Supplementary Materials

Neural network details. The initialization MLP f_{init} takes as input a $(C+6)$-dimensional raw object representation vector, where C is the number of object categories, and the rest 6 dimensions represent the object's 3D position $\mathbf{p} \in \mathcal{R}^3$ and scale $\mathbf{d} \in \mathcal{R}^3$. It processes the input with a hidden layer of 300 units, then ReLUs, and another hidden layer of 300 units followed by ReLUs. It outputs a 100-dimensional node representation.

The message MLP f^r_{msg} takes as input two concatenated 100-dimensional node representations. It processes the input with a hidden layer of 300 units, then ReLUs, and another hidden layer of 300 units followed by ReLUs. It outputs a 100-dimensional message representation.

The attention network f_{att} takes as input two concatenated raw object representations (input $2C + 12$ dimensions). It processes the input with a hidden layer of 300 units, then ReLUs, and another hidden layer of 300 units followed by a sigmoid activation. The output is a single scalar weight between 0 and 1.

The aggregation network f_{GRU} processes 100-dimensional message representations at each step. Its internal memory and output is 100-dimensional.

The update network f_{upd} operates on the node representation concatenated with the aggregated messages from six relationships (700$-$dimensional input). It processes the input with a hidden layer of 300 units, then ReLUs, and another hidden layer of 300 units followed by ReLUs. It outputs a 100-dimensional updated node representation.

The prediction MLP processes the input 100-dimensional node representation through a hidden layer of 300 units, then ReLUs, and another hidden layer of 300 units followed by a softmax activation to output object category probabilities. We also use a MLP with the same structure to regress to object size. The output of the last hidden layer for this MLP is linearly transformed to object size.

Details on relationship extraction. The "supporting" relationship between two objects (i, j) is calculated by checking if the bottom of the object i's bounding box is higher and also within a short range of distance (set as 0.05 meters to prevent placement errors) compared to the top surface of the object j's bounding box. The "surrounding" relationship is calculated by finding whether objects $(k_1, k_2, ...)$ form a symmetry wrt the central object (using a threshold difference of 0.2 meters). In addition, the surrounding objects should have similar sizes (they qualify as similar if their bounding boxes differ less than a factor 1.2x when compared to each other).

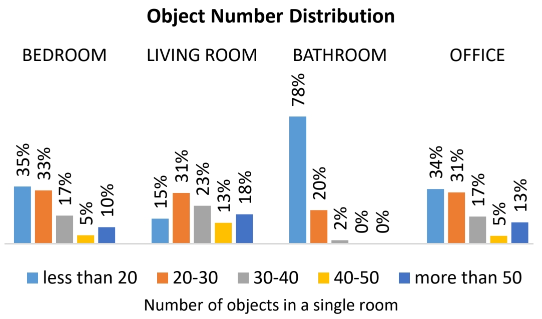

Figure 1. Distribution of #objects for each room type.

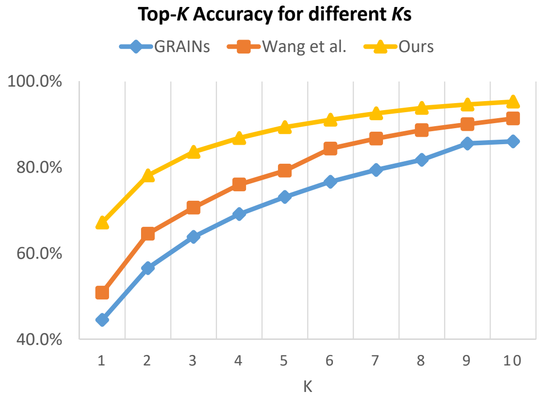

Figure 2. Top-K accuracy of object category prediction for different Ks.

Dataset details. Following Wang et al. [20], the experiments are performed on the SUNCG dataset with four room types: bedroom, living room, bathroom and office. We have 51 object categories in bedrooms, 31 in bathrooms, 51 in living rooms, and 42 in offices. We also count the number of objects in each room per room type. Figure 1 shows a histogram over number of objects inside a room per each room type. We note that we will publish the splits and implementation upon acceptance.

Detailed results per room type. Table 1 shows the top-K accuracy of category prediction for all different methods and our degraded variants per each room type ($K = 1, 3, 5$) separately. Based on Figure 1 and Table 1, we can see that for bathrooms that have low number of objects (78% of them has less than 20 objects), our method has 8.4% higher prediction accuracy than Wang et al. [20] in terms of top-1 accuracy, while for living rooms that contain more objects (only 15% of them has less than 20 objects), ours significantly outperforms Wang et al. [20] by a margin of 21.0% in terms of top-1 accuracy.

Method	Bedroom			Living room			Bathroom			Office		
	Top1	Top3	Top5	Top1	Top3	Top5	Top1	Top3	Top5	Top1	Top3	Top5
GRAINS [12]	45.1	63.4	72.2	43.7	64.3	73.7	42.4	63.4	75.1	45.6	64.5	73.5
Wang et al. [20]	48.9	69.8	79.5	46.6	61.1	69.0	61.4	82.8	89.7	46.6	69.0	78.8
SGNet-tree	60.0	78.1	85.4	63.7	80.7	87.2	61.3	83.4	90.4	59.6	76.5	84.9
SGNet-sparse	59.3	77.7	84.8	62.4	79.7	86.5	60.5	82.6	90.4	57.9	74.5	84.1
SGNet-co-occur	57.6	76.1	83.4	41.1	62.8	72.7	67.7	86.8	92.3	59.4	75.7	84.7
SGNet-sum	57.3	76.2	83.6	59.0	77.9	84.1	55.5	79.7	88.2	59.0	76.8	84.4
SGNet-max	63.1	79.6	86.1	61.1	79.3	85.3	67.7	88.2	92.9	60.3	78.9	85.0
SGNet-vanilla-rnn	64.3	81.3	87.8	65.0	81.1	87.3	67.6	86.5	92.0	62.3	**80.4**	**86.8**
SGNet-no-attention	60.2	77.6	83.8	62.2	79.8	86.4	61.6	82.1	89.7	57.1	76.8	83.6
SGNet-dist-weights	61.8	79.3	86.8	63.6	81.2	87.4	67.0	86.6	92.3	62.7	79.3	85.3
SceneGraphNet (full model)	**66.8**	**82.9**	**88.6**	**67.6**	**83.8**	**89.6**	**69.8**	**88.6**	**93.5**	**64.8**	79.9	86.5

Table 1. Top-K accuracy for category prediction for each room type.

Performance for different top-K accuracy. Figure 2 shows the top-K accuracy of object category prediction over different Ks ($K = 1, 2, ..., 10$) for competing methods.