a [object]” template.

Each text is then tokenized and encoded to their text embeddings  $f_1^t, f_2^t, \dots, f_n^t \in \mathbb{R}^D$  where  $D$  is the dimensionality of the text representation space. Note that we consider scene graphs where a pair of objects can share zero, one or multiple relationships. In case no relationship is present, we encode the predicate “and” as a neutral predicate to provide a target text embedding for the edge.

### 3.3. Language-based pre-training

**Contrastive loss.** For feature learning, we formulate a contrastive objective between the embeddings of the text model and the predicted 3D graph features by our network. We adopt the cosine similarity as our distance metric. This choice is inspired by CLIP [44] which has shown that it is a good distance metric for multi-modality contrastive learning as it provides more flexibility compared to  $l_1, l_2$  or MSE metrics:

$$\cos(f_i, f_{h(i)}^t) = \frac{f_i \cdot f_{h(i)}^t}{|f_i| \cdot |f_{h(i)}^t|} \quad (5)$$

where  $h(i)$  is the semantic text label for node/edge/triplet  $i$ .

During training, we differentiate between positive and negative samples. For the positive samples, our goal is to maximize the cosine similarity between our 3D graph feature encoding and the well-structured feature space of the text model. We do this by minimizing the following term

$$\mathcal{L}_{pos} = \sum_{i=1}^N \frac{1}{|K|} \sum_{j \in K} 1 - \cos(f_i, f_{h(j)}^t) \quad (6)$$

where  $N$  is the number of nodes/edges and  $K$  is the number of positive samples per node and edge. For objects  $|K| = 1$  since each node only maps to a singular object, but for edges  $|K| \geq 1$  because edges in 3D scene graphs can model more than one predicate/relationship.

Using the negative samples, our goal is to minimize the cosine similarity between our predicted 3D graph feature and the text feature

$$\mathcal{L}_{neg} = \sum_{i=1}^N \frac{1}{|M|} \sum_{j \in M} \max \left( 0, \cos(f_i, f_{h(j)}^t) - \tau \right) \quad (7)$$

where  $\tau$  is the negative margin and  $M \subset C$  consists of negative samples which are a set of labels different from  $i$  with  $C$  being the set of class label ids in the dataset.

In literature, negative samples are needed to prevent a collapse of the embedding space in most contrastive methods. However, in our case, since we are trying to distill knowledge from the text model it is not strictly necessary, but experimentally we found using negative samples improves the learned representation. We provide experimental evidence for improved representation learning using negatives in the supplementary. Our final language-3D graph pre-training loss is then:

$$\mathcal{L} = \mathcal{L}_{pos} + \lambda_{neg} \mathcal{L}_{neg} \quad (8)$$

**Hard Negative Samples.** We found that the selection of negative samples is very important for the quality of the learned representation with our contrastive learning. Thus, we pick the negative samples from the existing label set on our scene graph dataset. Both for objects and predicates we sample a random set of labels from the ones available in our dataset. However, for predicates, we make sure that the “and” predicate, which represents no relation, is always in the negative samples if it is not in the positive keys. By doing so, we enforce the boundary between existing and non-existing predicates. For negative relationship samples, we provide hard negatives by taking the true relationship as our source and modifying objects and predicates individually such that a negative relationship sample always shares one object or predicate with the positive sample (see right side of Fig. 2).

### 3.4. Scene graph fine-tuning

After pre-training using the text embeddings, the model needs to be fine-tuned using a supervised loss. This will allow to predict valid object class labels and predicate categories for the generated 3D scene graph. To do so, we discard the projectors and replace them with classification heads. We propose two classification heads, the first to classify the nodes in the graph and the second for predicting predicate labels for each edge. We train the classification heads using a cross-entropy loss  $L_{obj}$  for the object nodes and a per-class binary-cross-entropy loss  $L_{pred}$  for the predicates:

$$\mathcal{L} = \lambda_{obj} \mathcal{L}_{obj} + \lambda_{pred} \mathcal{L}_{pred} \quad (9)$$

**Implementation Details.** During the pre-training, we follow the approach described in the previous section. We choose *CLIP ViT-B/32* with the published weights from OpenAI [44] as our text model rather than larger CLIP models as a good compromise between the text understanding ability and inference speed. The text encoder of the *CLIP ViT-B/32* model provides features of dimensionality  $D = 512$ , which we match with our feature projectors. During the pre-training stage, we train our model for 50 epochs until convergence, with the Adam optimizer with a learning rate of 1e-3, linear learning rate decay and a batch size of 6.

We choose the number of negative samples  $M = 16$ , which we randomly sample from 160 object and 27 predicate categories. We set the negative loss weight to  $\lambda_{neg} = 1$  and use a negative margin of  $\tau = 0.5$ .

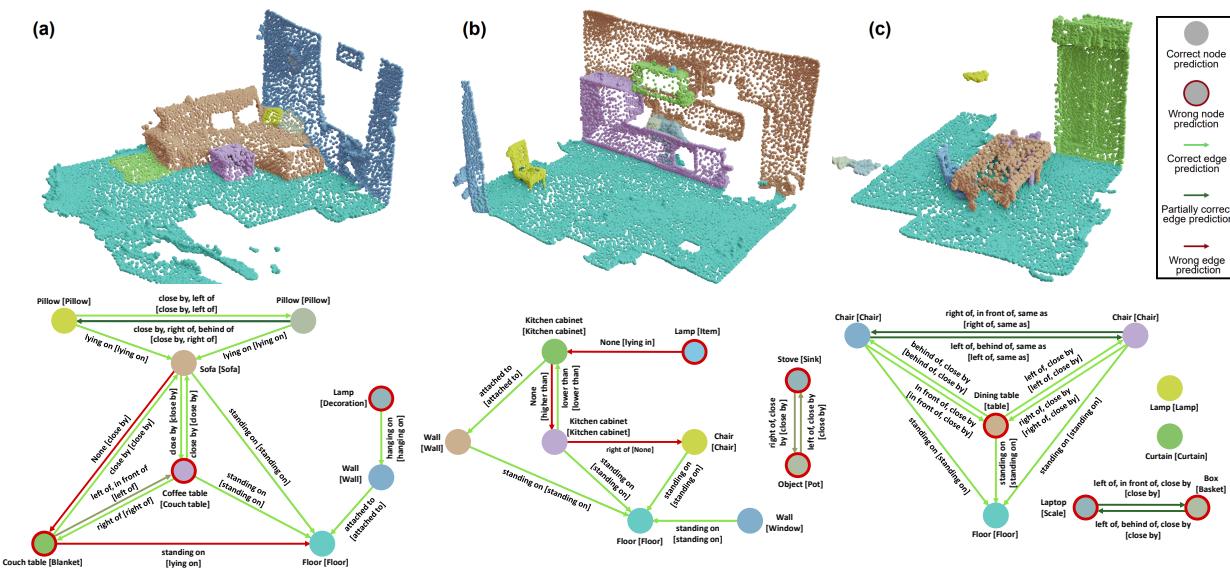
After pre-training, we proceed with fine-tuning the pre-trained 3D graph backbone using the available 3D scene graph labels. We use the same Adam optimizer with a learning rate of 1e-4, a batch size of 4 and train for 20 epochs. To ensure a balanced learning of object and predicate relationships, we set  $\lambda_{obj} = 0.1$  and  $\lambda_{pred} = 1.0$ ,

## 4. Experiments

### 4.1. Experimental Setup

**Dataset.** To validate the effects of our proposed 3D scene graph pre-training, we choose to evaluate our method after fine-tuning on publicly available 3D scene graphs datasets. The 3DSSG dataset [53] is at the time of writing this paper the only large-scale 3D dataset that provides semantic 3D scene graph labels with extensive relationship annotations. Another 3D scene graph dataset is [1], however, the scene graphs modeled in this dataset focus on hierarchical structuring and lack semantic relationship labels. In contrast, the most popular 3DSSG dataset provides semantic graph annotations with 160 object classes and 27 relationship categories over more than 1,000 indoor 3D point clouds reconstructions. These reconstructions are further subdivided into smaller scene graph splits with up to nine objects per split, resulting in more than 4,000 samples for training and evaluation. We follow Wald *et al.* [52] and use the original train/validation splits for training and evaluation. At the time of writing this paper, 3DSSG is the only large-scale 3D dataset that provides semantic 3D scene graph labels with extensive relationship annotations.

**Evaluation metrics.** For comparison with existing methods that predict 3D scene graphs without pre-training, we follow [53, 54, 62, 65, 67] and evaluate object and predicate predictions separately. Additionally, we jointly evaluate subject-predicate-object triples as relationships formed from two nodes in the graph and their enclosing edge. Since our approach predicts objects and predicates independently, we follow Yang *et al.* [65] and multiply the predicted object node and predicate edge probabilities to obtain a scored list of triplet predictions. We rank the triples by their score and use a top-k recall metric ( $R@k$ ) [37] for our main evaluation with existing works. The top-k recall metrics are used since one edge in the scene graph can represent multiple ground truth relationships. In our ablations, we additionally provide results on the less used top-k mean recall metric [4] for objects and predicates to cope with the high class imbalance present which affects the dataset, as shown in [53]. Using this metric gives better performance indication for under-represented classes in the dataset.



**Figure 3. 3D scene graph visualizations for 3DSSG scene splits.** Qualitative results of 3D scene graph prediction with Lang3DSG for three different example scenes. We visualize the top-1 object class prediction for each node and the predicates with a probability greater than 0.5 for each edge. Ground truth labels are shown in square brackets.

#### 4.2. 3D scene graph prediction

In Tab. 2 we report the results of our pre-trained model as described in Sec. 3. We compare against recent non-pretrained 3D scene graph methods (SGGPoint [67], 3DSSG [53], SGFN [57], Liu et al. [35]) and adapted 2D scene graph methods (MSDN [33], KERN [4], BGNN [32]). For the 2D scene graph methods, the 2D object detector was replaced by a PointNet-based feature extractor. Tab. 2 shows that we outperform all existing methods on the most used 3D scene graph prediction metrics. The exception is SGFN [57], which reports equal results in predicate predictions for R@5 and slightly better results for R@3. But especially for object classification, we achieve to outperform all other methods by a considerable margin with +7%/+4% improvements to our closest competitor SGFN [57]. For relationship prediction overall, we also outperform all other scene graph methods by a large margin for most methods and a considerable margin for SGFN. Regarding the close performance to SGFN, we hypothesize that for the predicates, we reached a saturation point on this dataset with these metrics. This is especially apparent for the R@5 metric, where we and SGFN both score 99%. In the following ablation, we try to overcome this saturation drawback by adopting the mR@k metric which has been less frequently used in literature. In Fig. 3, we provide qualitative examples of our 3D scene graph predictions for a diverse set of 3D scenes. We show the top-1 prediction for nodes and predicates that have a prediction score over 0.5. Our method is able to predict 3D scene graphs with a high accuracy. Nodes are generally classified correctly, but for a

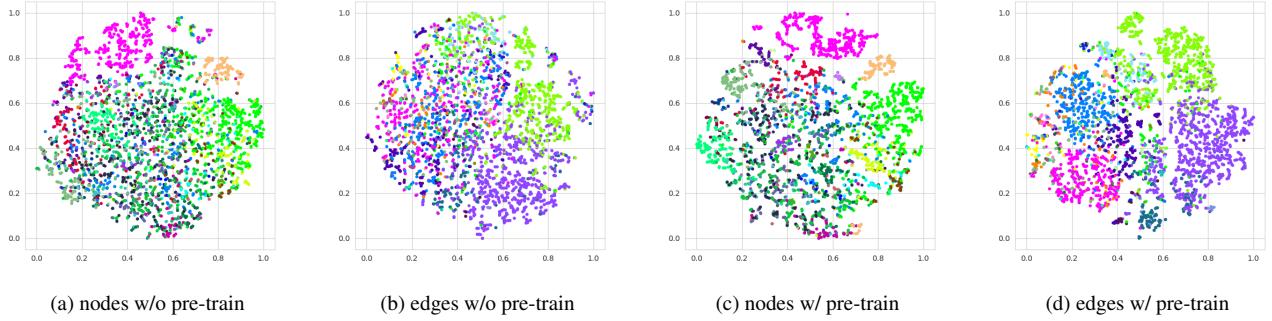
Method	Object		Predicate		Relationship	
	R@5	R@10	R@3	R@5	R@50	R@100
SGGPoint [67]	0.28	0.36	0.68	0.87	0.08	0.10
MSDN [33]	0.61	0.72	0.86	0.94	0.47	0.53
KERN [4]	0.67	0.77	0.83	0.96	0.51	0.58
BGNN [32]	0.71	0.82	0.87	0.94	0.55	0.60
3DSSG [53]	0.68	0.78	0.89	0.93	0.40	0.66
Liu <i>et al.</i> [35]	0.74	0.83	0.90	0.96	0.62	0.68
SGFN [57]	0.70	0.80	<b>0.97</b>	<b>0.99</b>	0.85	0.87
Ours	<b>0.77</b>	<b>0.84</b>	0.96	<b>0.99</b>	<b>0.87</b>	<b>0.89</b>

**Table 2. 3D scene graph prediction on 3DSSG.** Experimental results for 3D scene graph prediction on 3DSSG. We report the top-k recall values for object classification, predicate prediction as well as relationship prediction. For a fair comparison, all works use ground-truth class-agnostic instance segmentation.

few incorrectly classified objects, a label that nevertheless fits the context of the scene is chosen. Additionally, we observe that small objects with only a few relationships to others are more often misclassified, indicating that the graph nature is beneficial for object classification. Predicates are also predicted with a high accuracy, still, we observe that incorrectly predicted edges often coincide with misclassified nodes which propagate incorrect information.

### 4.3. Ablations

**Pre-training comparison.** In Tab. 3, we compare our pre-training with selected recent point cloud pre-training methods and a graph baseline that has the same graph backbone as our method but is not pre-trained using our approach. First, we want to highlight the large improvement between



**Figure 4. Learned latent representation.** We show a comparison of the learned representation for supervised training (a)-(b) and our CLIP-based pre-training (c)-(d). Using the language-based contrastive pre-training, our latent representation of objects and predicates is well-structured compared to the model with only supervised training.

	Object		Predicate	
	R@5	mR@5	R@3	mR@3
Graph-baseline (w/o pre-train)	0.63	0.30	0.94	0.57
STRRL [7]	0.75	0.35	0.94	0.50
DepthContrast [69]	<b>0.77</b>	0.36	0.94	0.51
Ours (no-graph)	0.74	0.37	0.94	0.60
<b>Ours</b>	<b>0.77</b>	<b>0.43</b>	<b>0.96</b>	<b>0.67</b>

**Table 3. Pre-training comparison.** A simple graph baseline outperforms existing point cloud-based pre-training methods on predicate prediction. Our novel pre-training shows high effectiveness outperforming existing pre-training approaches and graph baseline.

our method with pre-training and the graph baseline. Using our pre-training we are able to improve object classification by +14% for R@5 and +13% for the mR@5 metric. Predicate prediction improves similarly by +2% on the R@3 metric and +10% on the mR@3 metric. This indicates that our pre-training is especially effective for rare predicates. The improvements compared to point cloud-based pre-trainings are also large. While point cloud-based pre-trainings were able to improve object classification, but failed to improve predicate predictions for 3D scene graph predictions, our pre-training is effective for both tasks, outperforming STRRL [7] and DepthContrast [69] with a considerable margin of +7%/+8% on mR@5 object classification and +16%/+17% on mR@3 predicate prediction. We provide an additional ablation to our method with no graph backbone. This method demonstrates better performance than the non-pre-trained graph baseline, indicating the effectiveness of our pre-training. But compared to our method with graph-backbone, the model without a graph-backbone performs much worse, especially on predicate prediction, confirming our second takeaway from our pilot study that a graph-backbone is essential for scene graph prediction.

**Learned Representation.** In Fig. 4, we analyze the pre-trained representation space for objects and predicates by visualizing a t-SNE projection of the learned features. As a

	Object		Predicate	
	R@5	mR@5	R@3	mR@3
Relationship	0.75	0.38	0.94	0.63
Object + Predicate	0.77	0.41	<b>0.96</b>	0.66
Obj + Pred + Rel	<b>0.77</b>	<b>0.43</b>	<b>0.96</b>	<b>0.67</b>

**Table 4. Pre-training language supervision.** We ablate what language supervision is required for 3D scene graph pre-training. Combining relationship supervision with separate object and predicate supervision yields the best results.

comparison, we additionally provide feature projections for our method fine-tuned on 3DSSG without pre-training. Using our pre-training, we have learned a more structured 3D feature representation for objects and predicates, by having anchored our 3D graph features to the well-structured text embedding. Having this structured latent representation allows us to achieve significant improvements in the downstream task of 3D scene graph prediction when fine-tuning from this embedding space.

**Text supervision.** During pre-training, we provide three types of text embeddings as our supervision signal. First, the text embedding of objects, second the text embedding of predicates and third a composed embedding of the relationship in the form “A scene of a [subject] is [predicate] a [object]”. In Tab. 4, we ablate the effects for each target text embedding. Providing the composed relationship form already contains all the information about the objects and predicates, but we observe that only providing the relationship as supervision produces inferior results compared to providing individual embeddings for objects and predicates. We assume that this corresponds to the issue that CLIP and related models are not good at understanding complex compositional scenes [13, 39, 66]. In our supplementary, we provide further analyses to examine which parts in relationships text descriptions CLIP attends to. Combining object, predicate, and relationship embeddings during pre-training results in the best pre-trained model.

**Effect of the language model.** In Tab. 5, we consider al-

	Object		Predicate	
	R@5	mR@5	R@3	mR@3
CLIP ViT-B/32	0.77	0.43	0.96	0.62
CLIP ViT-L/14	0.77	0.43	0.96	0.62
CLIP ViT-L/14 (PCA)	0.76	0.41	0.96	0.61
BERT [10]	0.74	0.38	0.95	0.61

Table 5. **Text model ablation.** We ablate the effects of different text models and their embedding space. We find that CLIP works better than BERT, however different transformer sizes do not affect the final 3D scene graph prediction.

ternative language models to our selected *CLIP ViT-B/32* model. We consider BERT [10], which is a popular language model trained on large amounts of text data, rather than multi-modal image-text training from CLIP with its variant *bert\_untuned\_L-8\_H-512\_A-8* from [51]. We also choose *CLIP ViT-L/14* which is another CLIP model, but with a larger text transformer. The language features from *CLIP ViT-L/14* have a dimensionality of  $D = 768$ . We, therefore, have to adapt the architecture of our projectors to match the dimensionality of the language features from *CLIP ViT-L/14*. Additionally, we try to project the 768-dimensional features to 512 dimensions using PCA. We find CLIP’s rich embedding structure from the multi-model training produces better results compared to the text-based-only model BERT. However, we observe that using a larger CLIP model does not improve the pre-training effectiveness. The projected CLIP features using PCA produce slightly worse results. We assume this is because some information in the CLIP embedding gets lost when projecting its latent space to a lower dimension.

#### 4.4. Zero-shot room classification

In Fig. 5 we show one of many possible use cases granted by our language-aligned scene graph features. Given the unique property of our 3D scene graph method representing node and edge features in the well-structured CLIP representation space, we are able to query the graph in a zero-shot manner. One use case that leverages this representation, is querying the room type of the scene. Here, we exploit the fact that the features in our 3D scene graph are aligned with the language features from objects that have a high feature similarity with the room that they correspond to. For this, we encode candidate room types like *kitchen*, *bathroom*, *living room*, etc. using the CLIP language encoder to get the features  $f_1^q, f_2^q, \dots, f_n^q \in \mathbb{R}^D$  for each room query. Then we encode a scene  $s$  using our language-aligned graph backbone and use average-pooling to pool the features from all nodes in the graph  $f_{SG} = \varphi(f_{n,1}, f_{n,2}, \dots, f_{n,k})$ , where  $\varphi : \mathbb{R}^{K \times D \rightarrow D}$ . Finally, we compute the similarity cosine score between the pooled graph feature and the candidate room types and select the room type with the highest similarity score us-

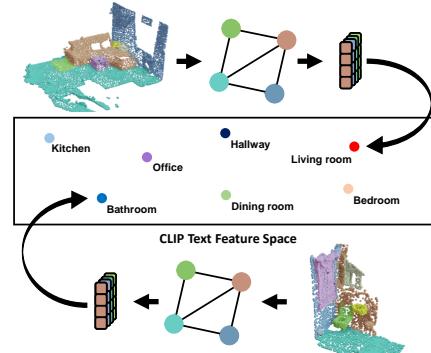


Figure 5. **Zero-shot room type classification.** Utilizing the language-aligned graph features we can classify the room type of a scene by similarity scoring the feature embedding of a room description and our 3D graph features. Two successful classification examples are shown.

ing  $\text{argmax}_q \{\cos(f_{SG}, f_n^q)\}$ . Fig. 5 shows qualitative results for two diverse examples for our room type prediction together with the abstracted methodology. We are able to successfully classify a bathroom and a living room by their language-aligned 3D graph features. Since this experiment is performed in a zero-shot manner, no quantitative results are available, however, we provide more predictions in our supplementary. This is one of many possible use cases made possible by our language-based pre-training and language-aligned latent graph features. Note, that this approach is different from recent open-vocabulary 3D understanding methods such as [41, 63]. Although we are able to exploit the relatedness of features in language space, we are not able to query unseen object and predicate classes since our language pre-training was done with a fixed vocabulary. We leave open-vocabulary 3D scene graph prediction to be investigated in future works.

## 5. Conclusion

In this paper, we find that recent pre-training approaches for 3D scene understanding on point clouds are ineffective for 3D scene graph prediction due to the inability to properly represent relations among objects. To this end, we introduce Lang3DSG, the first graph-based pre-training approach designed explicitly for 3D scene graphs, that exploits the tight connection between scene graphs and natural language. In the experimental study, we demonstrate that our method is more effective than existing pre-training baselines and achieves better performance than SOTA fully-supervised approaches on the 3DSSG dataset. Additionally, we show a zero-shot room type prediction use case based on exploiting language-aligned 3D graph features. In conclusion, our work contributes to 3D scene graph prediction, which is an important prerequisite for a wide range of downstream applications relying on accurate scene representations.

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

## References

- [1] 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. [1](#), [2](#), [5](#)
- [2] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. [2](#)
- [3] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. [2](#)
- [4] 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. [5](#), [6](#)
- [5] Ye Chen, Jinxian Liu, Bingbing Ni, Hang Wang, Jiancheng Yang, Ning Liu, Teng Li, and Qi Tian. Shape self-correction for unsupervised point cloud understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8382–8391, 2021. [2](#)
- [6] Yujin Chen, Matthias Nießner, and Angela Dai. 4dcontrast: Contrastive learning with dynamic correspondences for 3d scene understanding. In *Computer Vision – ECCV 2022*, pages 543–560, Cham, 2022. Springer Nature Switzerland. [1](#), [2](#)
- [7] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [2](#), [3](#), [7](#)
- [8] 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. [3](#)
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. [2](#)
- [10] 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. [8](#)
- [11] 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. [1](#), [2](#)
- [12] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Pla: Language-driven open-vocabulary 3d scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7010–7019, 2023. [2](#)
- [13] 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. [7](#)
- [14] 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. [2](#)
- [15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. [4](#)
- [16] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations*, 2022. [2](#)
- [17] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. [2](#)
- [18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [2](#)
- [19] Sunan He, Taian Guo, Tao Dai, Ruizhi Qiao, Xiujun Shu, Bo Ren, and Shu-Tao Xia. Open-vocabulary multi-label classification via multi-modal knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 808–816, 2023. [2](#)
- [20] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020. [2](#)
- [21] Ji Hou, Benjamin Graham, Matthias Niessner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15587–15597, 2021. [1](#), [2](#)
- [22] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6535–6545, 2021. [2](#)
- [23] N. Hughes, Y. Chang, and L. Carlone. Hydra: A real-time

- spatial perception system for 3D scene graph construction and optimization. 2022. 1
- [24] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Osama, 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. 2
- [25] 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 *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 2
- [26] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2
- [27] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. *arXiv preprint arXiv:2303.09553*, 2023. 2
- [28] 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
- [29] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 2
- [30] 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. 2
- [31] Liunian Harold Li\*, Pengchuan Zhang\*, Haotian Zhang\*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, 2022. 2
- [32] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11109–11119, 2021. 6
- [33] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 6
- [34] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7061–7070, 2023. 2
- [35] Yuanyuan Liu, Chengjiang Long, Zhaoxuan Zhang, Bokai Liu, Qiang Zhang, Baocai Yin, and Xin Yang. Explore contextual information for 3d scene graph generation. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–13, 2022. 6
- [36] 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. *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 1, 2
- [37] 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. 5
- [38] Changsheng Lv, Mengshi Qi, Xia Li, Zhengyuan Yang, and Huadong Ma. Revisiting transformer for point cloud-based 3d scene graph generation. *arXiv preprint arXiv:2303.11048*, 2023. 2
- [39] 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. 7
- [40] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2
- [41] 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. 2, 8
- [42] 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
- [43] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 3
- [44] 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. 2, 4, 5
- [45] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 125–141. Springer, 2022. 2
- [46] Aditya Sanghi. Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 626–642. Springer, 2020. 2
- [47] 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. 1, 2
- [48] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. *Advances in Neural Information Processing Systems*, 32, 2019. 2
- [49] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. 2023. 3
- [50] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020. 2
- [51] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*, 2019. 8
- [52] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Niessner. Rio: 3d object instance re-localization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2, 5
- [53] 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, 3, 4, 5, 6
- [54] 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. 5
- [55] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J. Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9782–9792, 2021. 2
- [56] 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. 2
- [57] 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. 1, 2, 6
- [58] 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. 1, 2
- [59] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Lin-guang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2
- [60] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [61] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision – ECCV 2020*, pages 574–591, Cham, 2020. Springer International Publishing. 2
- [62] 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. 5
- [63] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1189, 2023. 2, 8
- [64] Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding, 2023. 2
- [65] 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. 5
- [66] 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, 7
- [67] 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, 5, 6
- [68] Haotian\* Zhang, Pengchuan\* Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *arXiv preprint arXiv:2206.05836*, 2022. 2
- [69] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10252–10263, 2021. 2, 3, 7

# Lang3DSG: Language-based contrastive pre-training for 3D Scene Graph prediction

## Supplementary Material

This document supplements our work *Lang3DSG: Language-based contrastive pre-training for 3D scene graph prediction* by providing (i) reproducibility information on our implementation and architecture (Sec. 6), (ii) details on the importance of negative samples during training (Sec. 7), (iii) investigations into the understanding capabilities of CLIP for compositional scene descriptions (Sec. 8), (iv) additional 3D scene graph generations from diverse scenes (Sec. 9), (v) additional examples of zero-shot room type classification using our language-aligned features (Sec. 10),

## 6. Reproducibility

Our encoder consists of two PointNets which pass features of size 256 to a 4-layer GCN, where  $g1(\cdot)$  and  $g2(\cdot)$  are composed of a linear layer followed by a ReLU activation. The projectors are 3-layer MLPs with ReLU activation in the first two layers and feature dimensions of [256, 1024, 512] for CLIP ViT-B/32 and BERT and [256, 1024, 768] for CLIP ViT-V/14. During fine-tuning, we replace the projectors with object and predicate prediction MLPs consisting of 3 linear layers with feature dimensions [256, 512,  $class\_num$ ] with batch normalization and ReLU activation.

The training is performed on 1 NVIDIA A100 GPU with 80 GB memory.

## 7. Role of negatives during pre-training

In Sec. 3 of the main paper, we describe our contrastive pre-training. We use a cosine similarity loss to distill the knowledge of CLIP into our 3D graph model. For this, we differentiate between positive cases where our goal is to maximize the cosine similarity (see Eq. 6) and negative cases where our goal is to minimize the cosine similarity (see Eq. 7). This formulation is adapted from classical contrastive representation learning, where negative samples are needed to prevent a collapse of the latent representation. However, since we are trying to distill the knowledge directly from CLIP, negative samples are not strictly necessary. However, in Fig. 6, we show the difference in the learned latent embedding with and without negatives. It is important to note that both training with negatives and without negatives produce a structured latent embedding. However, we find that the latent representation for individual classes trained without negatives is much more mixed. This mix of class embeddings leads to a reduced effect of

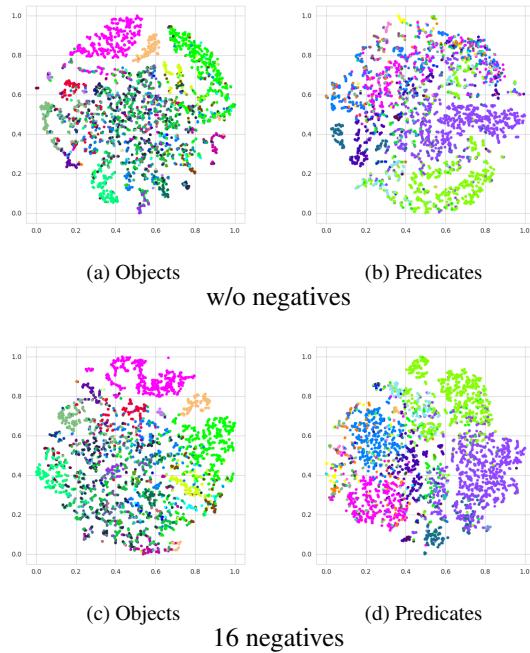


Figure 6. **t-SNE embedding for pre-training w/ and w/o negatives.** We observe better clustering for objects and predicates using negatives.

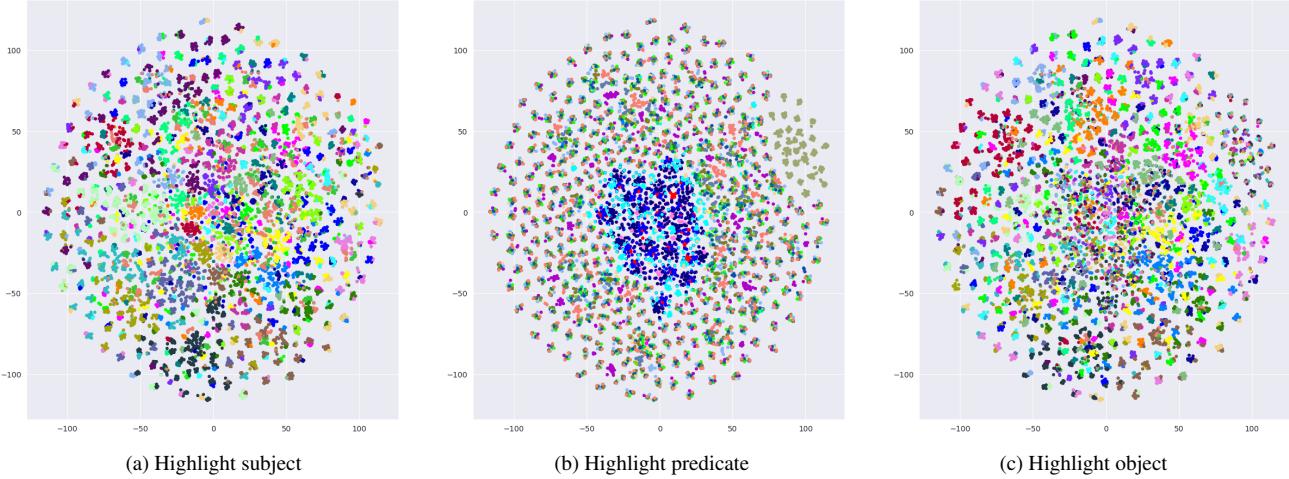
num negatives	Object		Predicate	
	R@5	mR@5	R@3	mR@3
0	0.74	0.40	0.95	0.65
16	0.77	0.43	0.96	0.67
32	0.76	0.42	0.96	0.66

Table 6. **Role of negatives.** Using negatives during pre-training produces better fine-tuned results. The best results are achieved using 16 negatives.

our pre-training which can be seen in Tab. 6.

## 8. CLIP relationship understanding

In this section, we investigate the relationship understanding capability of CLIP, raised from our findings in Tab. 4 from the main paper. A couple of works have investigated the understanding capabilities of CLIP for compositional scenes [13, 38, 65]. In contrast to these works which study the vision-language understanding of CLIP, our focus lies on language understanding only. The applicability of their investigations is therefore uncertain for our approach.



**Figure 7. CLIP embedding space for complex relationships.** We provide the same t-SNE projection for the same CLIP embeddings, highlighted by subject in (a), by predicate (b) and by object in (c). We observe that the embedding mostly clusters by subject. Some clustering can be observed for the object in the relationship. However, CLIP appears to attend less to the predicate in the relationship since no clustering for multiple classes can be observed.

However, our experiments in Tab. 4 indicate that CLIP is only able to extract limited knowledge from provided relationship descriptions of the form "A scene of a [subject] is [predicate] a [object]". To validate this issue, we perform an additional experiment visualizing the generated embeddings with CLIP in Fig. 7. We plot the embeddings for all combinations for subject, predicate and objects from a sub-sampled set of 27 object classes and 27 predicate classes resulting in  $27^3 = 19683$  unique relationships. We embed these relationships with CLIP and project their features using t-SNE. In Fig. 7 we provide this projection three times, with colored coding for the subject in (a), color coding for the predicate in (b), and color coding for the object in the relationship in (c). We observe that the feature projections generally cluster together by subject, with some clustering also appearing for objects. However, there does not appear to be particular clustering for predicates. This indicates that the investigations about compositionality from [13, 38, 65] also apply to the language-only part of CLIP. We see the potential for future research in training CLIP-like models to understand complex relationships.

## 9. Additional 3D scene graph predictions

Here we provide additional 3D scene graph predictions for a diverse set of 3D scenes. Overall, these examples confirm the results from the main paper. Most nodes and edges are predicted correctly with some edges being predicted partially correctly and only a very few nodes and edges are misclassified.

## 10. Additional zero-shot room type predictions

Here we provide additional qualitative examples for our proposed zero-shot room type classification. We provide different 3D scenes with their softmax probability based on the similarity scores between the pooled 3D graph feature and the text queries. We analyze the zero-shot capabilities on 3RScan, which consists of mostly indoor home scenes. We, therefore, choose to evaluate the room types "bathroom", "dining area", "kitchen" and "living room". There are no room-based labels in the datasets, therefore we evaluate the room type prediction on a qualitative basis only. We observe that the zero-shot predictions are very accurate and with high confidence in most of the cases. However, we notice that scenes representing dining areas get misclassified as living rooms. In this experiment, we note that the similarity between the text embedding of "dining area" and our language-aligned 3D graph features is generally low, resulting in a small softmax score for all scenes. For the dining area scenes, however, this similarity score is considerably higher but gets outscored by text queries with a stronger response.

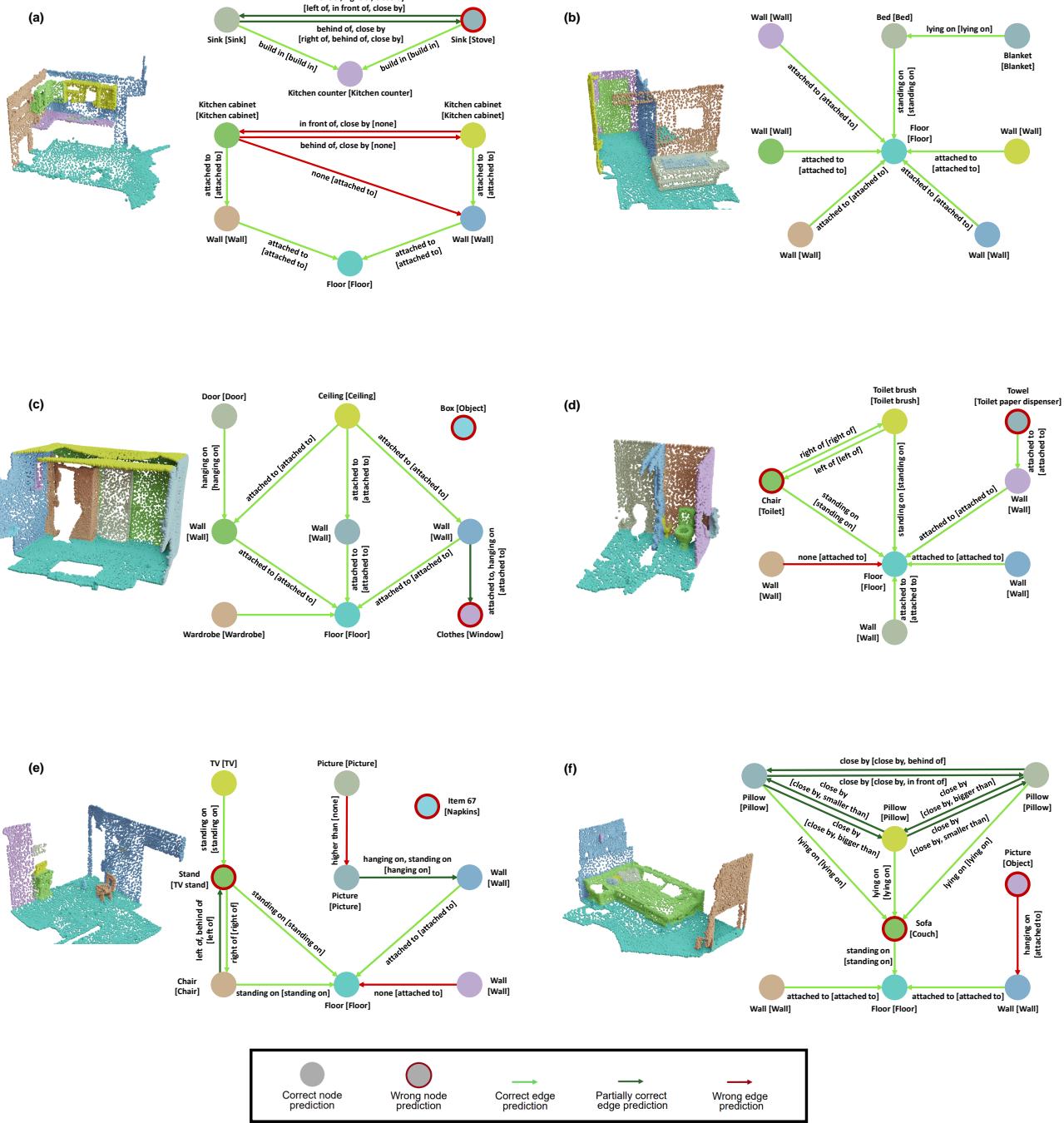
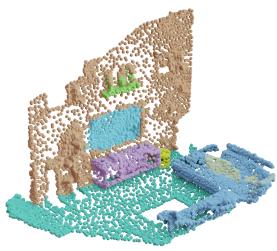
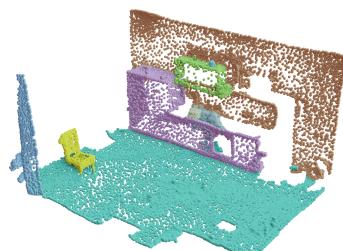


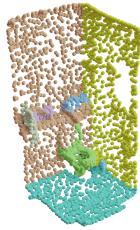
Figure 8. 3D scene graph predictions.



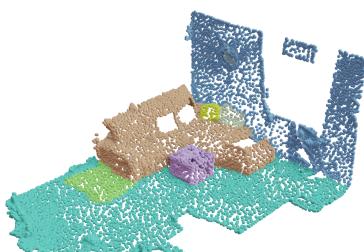
Bathroom: 06.80%  
 Dining area: 03.69%  
 Kitchen: 22.15%  
**Living room:** **67.34%**



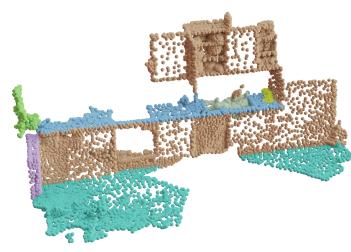
Bathroom: 12.90%  
 Dining area: 06.63%  
**Kitchen:** **59.72%**  
 Living room: 20.73%



**Bathroom:** **95.28%**  
 Dining area: 00.16%  
 Kitchen: 04.23%  
 Living room: 00.31%



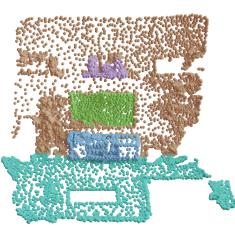
Bathroom: 01.15%  
 Dining area: 03.94%  
 Kitchen: 05.32%  
**Living room:** **89.57%**



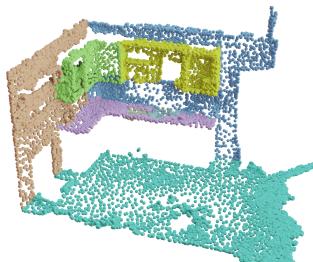
Bathroom: 12.24%  
 Dining area: 07.85%  
**Kitchen:** **43.29%**  
 Living room: 36.61%



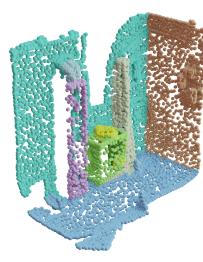
**Bathroom:** **78.34%**  
 Dining area: 02.33%  
 Kitchen: 13.03%  
 Living room: 06.27%



Bathroom: 04.84%  
 Dining area: 03.69%  
 Kitchen: 17.62%  
**Living room:** **74.07%**



Bathroom: 01.31%  
 Dining area: 00.38%  
**Kitchen:** **98.15%**  
 Living room: 00.14%



**Bathroom:** **92.07%**  
 Dining area: 00.60%  
 Kitchen: 05.97%  
 Living room: 01.34%



Bathroom: 01.80%  
**Dining area:** **29.51%**  
 Kitchen: 07.07%  
 Living room: 61.61%



Bathroom: 03.11%  
**Dining area:** **31.42%**  
 Kitchen: 11.59%  
 Living room: 53.86%

Figure 9. Zero-shot room type predictions.