Learning to Group and Label Fine-Grained Shape Components
Supplementary Material

Xiaogang Wang, Bin Zhou, Haiyue Fang, Xiaowu Chen, Qinping Zhao, Kai Xu*

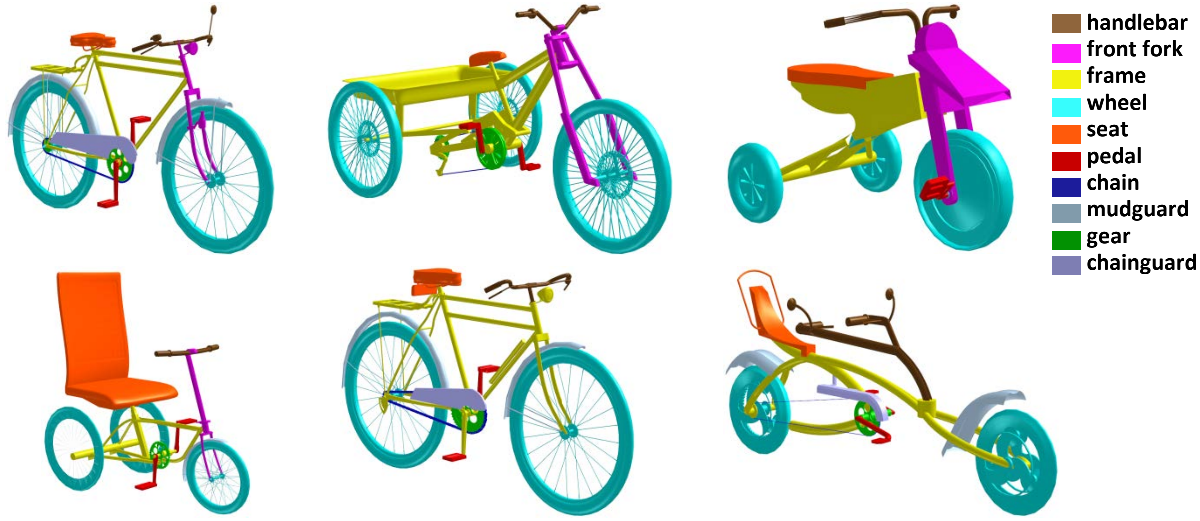
Part 1. Statistic on model components for all categories of ShapeNetCore (57452 models in 57 categories).

Category ID	# Models (Multi-comp.)	# Avg. Comp. (Multi-comp.)	# Models (Single-comp.)	# Models (Total)	# Avg. Comp. (All)
2691156	4028	107.83	17	4045	107.39
2747177	326	32.28	17	343	30.73
2773838	83	22.55	0	83	22.55
2801938	103	32.18	10	113	29.42
2808440	789	14.36	68	857	13.3
2818832	251	33.76	3	254	33.38
2828884	1747	42.04	69	1816	40.48
2834778	59	1020.44	0	59	1020.44
2843684	67	11.6	6	73	10.73
2858304	1118	89.11	19	1137	87.64
2871439	414	33.9	52	466	30.23
2876657	431	6.55	67	498	5.81
2880940	130	11.54	56	186	8.37
2924116	939	121.6	0	939	121.6
2933112	1533	34.32	39	1572	33.5
2942699	112	24.66	1	113	24.45
2946921	98	7.89	10	108	7.25
2954340	55	20.31	1	56	19.96
2958343	7497	370.92	0	7497	370.92
2992529	521	29.54	6	527	29.22
3001627	6432	27.33	346	6778	25.99
3046257	647	32.51	8	655	32.12
3085013	65	122.12	0	65	122.12
3207941	92	34.09	1	93	33.73
3211117	1090	20	5	1095	19.91
3261776	72	15.99	1	73	15.78
3325088	734	23.17	10	744	22.87
3337140	287	45.54	11	298	43.89
3467517	796	71.24	1	797	71.16
3513137	159	13.23	3	162	13.01
3593526	502	41.14	95	597	34.75

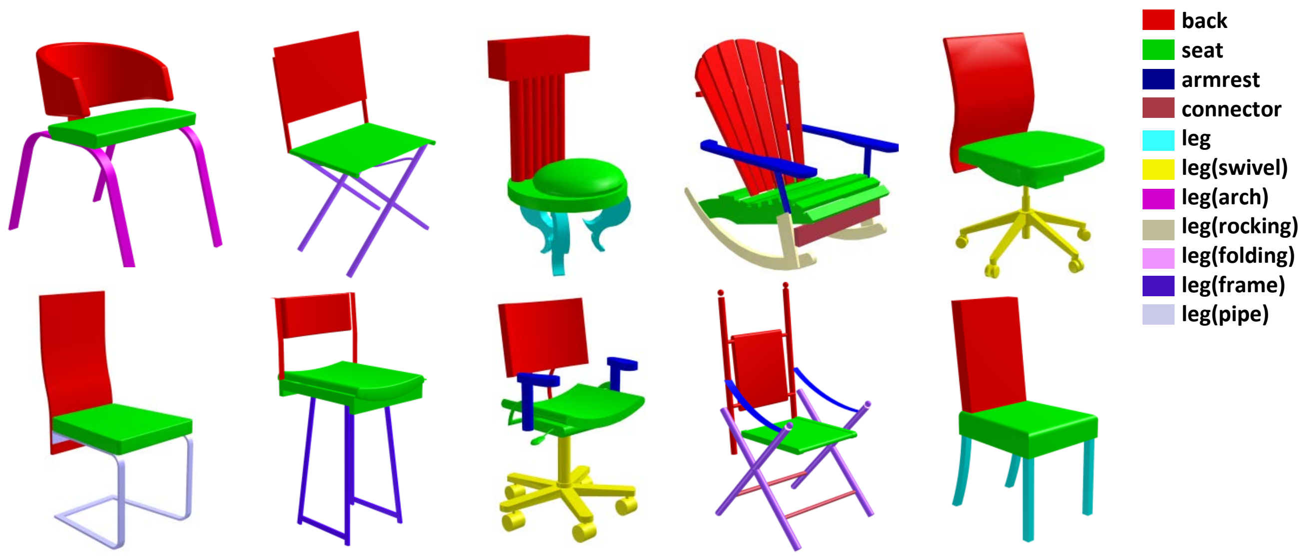
3624134	408	8.2	16	424	7.93
3636649	2293	26.43	25	2318	26.16
3642806	456	56.35	4	460	55.87
3691459	1592	26.13	26	1618	25.73
3710193	91	21.7	3	94	21.04
3759954	66	11.92	1	67	11.76
3761084	152	39.17	0	152	39.17
3790512	337	156.8	0	337	156.8
3797390	199	5.02	15	214	4.74
3928116	239	70.99	0	239	70.99
3938244	70	3.36	26	96	2.72
3948459	300	20.66	7	307	20.21
3991062	537	357.82	65	602	319.29
4004475	161	24.07	5	166	23.37
4074963	66	40.2	1	67	39.61
4090263	2347	32.56	26	2373	32.21
4099429	85	70.95	0	85	70.95
4225987	149	34.46	3	152	33.8
4256520	3100	22.2	73	3173	21.71
4330267	217	38.51	1	218	38.33
4379243	8143	18.21	366	8509	17.47
4401088	1042	27.62	10	1052	27.37
4460130	130	75.47	3	133	73.79
4468005	389	151.53	0	389	151.53
4530566	1911	101.44	28	1939	99.99
4554684	166	35.06	3	169	34.46

Part 2. An overview of our Multi-Component Labeling (MCL) benchmark dataset (eight object categories and two scene categories).

Bicycle:

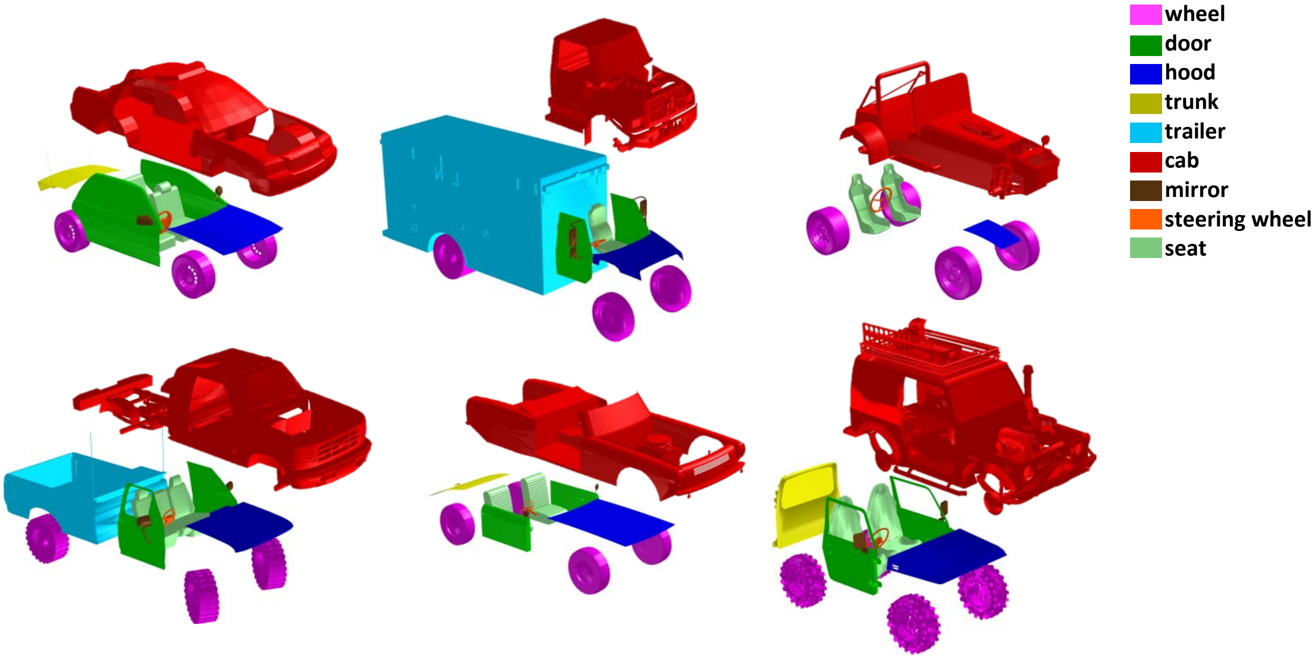


Chair:

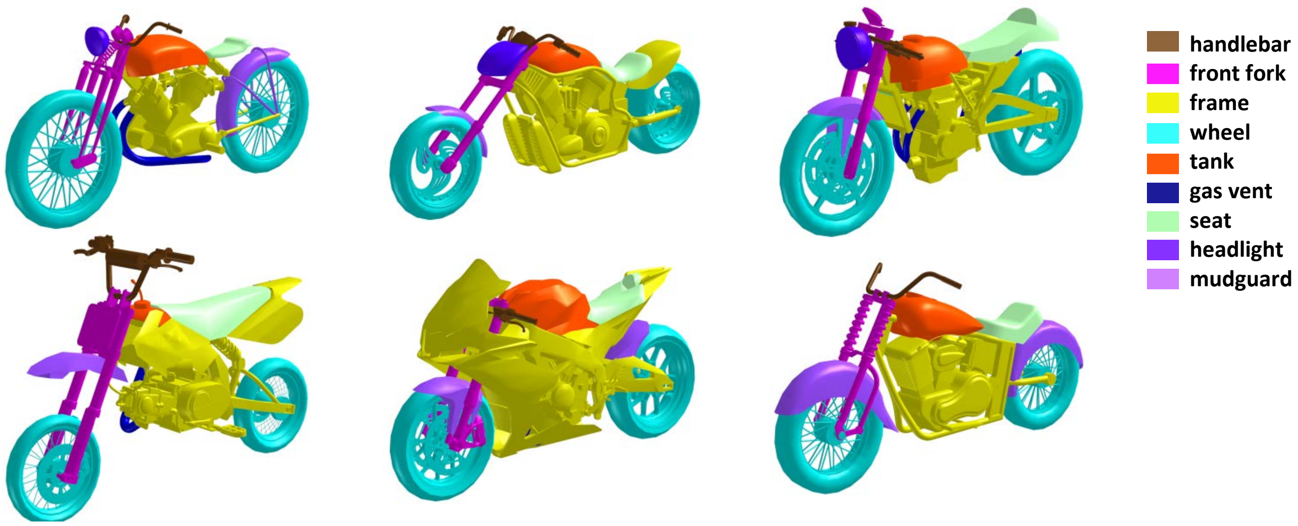


Cabinet:

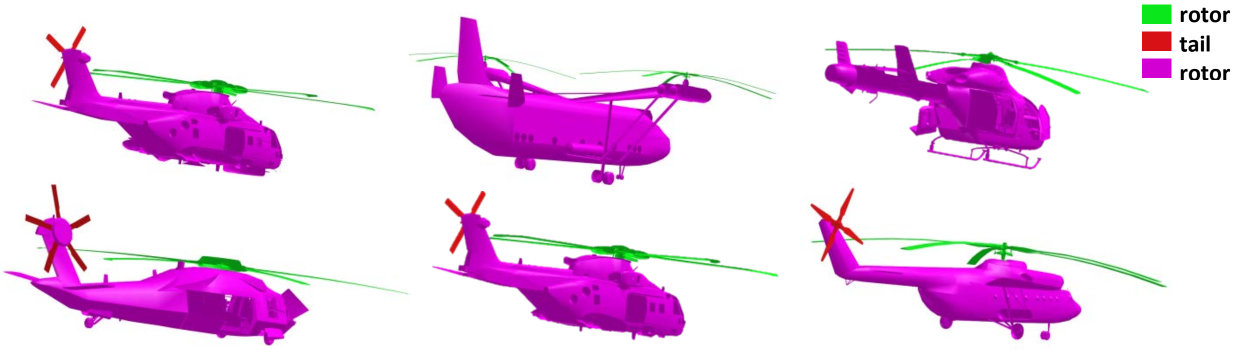
Vehicle:



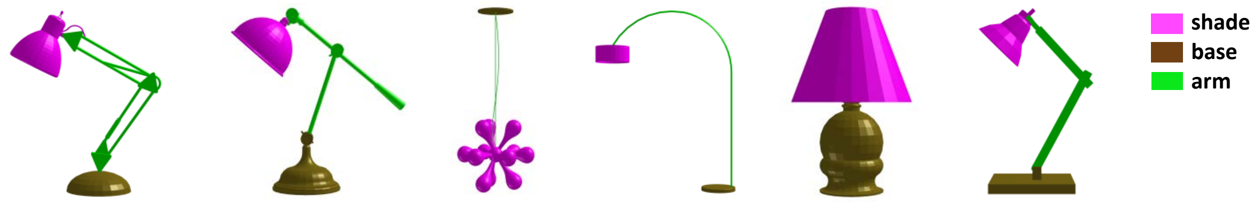
Motor:



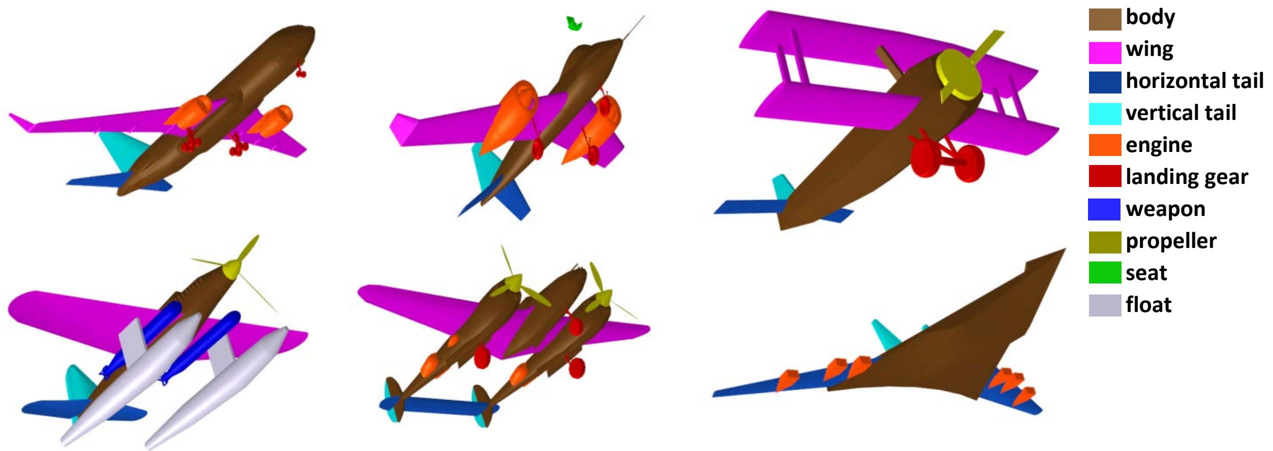
Helicopter:



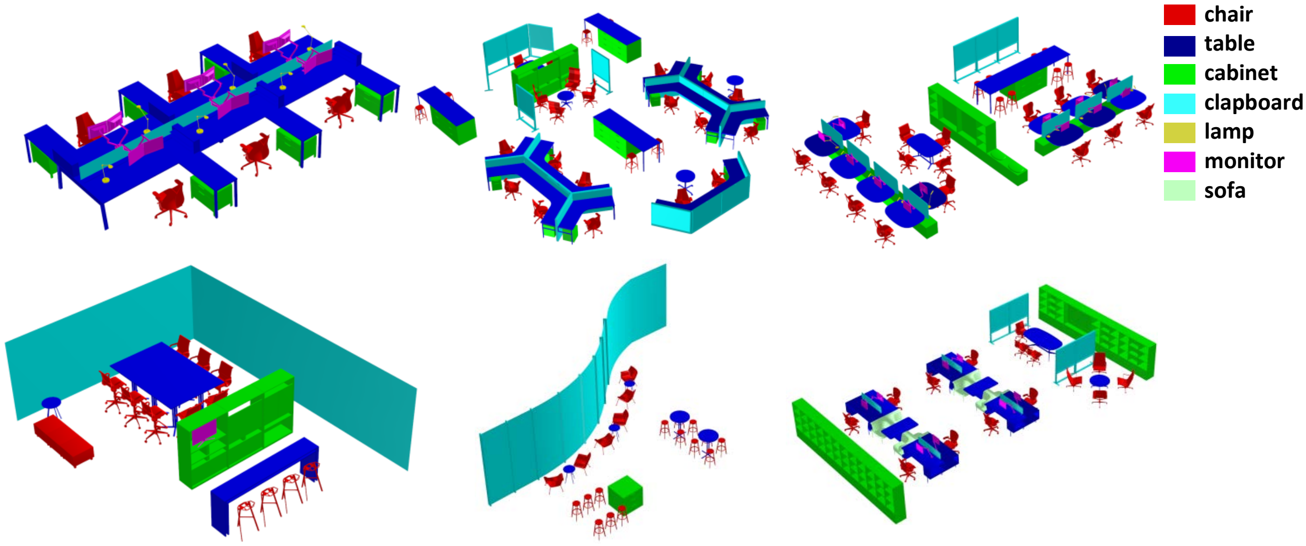
Lamp:



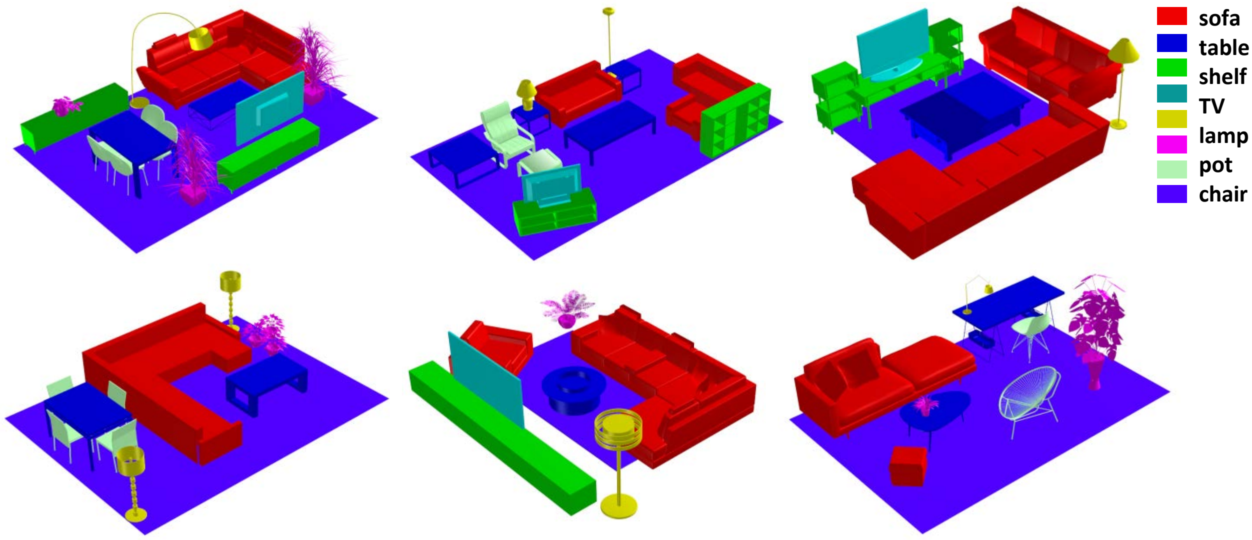
Plane:



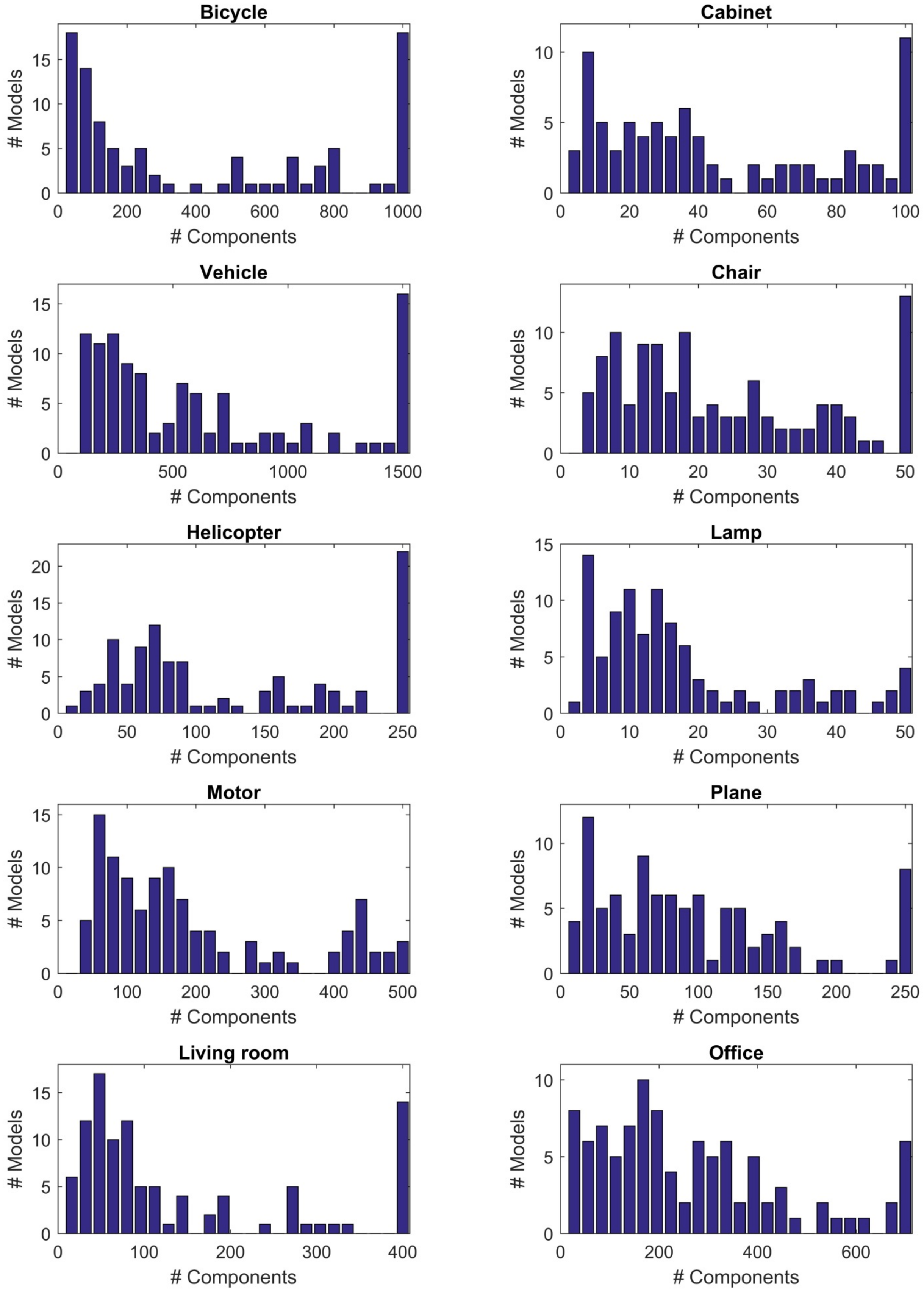
Office:



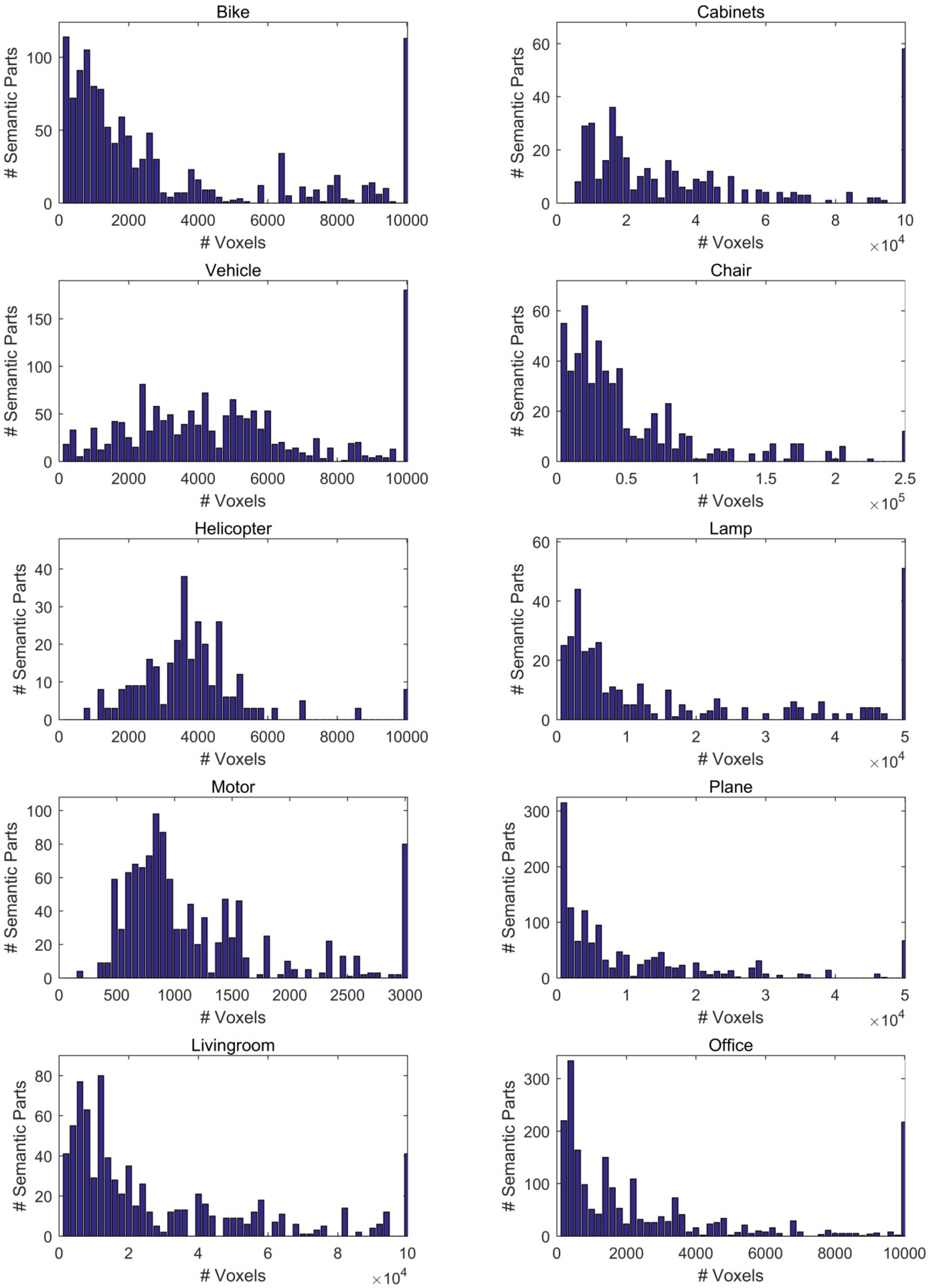
Living room:



Part 3. Statistics on components for Multi-Component Labeling (MCL) benchmark dataset.



Part 4. Statistics on semantic part size for Multi-Component Labeling (MCL) benchmark dataset.



Part 5. Baseline method – CNN-based hypothesis generation – network architecture.

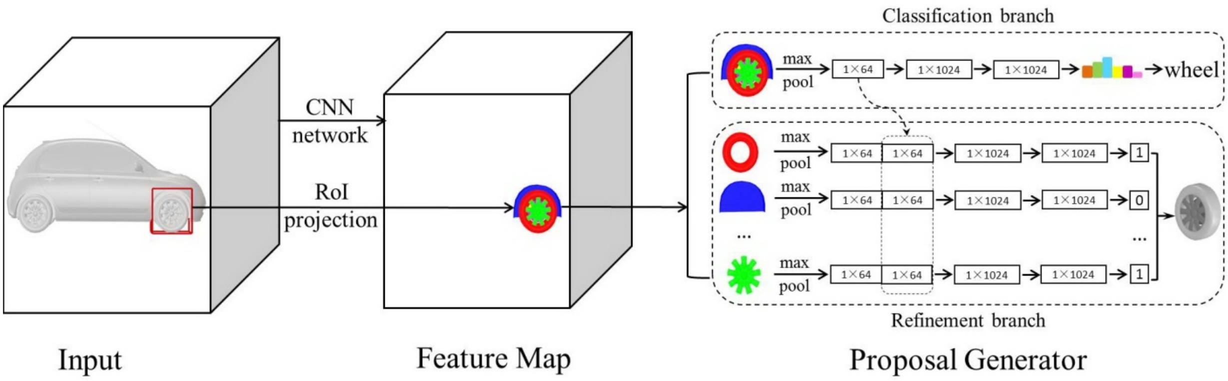


Figure 1. CNN-Based Hypothesis Generation Network. The network takes a complete shape as input. Then multi-scale boxes are applied to produce candidate regions on different scales and align corresponding feature maps. Classification and Refinement branches are responsible for classifying and refining candidate regions respectively. In which, classification branch outputs a discrete probability distribute for each candidate region over K+1 categories. Refinement branch takes each component in current proposal as input, and outputs a probability distribute over two categories (is adopted by current proposal vs. not). The architecture is trained end-to-end with a multi-task loss.

Inspired by Fast-RCNN, we designed a network architecture (CNN Hypothesis Generation Network) to generate proposals by end-to-end, see Figure 1.

For a given 3D CAD shape, we first convert it to the volumetric representation as a occupancy grid with resolution 64 *64 *64. The CNN network consists of five 3D convolution layers. For all convolution layers, the kernel size is 2*2*2 , and stride is 1, with numbers of channels {32,32,32,32,64}, respectively. We also add Batch normalization and ReLU layers between convolutional layers.

For each occupancy voxel location, we will predict N candidate proposals. Each of the proposals corresponds to one of the N boxes with various sizes. In our case, based on statistics of semantic parts sizes in our dataset, we define a set of N=20 boxes. Note that, our proposal is not a regular cube region, but the region that related components covered in this box.

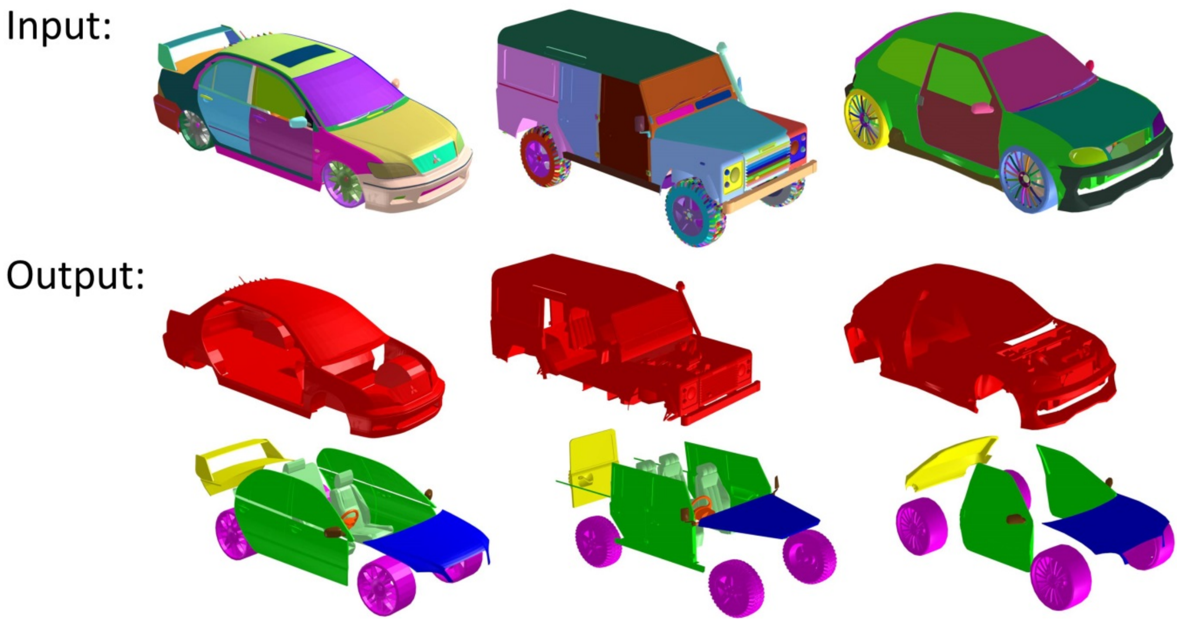
Then, classification and refinement branches are responsible for classifying and refining proposals respectively. For classification branch, each proposal is pooled into a fixed-size feature vector by max-pooling, and then mapped to a feature vector by two fully connected layers (FCs). This branch outputs a discrete probability distribution (per proposal), $p=(p0,...,pK)$, over K+1 categories. Refinement branch takes each component in current proposal as input, and outputs a probability distribute, $b=(b0, b1)$, over two categories (is adopted by current proposal vs. not).

Each training proposal is labeled with a ground-truth semantic class u, and each component in proposal has a binary label v, which represents whether the component should be adopted by the proposal. We use a multi-task loss L on each labeled proposal to jointly train for classification and refinement:

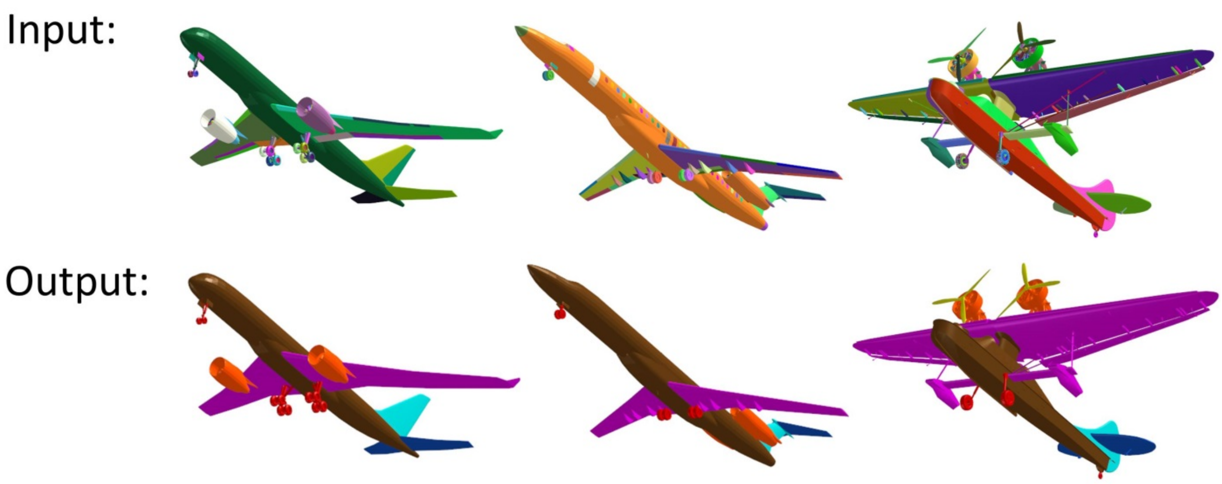
$$L(p,u,b,v) = L_{cls}(p,u) + \sum_{c \in h} L_{mask}(b,v)$$

Where $L_{cls}(p,u) = -\log p_u$ is cross-entropy loss for label u . And for each component c in current proposal h, $L_{mask}(b,v) = -\log b_v$ is log loss over two categories (is adopted by proposal vs. not).

Part 6. More results on the CAD models from the INRIA GAMMA database.




Input:
Output:


Input:
Output:

Part 7. Interactive annotation tool.

First, we directly load the original model file 'xxx.obj', and extract all components information from the model by identifying the identification information 'g ….' (see Figure 1), and display it in our interactive tool, as shown in Figure 2.

Then, we create an empty folder called 'aa-bb', where the 'aa' represents semantic category, such as 'wheel'. And 'bb' means which one, because of each semantic category is likely to have more than one semantic part, such as the current vehicle consists of four wheels, therefore, 'bb' means which wheel, see Figure 3. Meanwhile, move all components associated with this semantic part into the current folder, see Figure 4. Finally, we Repeat this process for other semantic parts, as shown in Figure 5.

```
# Alias OBJ Model File
# Exported from SketchUp, (c) 2000-2012 Trimble Navigation Limited
# File units = inches
mtllib model.mtl
g Mesh1 LROVER_BL0_1 skp4B4_1 Model
usemtl _GLOBAL_6
v 0.123658 -0.103775 0.0317404
vt -486.183 423.498
vn 0.577350 -0.577350 -0.577350
v 0.123658 0.0231924 0.0317404
vt -486.183 1023.5
vn 0.577350 0.577350 -0.577350
v 0.282367 0.0231924 0.0317404
vt -1236.18 1023.5
vn 0.577350 0.577350 0.577350
v 0.282367 -0.103775 0.0317404
vt -1236.18 423.498
vn 0.577350 -0.577350 0.577350
f 2/2/2 3/3/3 4/4/4
f 1/1/1 2/2/2 4/4/4
v 0.123658 -0.103775 -0.031744
vt 149.999 423.498
vn -0.577350 -0.577350 -0.577350
v 0.123658 0.0231924 -0.031744
vt 149.999 1023.5
```

Figure 1. Original model file 'xxx. obj'

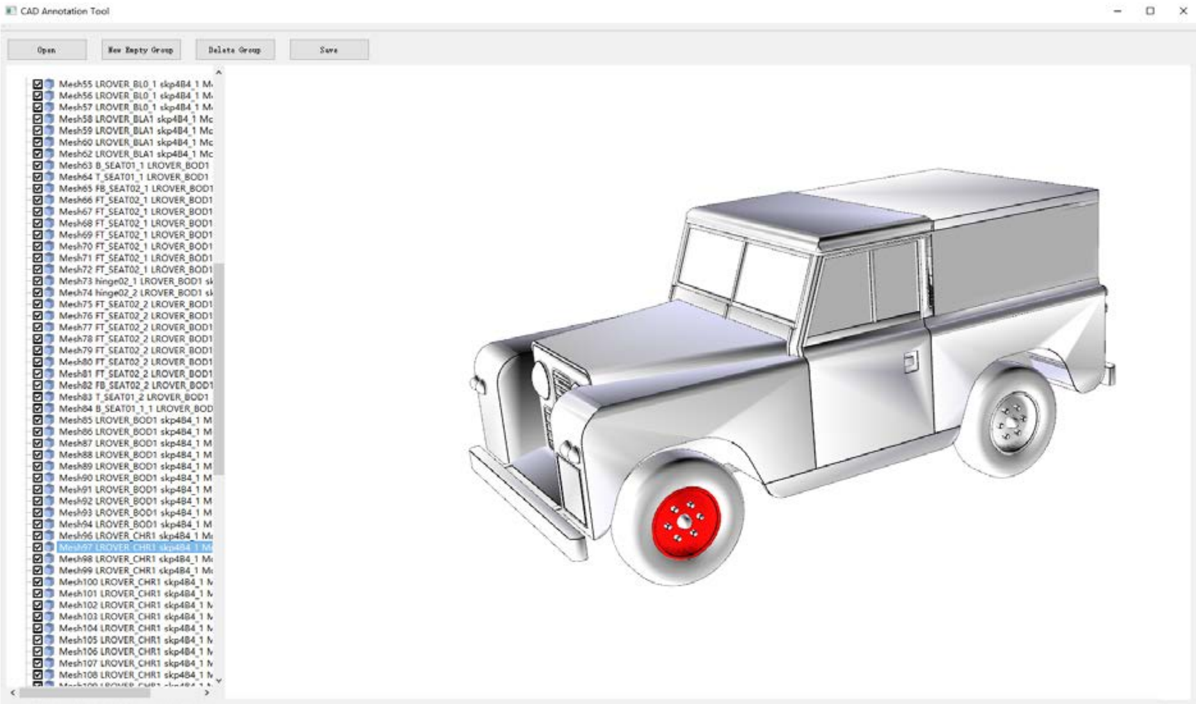


Figure 2． Load model file into interactive tool

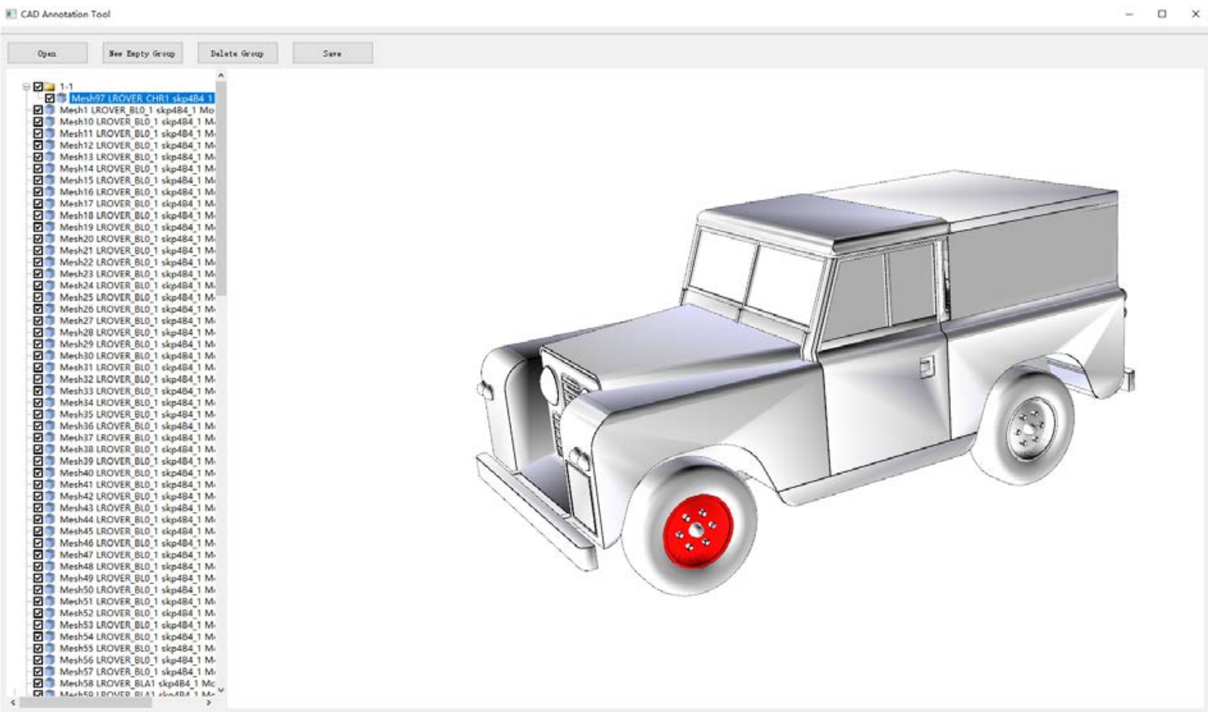


Figure 3． Create an empty folder called ‘aa-bb’

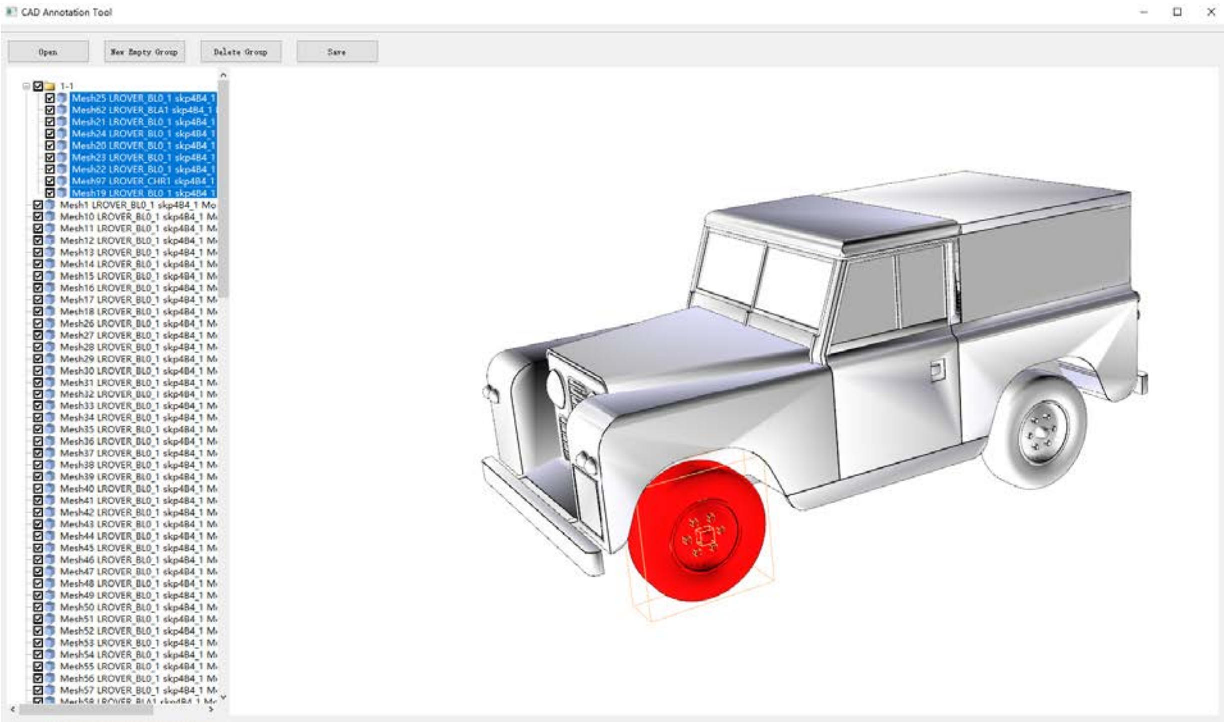


Figure 4． Move all components associated with this semantic part into the current folder

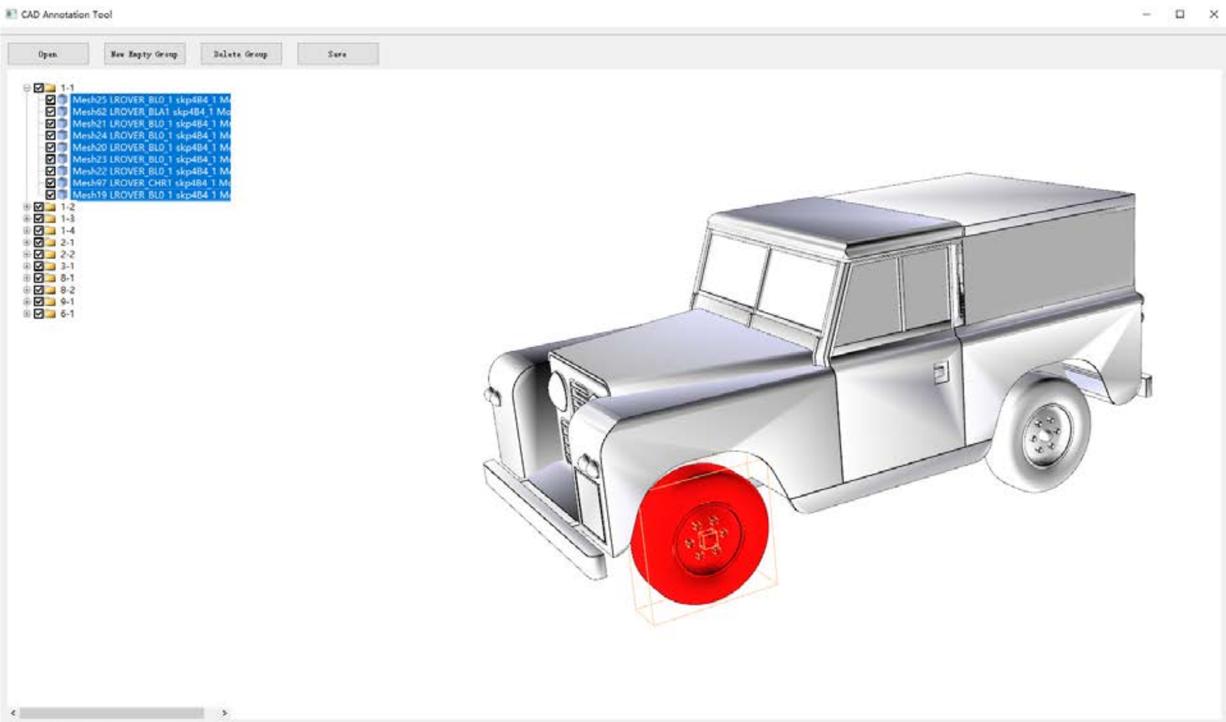


Figure 5． Repeat this process for all the other semantic parts

Part 8. Plots for the remaining four semantic categories (with correspondence to the figures in paper).

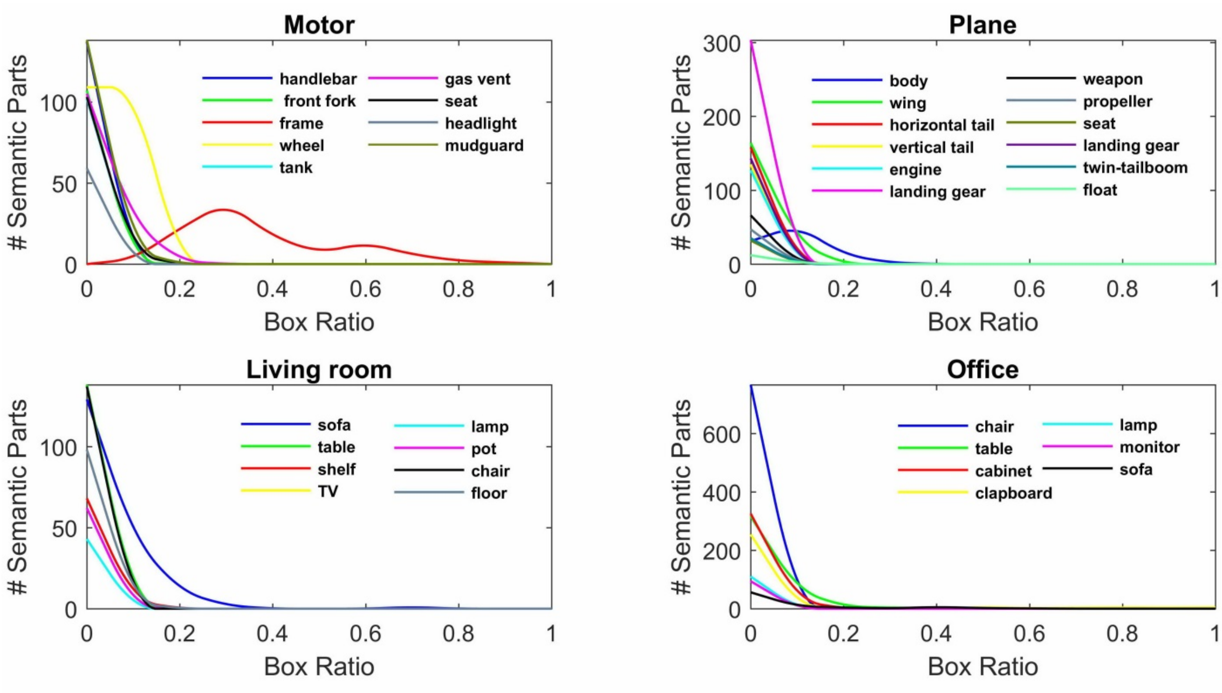


Figure 5 in paper. The occupancy ratio of the bounding box of varying number of semantic parts over the entire model. The statistics are performed with our benchmark dataset.

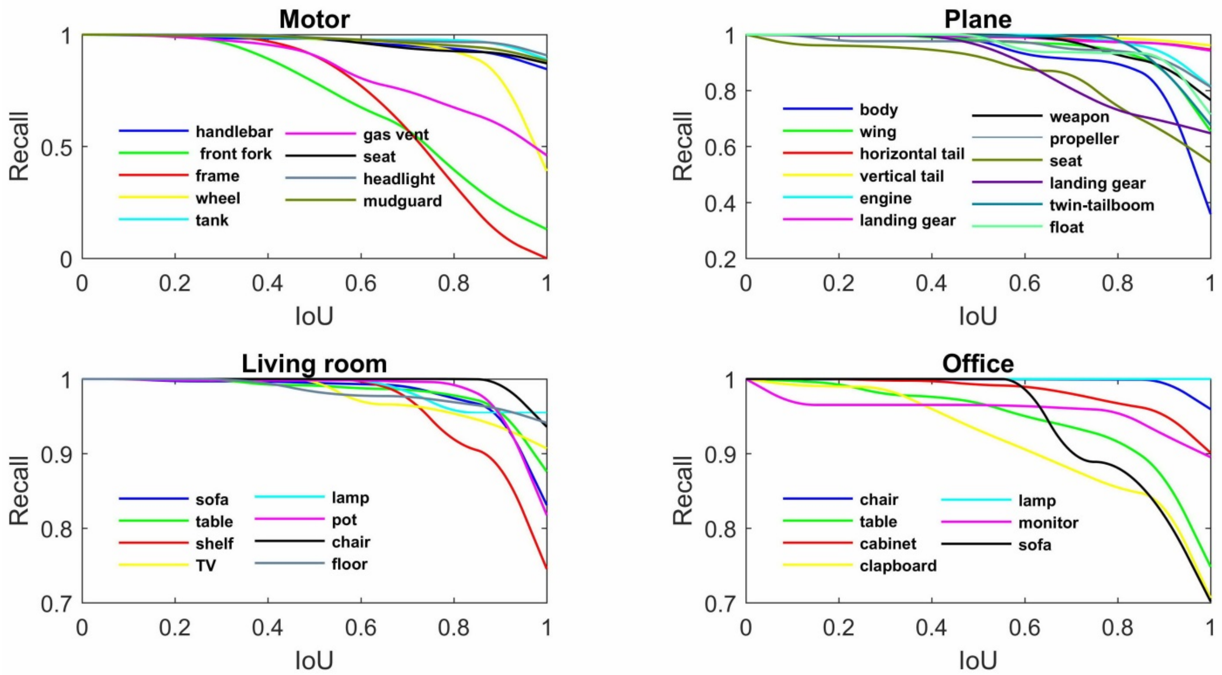


Figure 8 in paper. Performance (recall rate over IoU) of part hypothesis generation in all object/scene categories.

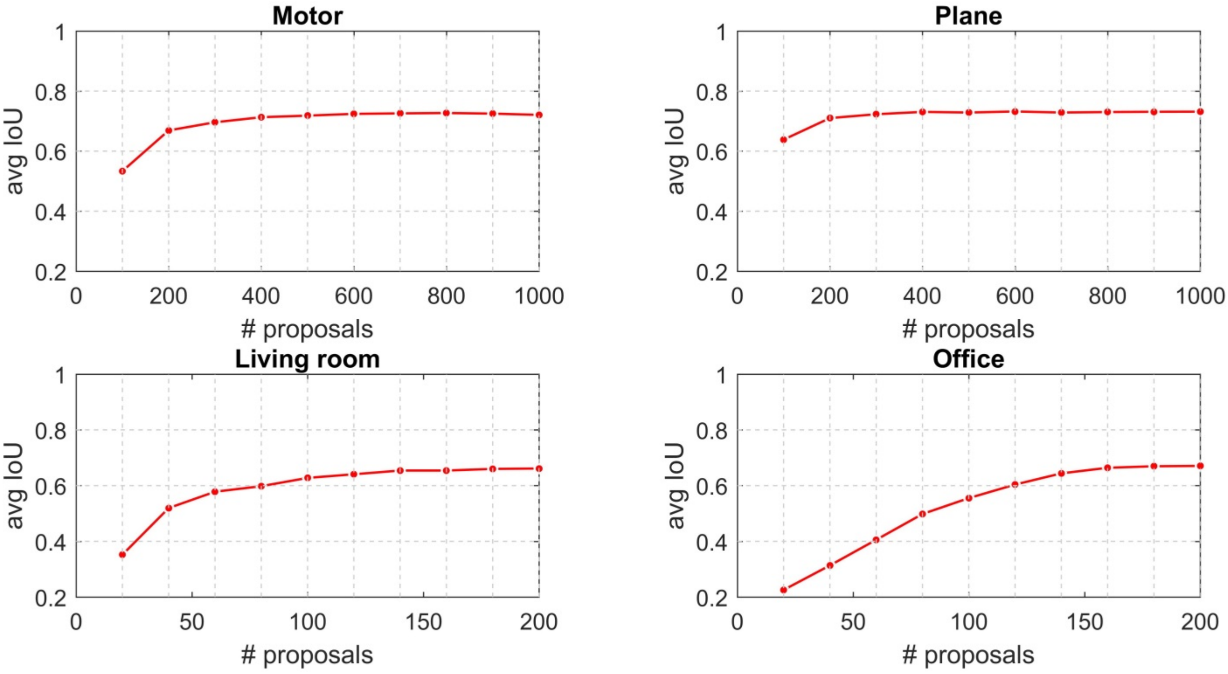


Figure 12 in paper. Labeling accuracy (average IoU) vs. number of part hypotheses.

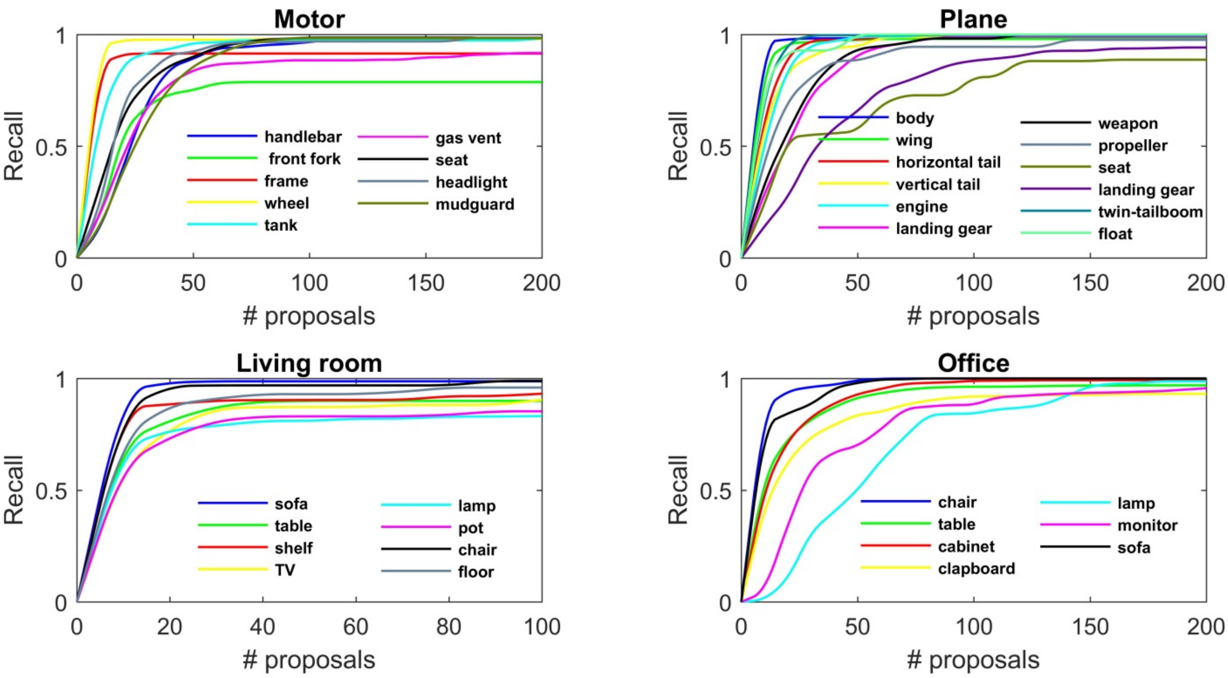


Figure 13 in paper. Recall rate on semantic parts over varying number of part hypotheses, when IoU against ground-truth is fixed to 0.5, tested on our benchmark dataset.

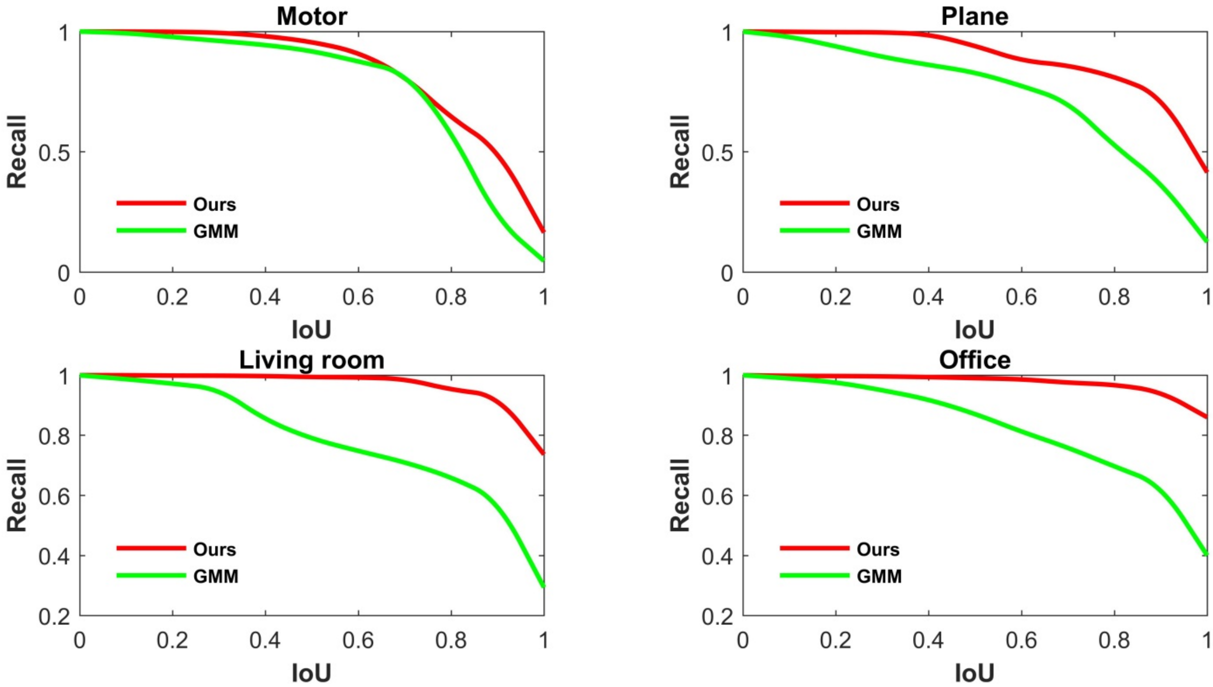


Figure 14 in paper. Performance (recall rate vs. average IoU) comparisons between our hierarchical grouping algorithm and the GMM-based baseline method over all object/scene categories.