Supplementary Material: Incremental 3D Semantic Scene Graph Prediction from RGB Sequences

Shun-Cheng Wu[1] Keisuke Tateno[2] Nassir Navab[1] Federico Tombari[1,2]
[1]Technische Universität München [2]Google

6. Method Comparison and Implementation

We provide an overview of all the methods evaluated in Sec. 4.2, including our adaptation to enable comparisons among them in our experiments. As a brief recall, we compare our method to two image-based methods, *i.e*. IMP [16] and VGfM [2], and two point-based methods, 3DSSG [12] and SGFN [14]. All methods shared a similar scene graph generation pipeline:

1. using a node and an edge encode to compute an initial node and edge embeddings.
2. using message passing to calculate messages for updating node and edge features.
3. updating node and edge feature with the messages from step 2.
4. integrating class prediction over time.

For all methods, we closely follow the original implementation from the respective papers. We refer interested readers to check out their papers for all detail. Here, we describe our adaptation to enable fair comparison in our experiments. We report the comparison of all five methods in Tab. 6.

Object Detection. IMP and VGfM rely on a regional proposal network to detect objects. We replace it with our entity detection methods described in Sec. 3.1.2.

Geometric Features in VGfM. For VGfM, they extract geometric features from ellipsoids. In our implementation, we replace the use of ellipsoids with oriented bounding boxes, which can provide equivalent information as needed by VGfM.

Fusion for IMP. For IMP, as mentioned in the main paper, we added the voting mechanism in [5] to fuse multiple predictions. For handling the incremental nature, our experiment in Table 1 follows the setup in [14]. We do a single global estimation of all methods and provide incremental estimations of our methods. Hence, no modification is needed for all the baseline methods.

Entity visibility graph and the neighbour graph for the *GT* setup Unlike in *Dense* and *Sparse*, where we run an incremental estimation system to obtain $\mathcal{G}_c$ and $\mathcal{G}_p$, the *GT* requires an additional procedure to obtain the entity visibility graph $\mathcal{G}_c = (\mathcal{V}, \mathcal{K}, \mathcal{E}_c)$ and the neighbour graph $\mathcal{G}_p = (\mathcal{V}, \mathcal{E}_p)$. The entities $\mathcal{V}$ are directly inherited from the ground truth annotation from the 3RScan dataset [11]. The proximity edges $\mathcal{E}_p$ is estimated with the same strategy as described in Sec. 3.1.4, using the ground truth bounding boxes of entities. Here we detail how we estimate $\mathcal{K}$ and $\mathcal{E}_c$ for the *GT* setup. For $\mathcal{K}$ and $\mathcal{E}_c$, we follow a similar approach as in VGfM [2], where we first find all relevant frames across all entities and then select keyframes with our keyframe selection strategy (Sec. 7). We use the ground truth instance mask from [11] to check if an entity appears in an image. However, an entity may be heavily occluded and cannot provide reasonable image features. To avoid this, we estimate the occupancy of an entity in an image as the ratio of the number of relevant pixels over all pixels within the bounding box of the entity. Furthermore, to prevent an entity is not aligned to the image coordinate, which will cause the occupancy value to be very low even if it is not occluded, we downscale each input mask by a factor of 8 with a maximum relevant selection, *i.e*. a downscaled pixel is considered relevant if one among the eight pixels in the original image is relevant. This gives the visibility of each entity on each input frame. We then apply the keyframe selection strategy to prevent duplicate views and to ensure good view coverage.

7. Keyframe selection strategy.

Selecting keyframes is crucial when the estimation quality is solely based on multiview images. Having diverse view coverage of objects usually results in better feature representation of objects [4, 13, 17]. Unlike the keyframes selection in ORBSLM [3], which focuses more on the pose estimation quality than the view coverage, we select keyframes mainly based on the pose difference and, in addition, the quality of detected objects. A frame is selected as a keyframe only 1) it has at least one valid object detected

Method	Node Type	Edge Type	Message Passing Type	Message Update Method	Fusion
IMP	Image ROI	Image ROI Union	Prime-Dual	GRU	Voting [5]
VGfM	Image ROI	Image ROI Union	Prime-Dual + Geo. description	GRU	Temporal Gate
3DSSG	Points	Points Union	Triplet	concatenation	N/A
SGFN	Points	geometric description	FAN	concatenation	running mean
Ours	MV Image ROIs	geometric description	FAN + Gated Points	GRU	running mean

Table 6. A summary of the modules used for different methods. The text colored in cyan involves our modification to make all methods comparable.

and 2) its pose is dissimilar to other existing keyframes. We measure the validity of bounding boxes by checking if their minimum width and height are larger than 200 pixels, and the pose difference threshold is set to 5 degrees in rotation and 0.3 meters in translation. This keyframe selection method is used for all input, *i.e*. *GT*, *Dense*, and *Sparse*, cases in our experiments.

8. Data Distribution

We provide the class distribution on objects and predicates in Fig. 5. It can be seen that the setup in [12] has severe long-tail data distribution. After mapping to 20 NYUv2 labels [8], the distribution is relatively well distributed but still unbalanced. The unbalanced distribution indicates that the *mRecall* metric reflects better the model performance.

9. Multi-view Feature Encoding

As mentioned in Sec. 3.2.1, the multi-view feature of a node is computed by aggregating image features of the node from multiple images. This is essentially the task of 3D shape recognition with arbitrary views. We compare the use of simple mean aggregation, as in MVCNN [9], with the state-of-the-art method, CVR [13] on ScanNet [1] dataset. Since the main comparison is on the multi-view feature aggregation, we use the same backbone, ResNet18 [3], for both methods. All networks are trained from scratch using the splits from ScanNet, following the same training approach described in the respective papers. We report the mean intersection-over-union (*mIoU*), precision (*mPrec*), and recall (*mRecall*).

The result is shown on Tbl. 7. It can be seen that the use of simple mean operation outperforms canonical transformation in CVR [13]. Although the number reported in CVR [13], it outperforms MVCNN in ModelNet40 [15], ScanObjectNN [10] and RGBD [6]. We investigated the difference between the three datasets mentioned above and ScanNet and found that the images from ScanNet [1] contain more background objects while the others have non-cluttered backgrounds. We demonstrate some example images in Fig. 6. The performance inconsistency may indicate that using the mean operator makes the model less sensitive to background things.

	mIoU(%)	*mPrec*(%)	*mRecall*(%)
CVR [13]	32.3	45.2	54.4
MVCNN [9]	**39.2**	**50.5**	**61.4**

Table 7. Object classification result on ScanNet dataset. A simple averaging of overall image features (MVCNN) outperforms the sophisticated multi-view image encoding method (CVR).

10. Comparing confidence and IoU-based methods

We provide an example of the difference between using the maximum IoU and our maximum mean confidence to find the most probable correspondence with a sparse point map. In figure 7, given three consecutive frames at $t = n$, $t = n+1$ and $t = n+2$ for the label association and fusion, their entity maps and the association result are shown on the second and third columns (separated by white space). The second column is the label fusion result with IoU [7], and the third column is ours. When using IoU (second column), the table label at $t = n$ is assigned to the floor at $t = n + 2$. This is due to the map points created at $t = n + 1$ on the floor (carpet) having larger IoU than the table. With our approach (third column), the table label at $t = n$ remains at the table at $t = n + 2$. This shows that our method provides more consistent label association. Note that the label colors are different between IoU and ours since the segment color are randomly in each run.

11. Ablation Study

We provide two additional ablation studies: (1) the use of edge descriptor (Tab. 8) and (2) the effect of the sigmoid gate on the geometric feature (Tab. 9). In Tab. 8, it can be seen that the use of our relative pose descriptor $R_{i\rightarrow j}$ leads to better overall performance in *GT* and *Sparse* settings while having slightly worse performance in *Dense* setting. In Tab. 9, using the gated geometric feature consistently achieves better performance in all three setups.

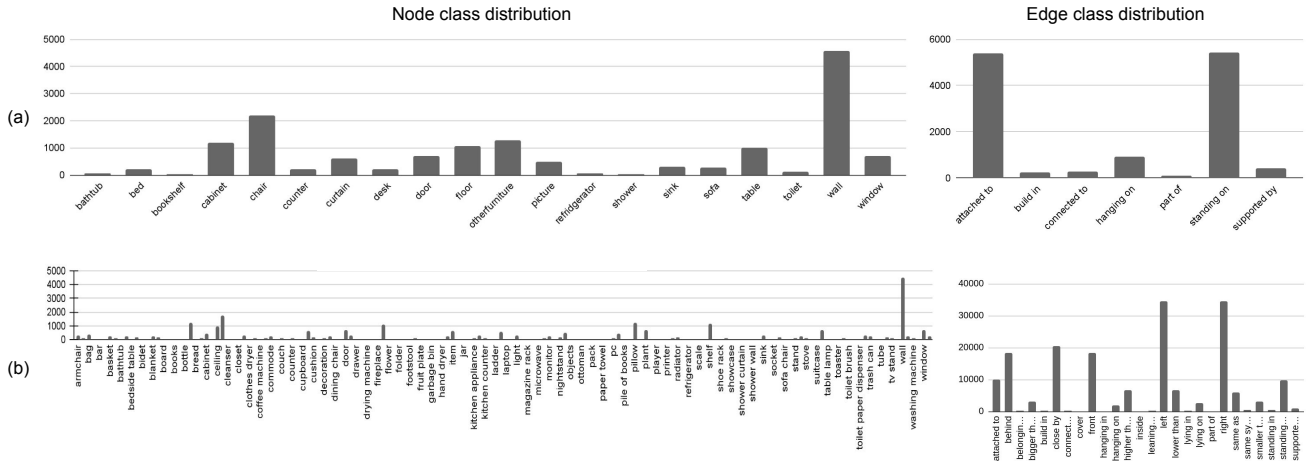

Figure 5. We provide the class distribution for two scene graph experiments in Tbl. 1 and Tbl. 2. The row (a) is the distribution for the experiment setup in [14] and in Tbl. 1. (b) is the distribution for the experiment setup in [12] and in Tbl. 2.

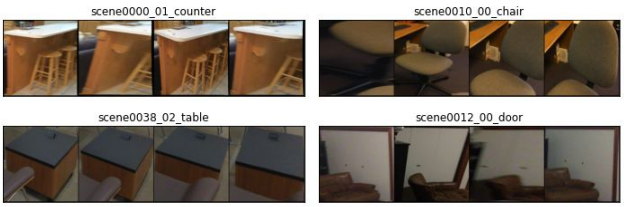

Figure 6. We provide examples of multi-view images of the same object in the ScanNet dataset. For display purposes, we only select four views of the same object. It can be seen that the multi-view observation of an object in a standard indoor dataset includes some non-target objects in the field of view. Those objects are considered noise which affects the multi-view image encoding.

	Edge Type	*Recall*(%) Rel.	Obj.	Pred.	*mRecall* Obj.	Pred.
GT	*Points*	62.0	77.9	**95.9**	68.4	69.0
	$R_{i\to j}$	**66.1**	**81.2**	95.6	**77.4**	**71.5**
Dense	*Points*	**35.1**	**57.5**	89.6	**47.9**	33.2
	$R_{i\to j}$	34.1	58.1	**89.9**	43.0	**33.3**
Sparse	*Points*	10.0	28.7	**90.6**	21.1	16.3
	$R_{i\to j}$	**9.9**	**29.5**	90.4	**23.5**	**16.5**

Table 8. Ablation study on the use of input type for computing edge feature. The experiment setup is the same as in Tbl. 1. Here *Points* means taking the point cloud union as in [12], and $R_{i\to j}$ is the relative pose descriptor described in Sec. 3.2.2.

11.1. Comparison in edge descriptor

compare non-learned and learned descriptors.

	Gate	*Recall*(%) Rel.	Obj.	Pred.	*mRecall* Obj.	Pred.
GT		61.5	77.1	95.3	77.1	70.9
	✓	**66.1**	**81.2**	**95.6**	**77.4**	**71.5**
Dense		32.9	55.5	89.1	41.0	31.4
	✓	**34.1**	**58.1**	**89.9**	**43.0**	**33.3**
Sparse		8.6	26.9	**90.5**	24.4	15.6
	✓	**9.9**	**29.5**	90.4	**23.5**	**16.5**

Table 9. Ablation study on the proposed gated geometric feature. We ablate the use of a sigmoid function of a gate for the input geometric feature using the same experimental setup as in Tbl. 1.

11.2. Gate on the geometric feature

whether to use the gate in the geometric feature or to use different ways of selecting keyframes. We provide an additional ablation study on the effect of using a sigmoid gate when including the geometric feature in our message passing payer.

12. Additional Results

12.1. Per-class prediction result

In Tbl. 10, We provide the per entity class recall for the experiment reported in Tbl. 1. Our method has dominant performance on most of the classes regardless of the input segmentation types.

12.2. Without consider *None* estimation

Our evaluations follow the line of work [12, 14] which consider *None* relationship is crucial, unlike other work [2,

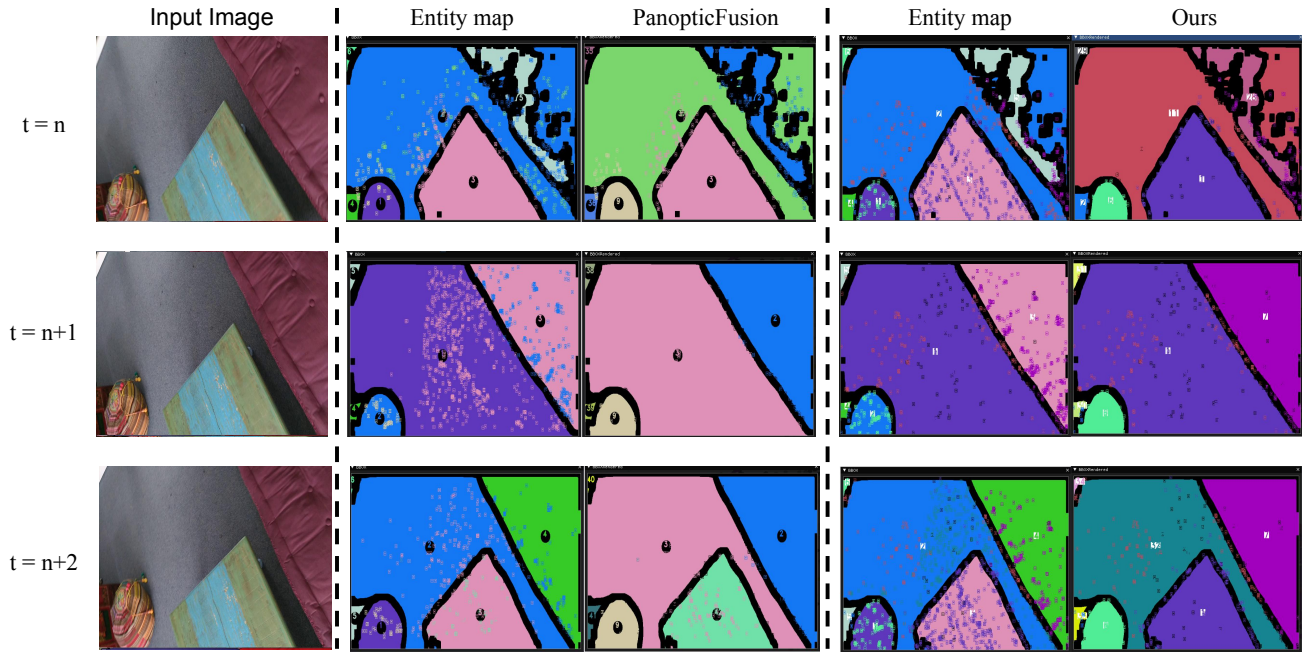

Figure 7. An example of comparing the use of IoU and our label association approach with the sparse input points. Our method handles non-uniformly distributed data better than the IoU-based method.

		bath.	bed	bkshf	cab.	chair	cntr.	curt.	desk	door	floor	ofurn	pic.	refri.	show.	sink	sofa	table	toil.	wall	wind.	avg
GT	IMP	0.000	0.667	0.143	0.562	0.688	0.677	0.712	0.292	0.541	0.957	0.262	0.727	0.000	0.143	0.617	0.579	0.731	0.889	0.856	0.560	0.530
	VGfM	0.750	0.833	**0.286**	0.423	0.854	0.677	0.712	0.042	0.658	**0.969**	0.319	0.693	0.000	0.286	0.633	0.592	0.641	0.852	0.847	0.429	0.575
	3DSSG	0.500	0.333	0.000	0.477	0.838	0.516	0.432	0.417	0.622	0.957	0.288	0.205	0.000	0.143	0.717	0.605	0.563	0.630	0.609	0.357	0.460
	SGFN	1.000	0.833	0.143	0.385	0.696	**0.839**	0.577	0.417	0.631	0.963	0.372	0.830	0.111	0.143	0.783	0.434	0.647	0.593	0.718	0.429	0.577
	Ours	**1.000**	**1.000**	0.000	**0.619**	**0.927**	0.645	**0.847**	**0.583**	**0.838**	**0.969**	**0.539**	**0.943**	**0.667**	**1.000**	**0.900**	**0.671**	**0.808**	**1.000**	**0.877**	**0.655**	**0.774**
Dense	IMP	0.000	0.667	0.000	0.381	0.453	0.000	0.477	0.000	0.081	0.951	0.199	0.023	0.000	0.000	0.200	0.474	0.485	**0.667**	0.770	**0.179**	0.300
	VGfM	0.000	0.667	0.000	0.346	0.494	0.000	0.486	0.042	0.198	0.957	0.141	0.011	0.000	0.000	0.233	**0.579**	0.569	0.630	**0.780**	**0.179**	0.316
	3DSSG	0.250	0.667	0.000	0.200	0.510	**0.258**	**0.505**	0.000	**0.477**	0.914	0.147	0.034	**0.222**	**0.143**	0.250	0.474	0.425	0.259	0.519	0.131	0.319
	SGFN	**0.750**	0.333	0.000	**0.508**	0.636	0.194	0.405	0.083	0.387	**0.969**	0.230	**0.114**	0.111	0.000	**0.383**	0.553	**0.623**	0.519	0.730	0.131	0.383
	Ours	**0.750**	**1.000**	0.000	0.504	**0.656**	0.194	0.459	**0.125**	0.342	**0.969**	**0.251**	0.057	0.000	**0.143**	**0.383**	**0.579**	0.599	**0.667**	0.761	0.155	**0.430**
Sparse	IMP	0.000	**0.333**	0.000	0.235	0.146	**0.129**	0.252	**0.167**	0.099	0.798	0.068	0.023	0.000	**0.286**	0.183	**0.329**	**0.281**	0.259	0.324	**0.214**	0.206
	VGfM	0.000	**0.333**	0.000	**0.273**	0.150	0.097	0.243	0.042	0.081	**0.810**	0.047	0.011	0.000	0.000	0.150	0.276	0.263	0.222	0.325	0.202	0.176
	3DSSG	0.000	0.000	0.000	0.085	0.045	0.000	0.108	0.000	0.063	0.558	0.005	0.011	0.000	0.000	0.017	0.000	0.186	0.000	0.048	0.060	0.059
	SGFN	0.000	0.000	0.000	0.058	0.081	0.032	0.072	0.042	0.063	0.712	0.010	0.034	0.000	0.000	0.083	0.092	0.174	0.000	0.097	0.107	0.083
	Ours	**0.500**	**0.333**	**0.286**	**0.273**	**0.227**	**0.129**	**0.270**	0.042	**0.135**	**0.810**	**0.110**	**0.068**	0.000	0.000	**0.217**	0.211	0.263	**0.296**	**0.342**	0.179	**0.235**

Table 10. The per-class *Recall* of all methods in 3RScan dataset [11] with 20 node classes.

16] which only consider edges with annotated non-*None* relationships. Both approaches have their advantages. Considering *None* estimation prevents excessive relationship estimation while also preventing potential relationship discovery, *e.g.* should exist but was not annotated. For further comparison and the interest of potential readers, we provide the evaluation result without considering the *None* relationship in Tab. 11 and Tab. 12, with the experiment setup as reported in Tbl. 1 and Tbl. 2.

References

[1] Angela Dai, Angel Xuan Chang, Manolis Savva, Maciej Halber, Tom Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *CVPR*, 2017. 2

[2] Paul Gay, James Stuart, and Alessio Del Bue. Visual Graphs from Motion (VGfM): Scene Understanding with Object Geometry Reasoning. In *ACCV*. Springer, 2018. 1, 3, 5

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[4] Chih-Hui Ho, Bo Liu, Tz-Ying Wu, and Nuno Vasconcelos. Exploit clues from views: Self-supervised and regularized learning for multiview object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9090–9100, 2020. 1

[5] Ue-Hwan Kim, Jin-Man Park, Taek-Jin Song, and Jong-Hwan Kim. 3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents. *IEEE transactions on cybernetics*, 50(12):4921–4933, 2019. 1, 2

	Method	*Recall*(%)			*mRecall*(%)	
		Rel.	Obj.	Pred.	Obj.	Pred.
GT	IMP [16]	43.5	70.1	54.2	53.0	38.1
	VGfM [2]	49.0	69.4	61.9	57.5	44.6
	3DSSG [12]	40.4	58.0	69.3	46.8	58.7
	SGFN [14]	47.2	63.8	71.4	57.7	65.5
	Ours	**64.8**	**81.2**	**76.8**	**77.4**	**71.5**
Dense	IMP [16]	18.3	51.8	19.3	30.0	23.0
	VGfM [2]	20.8	53.3	22.1	31.6	24.4
	3DSSG [12]	15.1	41.4	26.1	31.9	26.6
	SGFN [14]	24.4	56.7	27.2	38.3	30.5
	Ours	**25.5**	**58.1**	**27.3**	**43.0**	**33.3**
Sparse	IMP [16]	3.5	27.5	4.0	20.6	14.0
	VGfM [2]	3.7	26.9	4.2	17.6	15.4
	3DSSG [12]	1.0	9.7	**9.2**	5.9	15.1
	SGFN [14]	2.3	12.6	7.0	8.3	14.4
	Ours	6.9	29.5	8.1	23.5	**16.5**
	Ours (i)	**7.1**	**30.2**	8.4	**24.5**	15.9

Table 11. We report top-1 recall of method without considering *None* relationship. The experiment setup is the same as in Tbl 1.

Method	*Recall*(%)			*mRecall*(%)	
	Rel.	Obj.	Pred.	Obj.	Pred.
IMP [16]	3.3	35.9	9.0	18.7	4.9
VGfM [2]	3.4	37.9	14.7	17.9	6.5
3DSSG [12]	7.3	29.6	**68.8**	11.7	**25.5**
SGFN [14]	4.5	29.4	45.8	11.8	13.5
Ours	**17.9**	**56.7**	50.4	**27.2**	23.9

Table 12. We report top-1 recall of method without considering *None* relationship. The experiment setup is the same as in Tbl 2.

[6] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *2011 IEEE international conference on robotics and automation*, pages 1817–1824. IEEE, 2011. 2

[7] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things. In *IEEE Conf. Intelligent Robots and Syst.*, 2019. 2

[8] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*, 2012. 2

[9] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 2

[10] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019. 2

[11] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. RIO: 3D Object Instance Re-Localization in Changing Indoor Environments. In *ICCV*, 2019. 1, 4

[12] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3D Semantic Scene Graphs from 3D Indoor Reconstructions. In *CVPR*, 2020. 1, 2, 3, 5

[13] Xin Wei, Yifei Gong, Fudong Wang, Xing Sun, and Jian Sun. Learning canonical view representation for 3d shape recognition with arbitrary views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 407–416, 2021. 1, 2

[14] 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*, pages 7515–7525, 2021. 1, 3, 5

[15] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang 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*, pages 1912–1920, 2015. 2

[16] Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. Scene Graph Generation by Iterative Message Passing. In *CVPR*, 2017. 1, 3, 5

[17] Ze Yang and Liwei Wang. Learning relationships for multi-view 3d object recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7505–7514, 2019. 1