Supplementary Material for "Heterogeneous Graph Learning for Scene Graph Prediction in 3D Point Clouds"

Yanni Ma[1], Hao Liu[2], Yun Pei[1], and Yulan Guo[1*]

[1] The Shenzhen Campus of Sun Yat-Sen University, Sun Yat-Sen University,
[2] Nanyang Technological University

A Implementation Details

Baseline Models. Our model is developed based on KISGP [7], which is the best baseline model so far, and the first 3D scene graph prediction work evaluated on two standard tasks: PredCls and SGCls, proposed in [6]. This work first embed class-dependent prototypical knowledge meta-embedding from class labels, then fuse prior knowledge meta-embedding into the graph neural networks to obtain significant improvement. The graph reasoning process of our 3D-HetSGP follows the similar procedure as KISGP [7], but removes the knowledge fusion, that is, our model does not fuse knowledge meta-embedding into the graph neural network (GNN). It may cause a slight performance drop, but our 3D-HetSGP can make up for it.

Initial features. Following KISGP, we pretrain the multi-scale PointNet [1] on the 3DSSG dataset and utilize the pre-trained PointNets to encode the point cloud into initial object and predicate features as the input of scene graph prediction model. In the PredCls task, since the ground truths of objects are given, we assign the corresponding object meta-embedding as the initial object features.

Type edges. For heterogeneous graph structure learning, we collect the predicted type scores $\boldsymbol{s}_t$ as type weights and filter them with a threshold α into type edges $\boldsymbol{w}_t$ indicated by 0 and 1.

$$\boldsymbol{w}_t = [(\boldsymbol{s}_t) > \alpha], t \in [1, 4]. \tag{1}$$

where $\boldsymbol{w}_1$ is the edge of 'none' class, $\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4$ are learned edges of type *support*, *proximity* and *comparative*, respectively. The Iverson bracket $[\cdot]$ is a guard function that equal 1 if their conditions are true, or equal 0 otherwise. α is a hyper-parameter, we set it as 0.1.

For heterogeneous graph structure updating, we collect 27-dimensional predicate scores predicted by heterogeneous graph reasoning stage and compute 4-dimensional type scores: we sum the corresponding dimensions of the predicate scores belonging to the same type. The computed type scores are also filtered by Eq. (1), to get updated type edges.

B Fair comparison

For unified evaluation and fair comparison on 3D scene graph prediction works, we adopt the evaluation code of KISGP [7]. We only reproduce and analyze existing 3D scene graph prediction methods with open-source code. According to our knowledge, the open-source methods include: KISGP, VL-SAT [4], SGFN [5]. KISGP is our baseline model, and VL-SAT is developed based on SGFN [5] combined with CLIP [2]. For VL-SAT and SGFN, we train their models using open-source code of VL-SAT[3], and evaluate by code of KISGP[4].

B.1 Quantitive results

Tables 1,2 report the quantitative results for the PredCls and SGCls tasks. Compared to the baseline model KISGP [7], our method achieves a significant performance improvement. Compared to the model VL-SAT, our method is still competitive. In PredCls, our 3D-HetSGP has superior performance. In SGCls, although our 3D-HetSGP is slightly inferior to VL-SAT on top-k recall (R@k) and no graph constraint top-k recall (ngcR@k), it still competitive on mean recall (mR@k) (to evaluate the performance on long-tail distribution relations). We believe that our evaluation results for VL-SAT are correct because the improvement compared to its baseline model SGFN is almost consistent with the quantitative results in the paper published by VL-SAT [4].

Table 1: Quantitative results of the evaluated methods in PredCls tasks. The model with * represents the results we have reproduced.

Model	R@20	R@50	R@100	ngcR@20	ngcR@50	ngcR@100	mR@20	mR@50	mR@100
SGFN* [5]	0.545	0.610	0.615	0.614	0.801	0.900	0.305	0.364	0.366
VL-SAT* [4]	0.582	0.653	0.659	0.658	0.859	0.936	0.375	0.445	0.447
KISGP* [7]	0.599	0.651	0.654	0.631	0.791	0.886	0.571	0.619	0.620
Ours	**0.618**	**0.659**	**0.659**	**0.703**	**0.889**	**0.946**	**0.637**	**0.681**	**0.682**

Table 2: Quantitative results of the evaluated methods in SGCls tasks. The model with * represents the results we have reproduced.

Model	R@20	R@50	R@100	ngcR@20	ngcR@50	ngcR@100	mR@20	mR@50	mR@100
SGFN* [5]	0.272	0.289	0.289	0.300	0.346	0.369	0.182	0.208	0.209
VL-SAT* [4]	**0.292**	**0.308**	**0.309**	**0.329**	**0.369**	0.387	0.239	0.256	0.256
KISGP* [7]	0.288	0.305	0.306	0.303	0.347	0.376	0.251	0.281	0.283
Ours	0.285	0.298	0.299	0.312	0.364	**0.387**	**0.273**	**0.294**	**0.295**

[3] https://github.com/wz7in/CVPR2023-VLSAT
[4] https://openreview.net/forum?id=OLyhLK2eQP

B.2 More evaluation about predicate and relationship

Type results We also add the comparison results of our 3D-HetSGP with VLSAT on per predicate type: *support, proximity*, and *comparative*, and their mean values. As shown in 3, VL-SAT achieves the best results on the *proximity* type, mainly because of the high frequency of predicates in this type. Our 3D-HetSGP achieves the best results on the mean of these three types, which shows that our prediction for these three types are better overall.

Table 3: Comparison results of no graph constrain Recall@20/50/100 for *Support, Proximity, Comparative* types.

Type	*Support*			*Proximity*			*Comparative*			Mean		
Methods	R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100
SGFN	71.82	82.29	89.72	51.44	74.14	84.28	10.34	16.75	21.91	10.34	16.75	21.91
VL-SAT	71.68	83.63	90.72	**58.98**	**81.23**	**88.25**	12.29	20.35	24.98	47.65	61.74	67.98
KISGP	**89.00**	**94.54**	**96.43**	41.88	61.85	73.94	20.12	24.14	26.32	50.33	60.18	65.56
Ours	88.88	93.22	94.93	51.84	79.07	86.31	**25.63**	**27.98**	**29.07**	**55.45**	**66.76**	**70.10**

Long-tail distribution We also add the comparison results of our 3D-HetSGP with VL-SAT on long-tail distribution problem. We report the R@k and mR@k metric on each groups in Table 4. VL-SAT achieves the best performance on the head classes but relatively inferior on body and tail classes, which means it is good at predicting high-frequency predicates. Our method generally achieves relatively high performance in dealing with long-tail distribution problems.

Table 4: Comparison results of Recall@20/50 and mean Recall@20/50 for the head, body, tail predicate classes in 3DSSG [3].

	Head		**Body**		**Tail**	
Methods	R@20/50	mR@20/50	R@20/50	mR@20/50	R@20 / 50	mR@20/50
SGFN	58.50/65.17	50.21/58.67	23.23/27.37	33.63/42.49	8.58/9.85	17.36/19.58
VL-SAT	**62.00/68.96**	**53.65/62.51**	28.04/33.61	39.93/52.83	9.73/10.64	27.95/29.56
KISGP	59.84/65.00	51.43/58.84	49.13/52.61	69.66/76.46	20.50/20.74	61.87/62.52
Ours	59.53/64.01	50.39/57.40	**54.56/56.51**	**80.48/82.76**	**22.3/23.66**	**66.30/70.75**

Splits of Predicates. To evaluate the performance on the long-tail distribution problem, we split the 26 predicate classes into three parts: head, body, tail (as shown in Table 5. In detail, we sort the predicates in descending order based on their frequencies in the test set and divide them into three groups, with the frequencies of categories within a group not exceeding four times.

Table 5: Splits of predicate categories.

Split	Predicate categories
Head	close by, left, right, standing on, attached to, front, behind
Body	same as, lying on, higher than, lower than, hanging on, bigger than, smaller than, supported by
Tail	standing in, connected to, same symmetry as, leaning against, belonging to, lying in, build in, part of, cover, hanging in, inside

C Ablation study

Ablation on spatial features f^{spat} and geometric features f^{geom}. As shown in Table. 6, we conduct the ablation experiments on spatial features f^{spat} and geometric features f^{geom}. When the spatial f^{spat} and geometric features f^{geom} are provided, the mR@20 is significantly improved by 2.36%.

Table 6: Ablation study on spatial features f^{spat} and geometric features f^{geom}.

f^{spat}	f^{geom}	mR@20	mR@50	mR@100
		61.35	66.67	66.69
	✓	62.87	67.68	67.81
✓		61.92	67.03	67.06
✓	✓	**63.71**	**68.14**	**68.19**

Ablation on hyperparameter α. We conduct ablation experiment on α, as shown in Table. 7. The best performance is achieved at $\alpha = 0.1$. In addition, the highest average values of type_acc and edge_acc are also achieved at $\alpha = 0.1$.
Ablation on the number of message passing iterations. The mR@100 results with different iteration numbers are reported, as shown in Table. 8. The iteration number is set to 3.

Table 7: Ablation study on hyperparameter α.

α	0.1	0.2	0.3	0.4	0.5
avg_acc	**94.0**	93.5	93.0	92.5	92.0

Table 8: Ablation study on iteration number.

Iterations	2	3	4	5	6
mR@100	0.675	**0.682**	0.681	0.671	0.677

D Qualitative Results

We visualize more qualitative results to show our prediction of the scene graph on 3DSSG [3], as shown in Fig. 1.

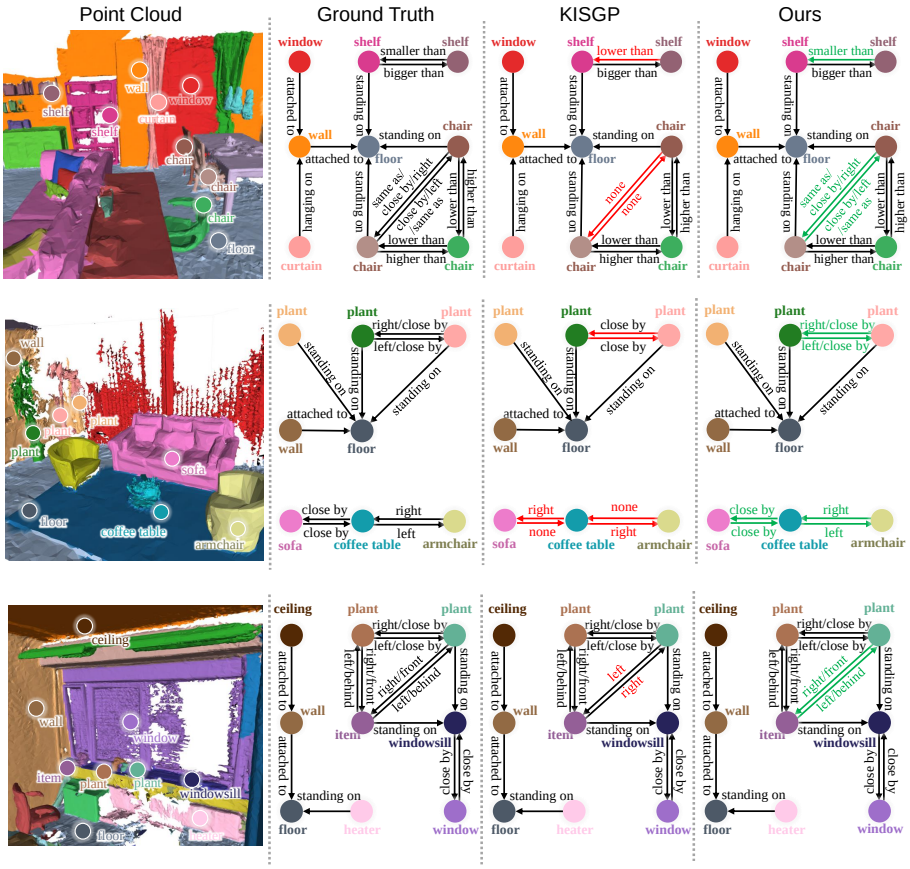

Fig. 1: Qualitative results between ours and KISGP [7] on no graph constraint Recall@20 for the PredCls task. Red arrows: incorrect predictions by KISGP, Green arrows: correct predictions by our model.

E Limitation

Although our proposed 3D-HetSGP achieves promising performance in 3D scene graph prediction, it still has several limitations. The improvement in the SGCls task is not as significant as in the PredCls task. This limitation arises due to the intrinsic characteristics of point clouds and issues with object labeling (such as severe long-tail distribution, ambiguous labeling among 160 object classes). In the future, we plan to investigate the heterogeneity and hierarchy of 3D objects, minimize the gap between machine and human understanding of the real world.

F Potential Negative Social Impact

It would not be a concern for our 3D-HetSGP, which is trained and tested on open-sourced dataset with delicate data protection.

References

1. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 652–660 (2017)
2. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
3. Wald, J., Dhamo, H., Navab, N., Tombari, F.: Learning 3D semantic scene graphs from 3D indoor reconstructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3961–3970 (2020)
4. Wang, Z., Cheng, B., Zhao, L., Xu, D., Tang, Y., Sheng, L.: VL-SAT: Visual-linguistic semantics assisted training for 3D semantic scene graph prediction in point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 21560–21569 (2023)
5. Wu, S.C., Wald, J., Tateno, K., Navab, N., Tombari, F.: Scenegraphfusion: Incremental 3D scene graph prediction from RGB-D sequences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7515–7525 (2021)
6. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5410–5419 (2017)
7. Zhang, S., Hao, A., Qin, H., et al.: Knowledge-inspired 3D scene graph prediction in point cloud. Proceedings of the Advances in Neural Information Processing Systems (NeruIPS) **34**, 18620–18632 (2021)