Auto3DSG: Autoencoding for 3D Scene Graph Learning via Object-Level Scene Reconstruction

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

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

Abstract

3D scene graphs are an emerging representation for 3D scene understanding, combining geometric and semantic information. However, fully supervised learning of 3D semantic scene graphs is challenging due to the need for object-level annotations and especially relationship labels. Self-supervised pre-training methods have improved performance in 3D scene understanding but have received little attention in 3D scene graph prediction. To this end, we propose Auto3DSG, an autoencoder-based pre-training method for 3D semantic scene graph prediction. By reconstructing the 3D input scene from a graph bottleneck, we reduce the need for object relationship labels and can leverage large-scale 3D scene understanding datasets. Our method outperforms baseline models on the main 3D semantic scene graph benchmark and achieves competitive results with only 5% labeled data during fine-tuning.

1. Introduction

Scene graphs provide a graph-based representation of a scene by not only representing the semantic properties of objects in the scene but also their semantic relationships. In recent years, scene graphs have seen a wide range of applications in computer vision and robotics [15, 3, 2, 9].

However, predicting 3D scene graphs comes with several challenges such as noisy and incomplete sensor data as well as ambiguous object relationships. While first approaches have been proposed to learn a 3D scene graph based on a 3D point cloud [25, 31, 32, 26], these approaches require labels to be available, as they learn scene graph prediction in a fully supervised manner. However, the task of annotating data for 3D scene graph prediction requires much effort, which is underlined by the scarcity of labeled training data in this domain. Therefore, our goal is to reduce the need for such labels when learning to predict 3D scene graphs.

To this end, our contributions are: **(i)** We propose a novel autoencoder-based pre-training method for the downstream task of 3D scene graph prediction. To the best of our knowledge, we are the first to propose a pre-training strategy for 3D scene graph learning which does not require any additional scene graph labels. **(ii)** We demonstrate how to further boost downstream task performance by pre-training on common 3D datasets, which do not contain scene graph labels. **(iii)** We show that our method outperforms fully supervised baselines on a common 3D scene graph learning dataset by a considerable margin. **(iv)** Our pre-trained method demonstrates significantly improved label efficiency by requiring only 5%-10% of scene graph labels to outperform the same model trained from scratch on a completely labeled dataset.

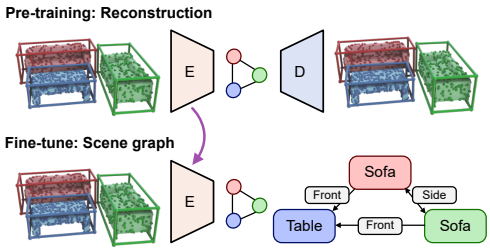


Figure 1: **Auto3DSG Overview.**

2. Related Work

3D Scene graph prediction. Wald *et al*. [25] introduce the first 3D scene graph dataset 3DSSG, with focus on semantics with 3D graph annotations, build upon the 3RScan dataset [24]. Based on this dataset, subsequent works extended the common principles of 2D scene graph prediction to 3D [31, 26]. Other works explore 3D scene generation and manipulation from 3D scene graphs [10], the use of prior knowledge [32], or the dynamic construction of 3D scene graphs [27] during the exploration of a 3D scene. In contrast, our approach focuses on a novel pre-training strategy for scene graph prediction, without requiring additional scene graph labels.

Pre-training for 3D scene understanding. In the 2D domain pre-training on existing large-scale datasets, such as ImageNet [8], is a common practice. However, the

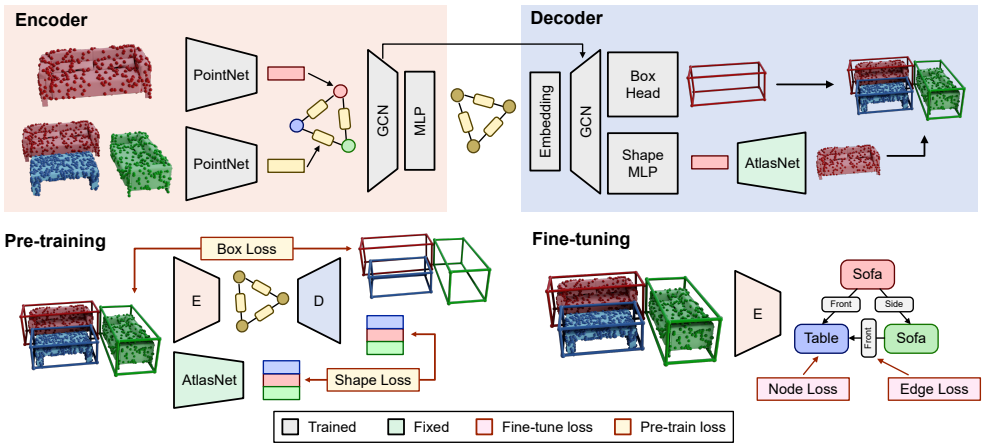


Figure 2: **Overview of Auto3DSG architecture.**

success of pre-training on the 3D equivalent ShapeNet [4] is often highly dependent on the intricate design with respect to the downstream task. For instance, [28] found that ShapeNet pre-training has no positive effect for 3D segmentation. Nevertheless, 3D representation learning approaches demonstrate that using only a fraction of available point labels can lead to similar results as obtainable with fully supervised methods when pre-trained with a self-supervised pretext task [13, 33, 28, 14, 6]. However, so far neither of these works have considered 3D scene graph prediction as the downstream task, nor are their approaches compatible with graph representations.

3. Method

We propose Auto3DSG, a novel pre-training method to learn 3D scene graphs from 3D data in an autoencoder-like manner, as shown in Fig. 2. Like all autoencoding approaches, our method consists of an encoder that maps the input to a latent representation and a decoder that reconstructs the original input from the encoder's output. However, our autoencoder-based approach fundamentally differs to existing autoencoder approaches because it maintains a graph representation within the network given a non-graph input and output. The encoder (see Sec. 3.1), takes a point cloud partitioned using object instances as input. From this input, the encoder generates a minimal representation as a 3D scene graph in a graph bottleneck, by learning to reconstruct from this representation the input scene using a decoder (see Sec. 3.3). This pre-trained architecture can then be fine-tuned to predict a semantic 3D scene graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where nodes $\mathcal{N}$ represent object instances within a corresponding 3D point cloud, while edges $\mathcal{E}$ express predicates that form together with object nodes semantic relationships (see Sec. 3.4). Each edge in the graph can represent zero or more relationships.

3.1. Encoder

Given a point cloud $\mathcal{P}$ of a scene $\mathcal{S}$ with class-agnostic instance segmentation $\mathcal{M}$ provided by an off-the-shelf instance segmentation method such as Mask3D [23] or a dataset, we extract each point set $\mathcal{P}_i$ containing instance i and its bounding box $\mathcal{B}_i$ using the mask $\mathcal{M}_i$. Moreover, for every instance pair $\langle i, j \rangle \in \|\mathcal{M}\| \times \|\mathcal{M}\|$, we get the point set $\mathcal{P}_{ij}$ using the union of their respective bounding boxes $\mathcal{B}_{ij} = \mathcal{B}_i \cup \mathcal{B}_j$. $\mathcal{P}_i$, $\mathcal{B}_i$ and $\mathcal{P}_{ij}$ serve as input to our encoder.

The encoder follows the common principles of scene graph prediction from prior 2D and 3D works [20, 29, 30, 25]. We construct an initial graph with node features ϕ_n and edge features ϕ_p from the extracted instance and bounding box features. Each point set $\mathcal{P}_i$ is fed into a shared PointNet [22] 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 $t_{ij} = \langle \phi_{n,i}, \phi_{p,ij}, \phi_{n,j} \rangle$ in a graph structure. This initial feature graph is passed into a Graph Convolutional Network (GCN) [16]. The GCN processes the triplets t_{ij} and propagates the information through the graph

$$\phi^{(e)}_{n,i}, \phi^{(e)}_{p,ij}, \phi^{(e)}_{n,j} = G(\phi_{n,i}, \phi_{p,ij}, \phi_{n,j}) \quad (1)$$

where $G(\cdot)$ is a GCN with a similar message passing procedure to [25] and $\phi^{(e)}_{n,i}, \phi^{(e)}_{n,j}$ are the encoded node features and $\phi^{(e)}_{p,ij}$ are the encoded edge features.

3.2. Graph bottleneck

Features are further processed through multiple layers of graph convolutions, propagating them to neighboring nodes. A final Multi Layer Perceptron (MLP) $f_n(\cdot)$ / MLP $f_p(\cdot)$ is applied to all node features and edge features respectively to express them as a probability distribution over the classes.

$$\phi^{(e)}_{n,i} = softmax(f_n(\phi^{(l_n)}_{n,i})), \quad \phi^{(e)}_{p,ij} = \sigma(f_p(\phi^{(l_n)}_{p,ij})). \quad (2)$$

3.3. Decoder

The goal of our scene decoder is to reconstruct the original scene from the bottleneck scene graph representation. To preserve the layout and object details, we first pass the low-dimensional features into an embedding MLP which lifts the latent graph representation to a high-dimensional feature-space. Then, we further decode the latent graph using another GCN with the same message passing structure as the encoder. Due to the low-dimensionality of the bottleneck and the ambiguity of the scene graph, the decoding step may be affected by information loss. Thus, we address this problem by introducing an additional skip-connection between the last GCN encoder layer and the first GCN decoder layer by concatenating ($\oplus$) the GCN features with the embedded feature from the bottleneck. For the node and edge features this is defined as follows

$$\phi_{n,i}^{(d_{in})} = (\phi_{n,i}^{(e)} \oplus \phi_{n,i}^{(l_n)}), \quad \phi_{p,ij}^{(d_{in})} = (\phi_{p,ij}^{(e)} \oplus \phi_{p,ij}^{(l_n)}). \tag{3}$$

Reconstructing a full 3D scene is a highly complex task, giving the sparsity of 3D data. Therefore, we choose to reconstruct each object individually rather than the complete scene. To this end, we combine the 3D bounding box of each object, predicted by the Box-Head, with the corresponding object reconstruction provided by the Shape-Head. For the final scene reconstruction, we place each generated object within its matching bounding box.

For the Box-Head, we implement an MLP to predict the box extents $[h, w, d]$ and the center location $[c_x, c_y, c_z]$. Predicting the 3D orientation of objects using regression has shown to be difficult given the non-linearity of the 3D rotation space [21]. Therefore, we predict the object's orientation angle α separately by means of classifying it into 1 out of 24 discrete bins, rather than regressing the angle directly. The Shape-Head consists of an MLP that predicts a 1D latent vector which is further processed by the decoder of AtlasNet [11] pre-trained on ShapeNet [4], which reconstructs the object from the latent vector.

3.4. Pre-training using scene reconstruction

Similarly to masked autoencoders [12], we are able to utilize an autoencoder architecture to pre-train our model using a pretext task before fine-tuning it on the downstream task. For pre-training we learn to reconstruct the 3D scene by predicting the bounding box and the shape of the objects. The loss for the object-level scene reconstruction is composed of three components: **(i)** a bounding box regression loss $\mathcal{L}_{\text{bbox}}$ which uses the L_1 distance for the bounding box parameters, **(ii)** a cross entropy classification loss $\mathcal{L}_{\text{angle}}$, and **(iii)** an L_1 loss $\mathcal{L}_{\text{shape}}$ for the shape embedding before applying the AtlasNet decoder:

$$\mathcal{L}_{\text{rec}} = \eta_1 \mathcal{L}_{\text{bbox}} + \eta_2 \mathcal{L}_{\text{angle}} + \eta_3 \mathcal{L}_{\text{shape}} \tag{4}$$

	Object		Predicate		Relationship	
Method	R@5	R@10	R@3	R@5	R@50	R@100
SGGPoint [31]	0.28	0.36	0.68	0.87	0.08	0.10
3D+MSDN [18]	0.61	0.72	0.86	0.94	0.47	0.53
3D+KERN [5]	0.67	0.77	0.83	0.96	0.51	0.58
3D+BGNN [17]	0.71	0.82	0.87	0.94	0.55	0.60
3DSSG [25]	0.68	0.78	0.89	0.93	0.40	0.66
Liu *et al*. [19]	0.74	0.83	0.90	0.96	0.62	0.68
SGFN [27]	0.70	0.80	**0.97**	**0.99**	0.85	0.87
Auto3DSG	**0.80**	**0.87**	**0.97**	**0.99**	**0.89**	**0.91**

Table 1: **3D scene graph prediction on 3DSSG.**

where η_i with $i \in \{1, \dots, 3\}$ are weighting factors. Note that this loss does not rely on scene graph labels which allows for the use of additional training data from larger data sets, such as ScanNet [7] or S3DIS [1] as we will demonstrate in Section 4.

After pre-training using $\mathcal{L}_{\text{rec}}$ loss, the model needs to be fine-tuned on the downstream task of predicting 3D scene graphs. For this, we discard the decoder and fine-tune the pre-trained encoder using the scene graph annotations from 3DSSG [25] with a fully supervised multi-task loss $\mathcal{L}_{\text{SG}}$. It consists of a cross-entropy loss $\mathcal{L}_{obj}$ for the node classification and a per-class binary cross entropy loss $\mathcal{L}_{pred}$ for the predicate prediction. The combined loss is defined as

$$\mathcal{L}_{\text{SG}} = \lambda_1 \mathcal{L}_{\text{obj}} + \lambda_2 \mathcal{L}_{\text{pred}} \tag{5}$$

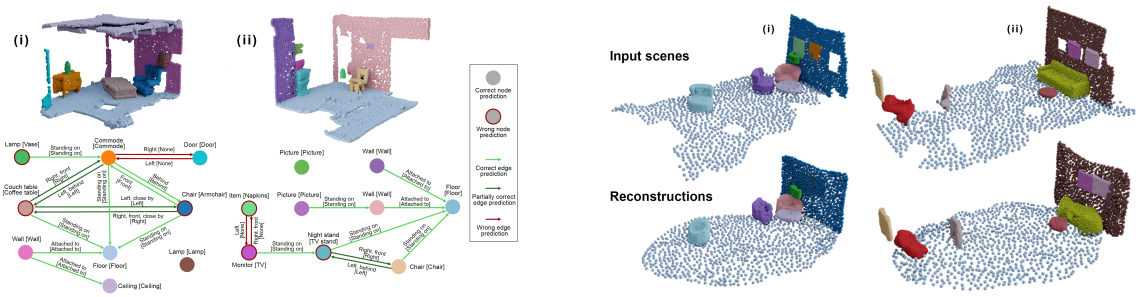
where λ_1 and λ_2 are the respective weighting factors.

4. Experiments

4.1. Experimental setup

Datasets. We evaluate the effectiveness of our proposed method on real-world 3D scans from the 3DSSG [25] dataset, which provides semantic 3D scene graph annotations. The 3D scene graphs are split into smaller sub-graphs for training and evaluation, resulting in over four thousand samples. We follow the training/evaluation splits introduced by Wald *et al*. [24]. The dataset is comparably small to other 3D datasets such as ScanNet [7] or S3DIS [1] without scene graph annotations. However, our approach allows us to utilize additional datasets without requiring ground truth scene graph annotations, unlike existing fully supervised baselines.

Evaluation metrics. Following previous works [26, 31, 25, 29, 30], we evaluate object node classification and predicate edge prediction separately. To analyze the overall scene graph prediction performance, we jointly compute the accuracy of relationships consisting of triples formed by two nodes and their connecting edge. For this, we adapt the approach first introduced by Yang *et al*. [30]. By multiplying the object node confidences with the predicate edge probability, we obtain a scored list of triplet predictions. We then compute the top-k recall metric



(a) (b)

Figure 3: **Scene graph predictions left, object-level scene reconstructions right**

first introduced by Lu *et al*. [20] for scene graph prediction.

4.2. Results

Scene Graph Prediction. We compare our proposed method on the 3DSSG dataset [25] against the fully supervised approaches in Tab. 1. Results show that Auto3DSG outperforms most recent baselines, including our closest competitor SGFN [27]. Especially on object node classification Auto3DSG outperforms SGFN by a large margin (+10%/+7%). We also observe improved results (+4%) in relationship prediction, while predicate edge prediction is similar to SGFN. We hypothesize that we have reached a saturation point for this task on this dataset.

Fig. 3a presents two predicted 3D scene graphs for complex scenes. Thanks to our pre-training, we are able to achieve nearly flawless scene graphs. In cases where nodes are incorrectly predicted, the predicted label is often just a synonym of the true class. As for the edges, no ground truth predicate is missing in the predictions. Occasionally incorrect (common-sense) predicates are predicted.

Object-level scene reconstruction pre-training. Since our downstream task includes learning relationships in scene graphs, it is most important that the relationships present in the original scene remain preserved in our reconstruction pretext task. This indicates that the model learns transferable knowledge for the downstream scene graph prediction during pre-training.

Fig. 3b shows two qualitative samples for reconstructing a 3D scene from 3DSSG. In general, our model correctly reconstructs the layouts of the scenes. In the predicted scenes, the reconstructed objects are located in similar positions to the ones in the original scene. Relationships that describe the relative proximity of objects are clearly preserved, such as *Close by, Left, Right,* etc. In scene (i), the objects hanging on walls are generated on the wrong side of the wall, however relationships like *hanging on, attached to* are still maintained in the generation. The shape of the objects differs in detail compared to the original scenes because we do not fine-tune the AtlasNet [11] decoder for more stable training. Still, the approximate shape of the objects is preserved and relationships like *same as, bigger than, smaller than* are maintained.

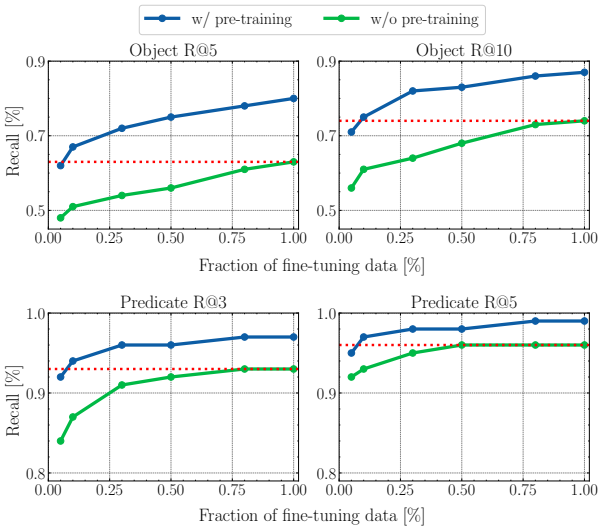


Figure 4: **Limited fine-tuning w/ and w/o pre-training.**

Limited fine-tuning data. Our pre-training reduces the need for labeled scene graph data. We demonstrate this by fine-tuning our pre-trained model on a fraction of labeled data, showing marginal impact on object, predicate, and relationship prediction. Even with 5% of labeled data, our model achieves acceptable results. Comparing limited labeled training data to training from scratch, our pre-trained model significantly outperforms in object classification and predicate prediction. It requires only 5%-10% of labeled data to outperform the model trained from scratch on the entire dataset.

5. Discussion and Conclusion

In this paper, we introduce Auto3DSG, a novel autoencoder-based pre-training method for 3D scene graph prediction. Our approach significantly improves object and relationship predictions in 3D scene graphs compared to fully supervised methods. We achieve these results by leveraging additional 3D datasets for pre-training, such as ScanNet and S3DIS that do not contain relationship labels. Notably, even with limited fine-tuning data, Auto3DSG yields competitive performance with recent methods.

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, Sasha Sax, Amir R. Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *CoRR*, abs/1702.01105, 2017. 3

[2] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1

[3] Andrzej Bielecki and Piotr Śmigielski. Graph representation for two-dimensional scene understanding by the cognitive vision module. *International Journal of Advanced Robotic Systems*, 14(1):1729881416682694, 2016. 1

[4] 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, 3

[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)*, June 2019. 3

[6] Yujin Chen, Matthias Nießner, and Angela Dai. 4dcontrast: Contrastive learning with dynamic correspondences for 3d scene understanding. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 543–560, Cham, 2022. Springer Nature Switzerland. 2

[7] 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, June 2017. 3

[8] 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. 1

[9] Helisa Dhamo, Azade Farshad, Iro Laina, Nassir Navab, Gregory D. Hager, Federico Tombari, and Christian Rupprecht. Semantic image manipulation using scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[10] 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, October 2021. 1

[11] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3, 4

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, June 2022. 3

[13] 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, June 2021. 2

[14] 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, October 2021. 2

[15] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[16] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[17] 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, June 2021. 3

[18] 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)*, Oct 2017. 3

[19] 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. 3

[20] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 852–869, Cham, 2016. Springer International Publishing. 2, 4

[21] Siddharth Mahendran, Haider Ali, and Rene Vidal. 3d pose regression using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017. 3

[22] 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)*, July 2017. 2

[23] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D for 3D Semantic Instance Segmentation. 2022. 2

[24] 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)*, October 2019. 1, 3

[25] 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)*, June 2020. 1, 2, 3, 4

[26] 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. 1, 3

[27] 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, June 2021. 1, 3, 4

[28] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 574–591, Cham, 2020. Springer International Publishing. 2

[29] 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)*, July 2017. 2, 3

[30] 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)*, September 2018. 2, 3

[31] 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, June 2021. 1, 3

[32] Shoulong Zhang, Shuai Li, Aimin Hao, and Hong Qin. Knowledge-inspired 3d scene graph prediction in point cloud. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18620–18632. Curran Associates, Inc., 2021. 1

[33] 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, October 2021. 2