CAPRI-Net: Learning Compact CAD Shapes with Adaptive Primitive Assembly

Fenggen Yu[1] Zhiqin Chen[1] Manyi Li[1,3] Aditya Sanghi[2] Hooman Shayani[2]
Ali Mahdavi-Amiri[1] Hao Zhang[1]
[1]Simon Fraser University [2]Autodesk AI Lab [3]Shandong University

1. Reconstruction from Voxels

We present more qualitative and quantitative results on shape reconstruction from voxels in Figures 4 and 5 and Tables 1 and 2. We also show some results without fine-tuning from BSP-Net [1], UCSG [4], CSG-Stump [3] and our method. These results are obtained by testing with the pre-trained networks, and we call them w.o FT. The results show that after fine-tuning, we obtain better output shapes that are more consistent with the input and attain more details.

Methods	CD ↓	NC ↑	ECD ↓	LFD ↓
BSP w.o FT	1.19	0.78	19.93	3050.2
UCSG w.o FT	2.29	0.66	8.91	4532.8
STUMP w.o FT	2.22	0.71	9.32	4128.8
Ours w.o FT	0.80	0.80	14.21	2801.3
BSP	0.49	0.87	10.10	1342.7
UCSG	0.30	0.88	5.02	1494.8
STUMP	1.18	0.83	11.85	2945.2
Ours	**0.14**	**0.91**	**2.21**	**800.2**

Table 1. Comparing 3D reconstruction from voxels on ABC.

Methods	CD ↓	NC ↑	ECD ↓	LFD ↓
BSP w.o FT	0.36	0.84	5.13	2562.8
UCSG w.o FT	1.75	0.75	5.02	4360.9
STUMP w.o FT	4.46	0.74	10.40	5929.6
Ours w.o FT	0.32	0.85	10.11	2663.7
BSP	0.22	0.87	2.11	2254.4
UCSG	1.32	0.82	5.23	3582.5
STUMP	2.29	0.79	10.46	5217.0
Ours	**0.18**	**0.87**	**2.10**	**1824.1**

Table 2. Comparing 3D reconstruction from voxels on ShapeNet.

2. Reconstruction from Point Clouds

Tables 3 and 4 show quantitative results on ABC dataset and ShapeNet. Compared with the state-of-the-art, *unstructured*, non-parametric-surface learning methods such as IM-Net [2] and SIREN [5], CAPRI-Net produces comparable numeric results, but it is slightly worse, due to the trade-off between the reconstruction quality and the desire to obtain a compact primitive assembly. However, in terms of visual quality, as shown in Figure 6 and Figure 7, the results from IM-Net and SIREN often possess geometric artifacts such as small bumps and pits, while the mesh surfaces produced by CAPRI-Net are smoother and more regular.

Methods	CD ↓	NC ↑	ECD ↓	LFD ↓	#P ↓	#C ↓
BSP-Net	0.38	0.90	8.88	920.8	151.1	13.63
UCSG	1.33	0.85	5.76	2848.3	-	12.16
STUMP	6.58	0.85	8.89	4649.1	-	62.35
IM-Net$_{128}$	**0.06**	0.95	2.34	**411.27**	-	-
SIREN$_{128}$	0.07	**0.96**	**1.07**	797.70	-	-
Ours	0.14	0.92	1.57	581.8	**64.06**	**6.82**

Table 3. Comparing 3D point cloud reconstruction on ABC dataset. Both IM-Net and SIREN are at 128^3 resolution.

Methods	CD ↓	NC ↑	ECD ↓	LFD ↓	#P ↓	#C ↓
BSP-Net	0.26	0.89	2.05	2075.6	206.43	17.83
UCSG	4.18	0.77	7.57	5503.2	-	12.46
STUMP	1.52	0.79	5.45	4541.6	-	75.18
IM-Net$_{128}$	1.85	0.86	3.43	2362.8	-	-
SIREN$_{128}$	0.21	**0.92**	1.85	3245.0	-	-
Ours	**0.16**	0.88	**1.83**	**1987.3**	**90.94**	**8.90**

Table 4. Comparing 3D point cloud reconstruction on ShapeNet. Both IM-Net and SIREN are at 128^3 resolution.

3. Post-processing for CAD Shapes

All shapes share the same selection matrix $\mathbf{T}$ during training but different selection matrix $\mathbf{T}$ after fine-tuning, which makes the learned primitives and CSG-tree adaptive to each shape.

After obtaining the primitives $\mathbf{P}$ and selection matrix $\mathbf{T}$, we can assemble primitives into convex shapes and perform CSG operations to output a CAD mesh. Given a primitive selection matrix for each shape, a primitive that has been selected for some convex shapes may not be used in the formation of that convex shape if it falls outside the convex

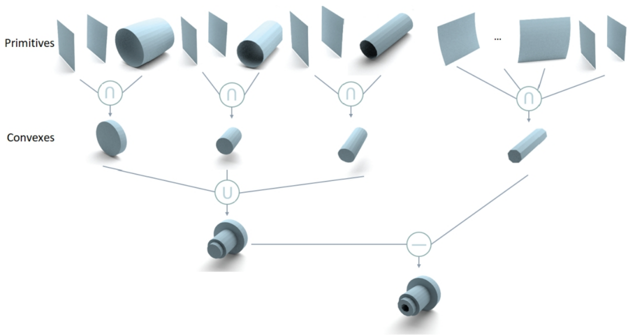

Figure 1. CSG-based CAD mesh outputting process of our method. The output meshes contain sharp edges and regular surfaces as expected from performing the CSG.

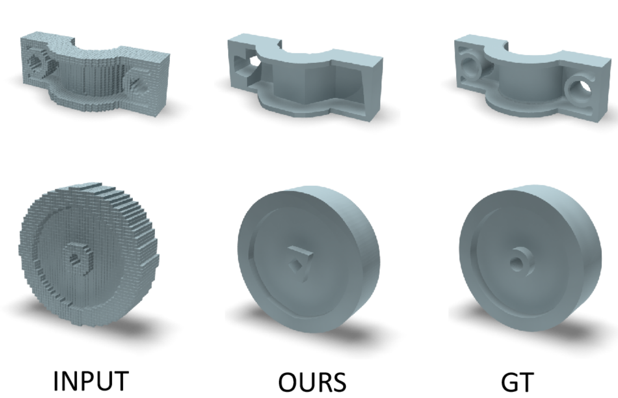

Figure 2. Two failure cases of our method.

shape. Therefore, such primitives do not have an influence on the final shape and should be removed.

To achieve this, we sample some points close to the surface of the reconstructed shape and obtain their occupancy values. Then, we remove each primitive from the list to test whether it changes the occupancy values. If after removing a primitive, the occupancy values of all points remain intact, we discard it from the primitive list. We construct the surface of primitives that are not removed via marching cubes on the signed distance field of each primitive (Top row of Figure 1) and perform CSG operations resulting from our network to reconstruct the final mesh. This process produces shapes with sharp edges and more regular surfaces.

4. Failure cases

Our network follows a *fixed* assembly order, i.e., intersection followed by union and then a single difference operation. As such, not all assemblies, e.g., a nested difference, can be represented. We show two failure cases in Figure 2, and the optimal CSG operations for both two shapes have to use difference before union operation. Our method can only get approximate reconstruction result with difference operation following union for both shapes.

5. CSG Trees

We also provide CSG tree visual comparisons to BSP, UCSG and CSG-Stump, see Figures 8–15. Leaf nodes represent convex shapes in these trees. For BSP and CAPRI-Net, we do not show surface primitives for simplicity. UCSG and CSG-Stump does not use surface primitives but convex, such as boxes and spheres, which are considered as convex shapes. We only show a small subset of primitives used in CSG-Stump considering actual number is large. Besides, CSG-Stump uses primitive inverse layer (indicated as operation C in figure) to achieve shape differ operation for ABC shapes. According to the figure, the shapes produced by our method need fewer convex shapes in comparison with the other methods, which makes our CSG tree more visually compact.

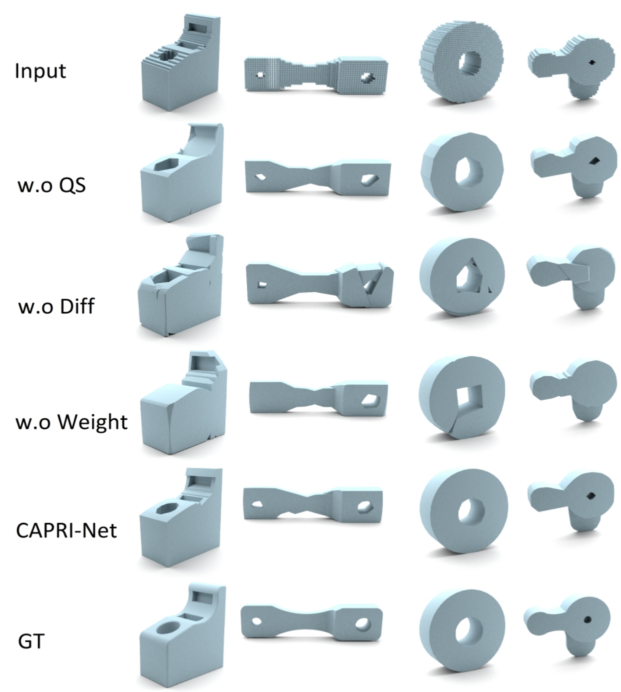

Figure 3. Visual comparison results for ablation study on ABC.

6. Ablation Study

Visual comparison examples of our ablation study is shown in Figure 3 It is apparent that quadric surface representation makes our method suitable for ABC dataset by using fewer appropriate primitives (e.g., cylinders) in the reconstruction. Difference operation can also offer compactness and fewer primitives in the final reconstruction. Finally, the weighted reconstruction loss helps CAPRI-Net reproduce fine details such as small holes.

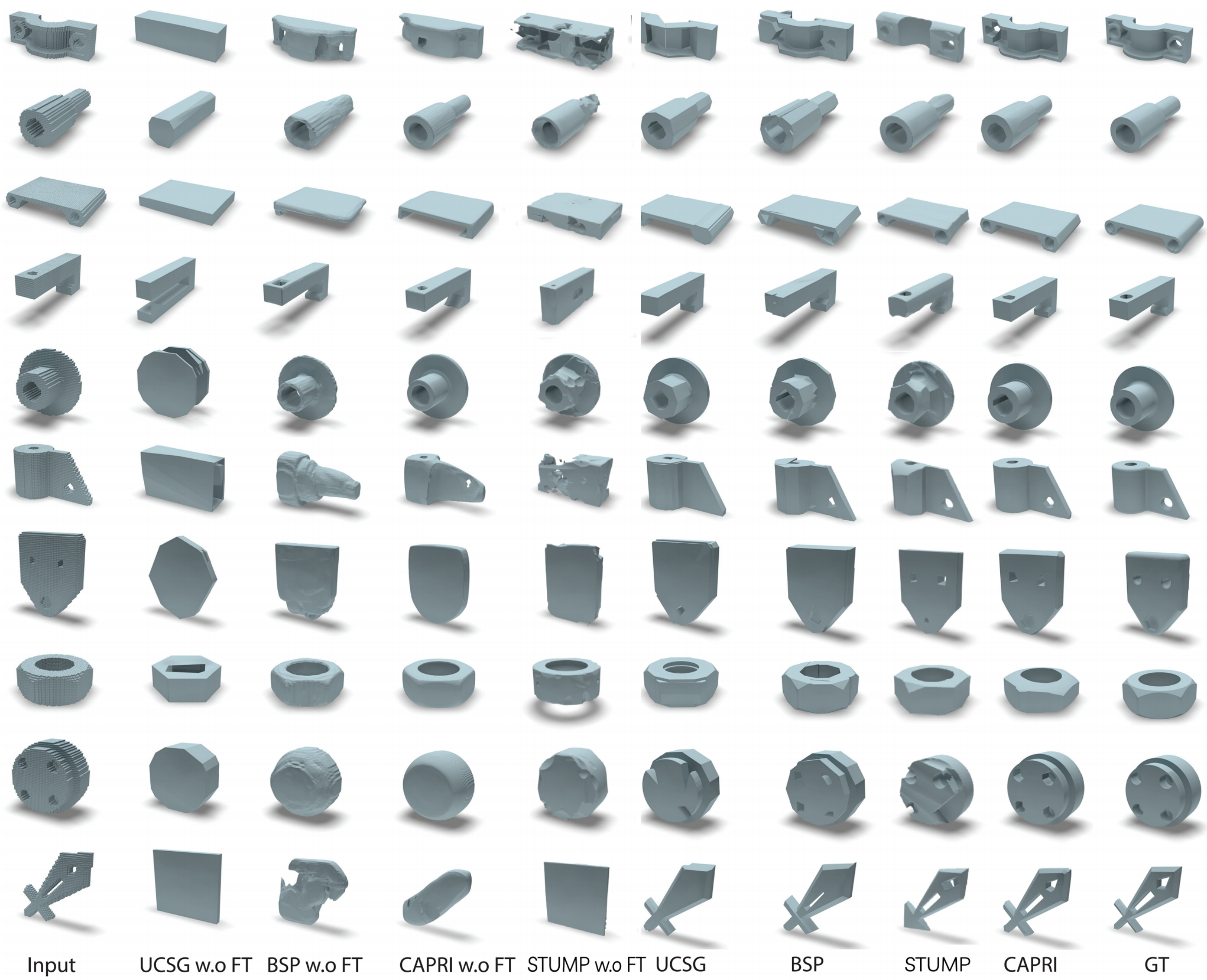

Figure 4. Visual comparisons between reconstruction results from 64^3 voxel inputs on ABC. We also show results before fine-tuning.

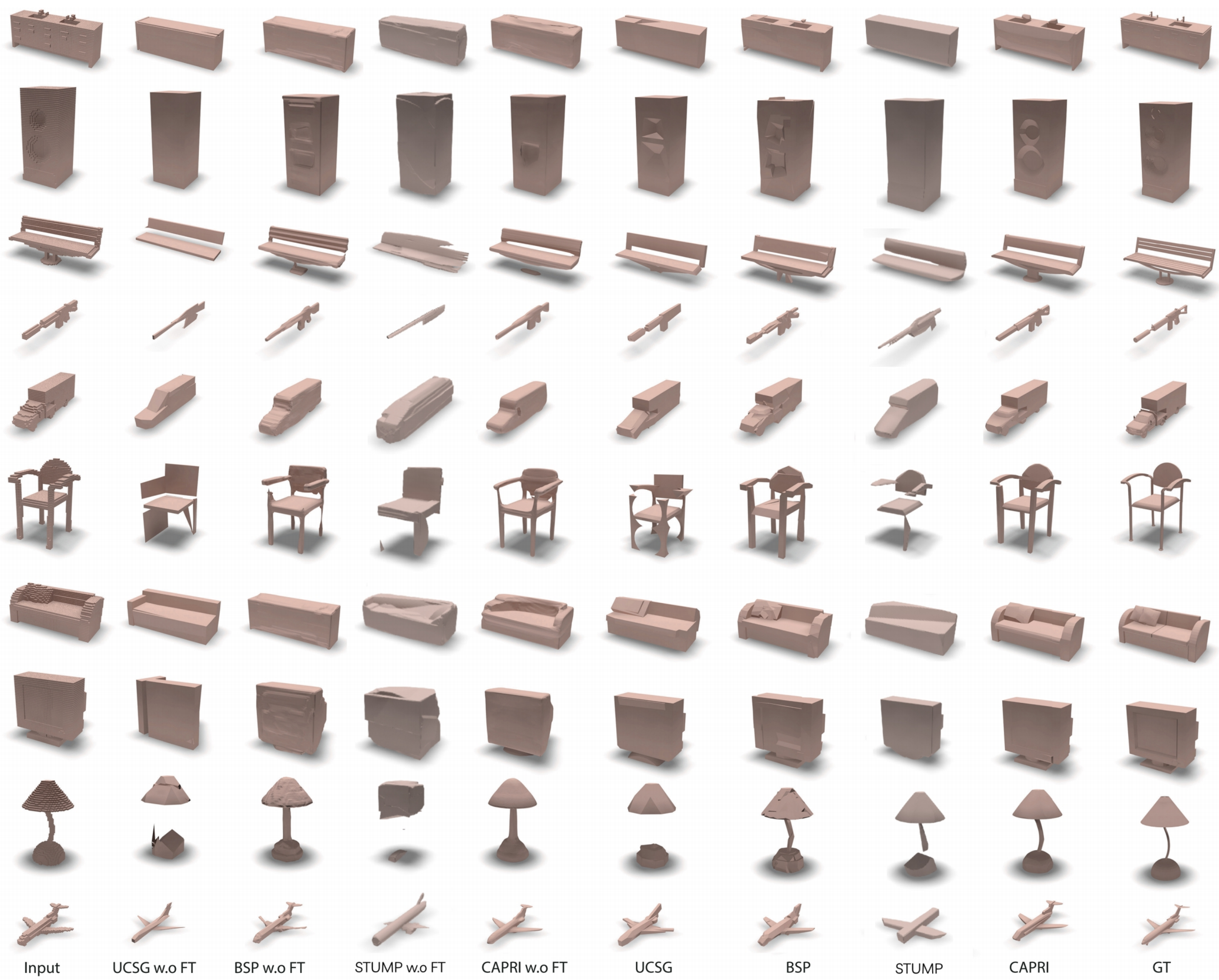

Figure 5. Visual comparisons between some reconstruction results from 64^3 voxel inputs on ShapeNet.

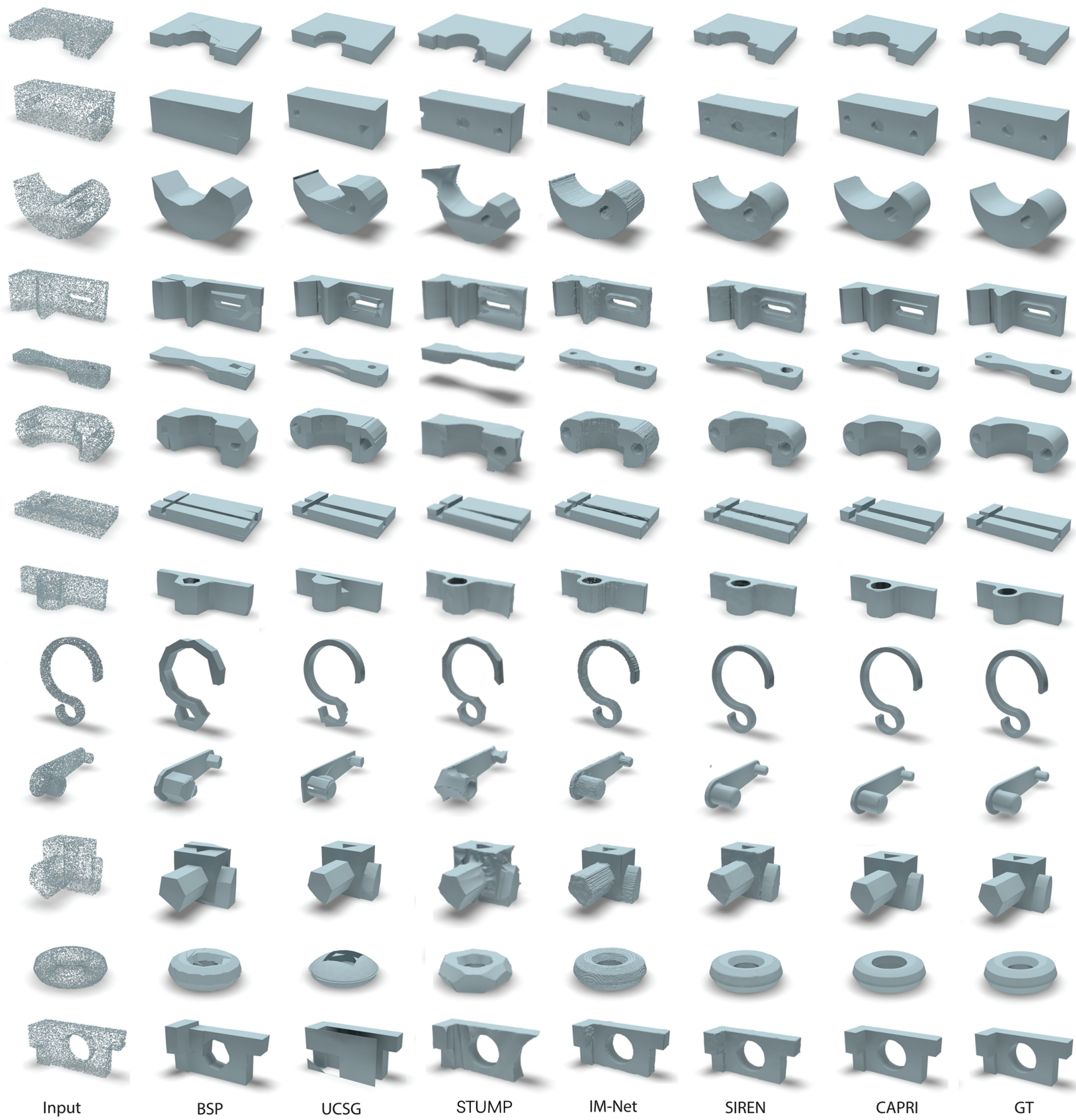

Figure 6. Visual comparisons between reconstruction results from point clouds (8,192 points) on ABC. Pay attention to the surface artifacts from IM-Net and SIREN results, both at 128^3 resolution.

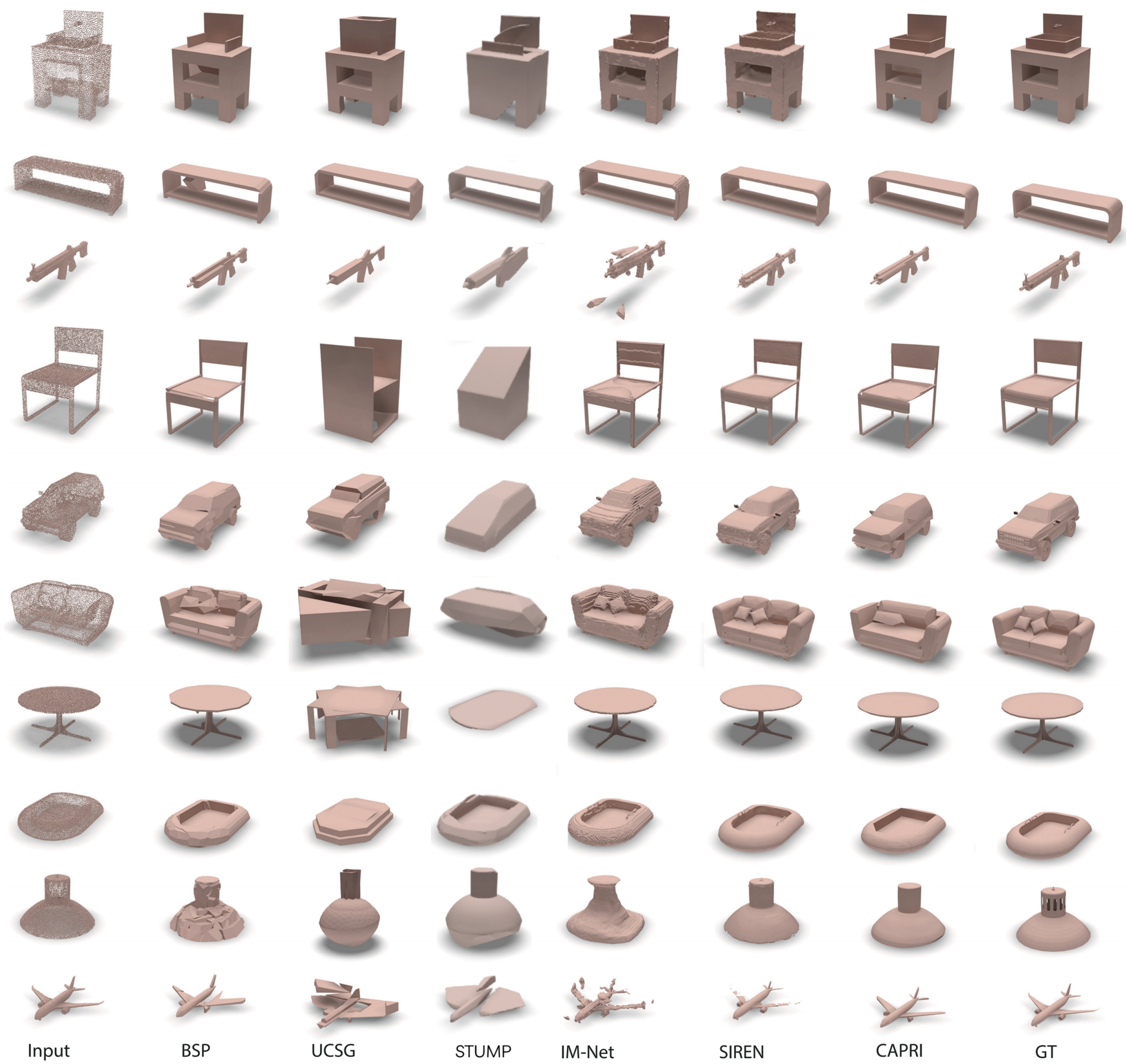

Figure 7. Visual comparisons between reconstruction results from point clouds (8,192 points) on ShapeNet.

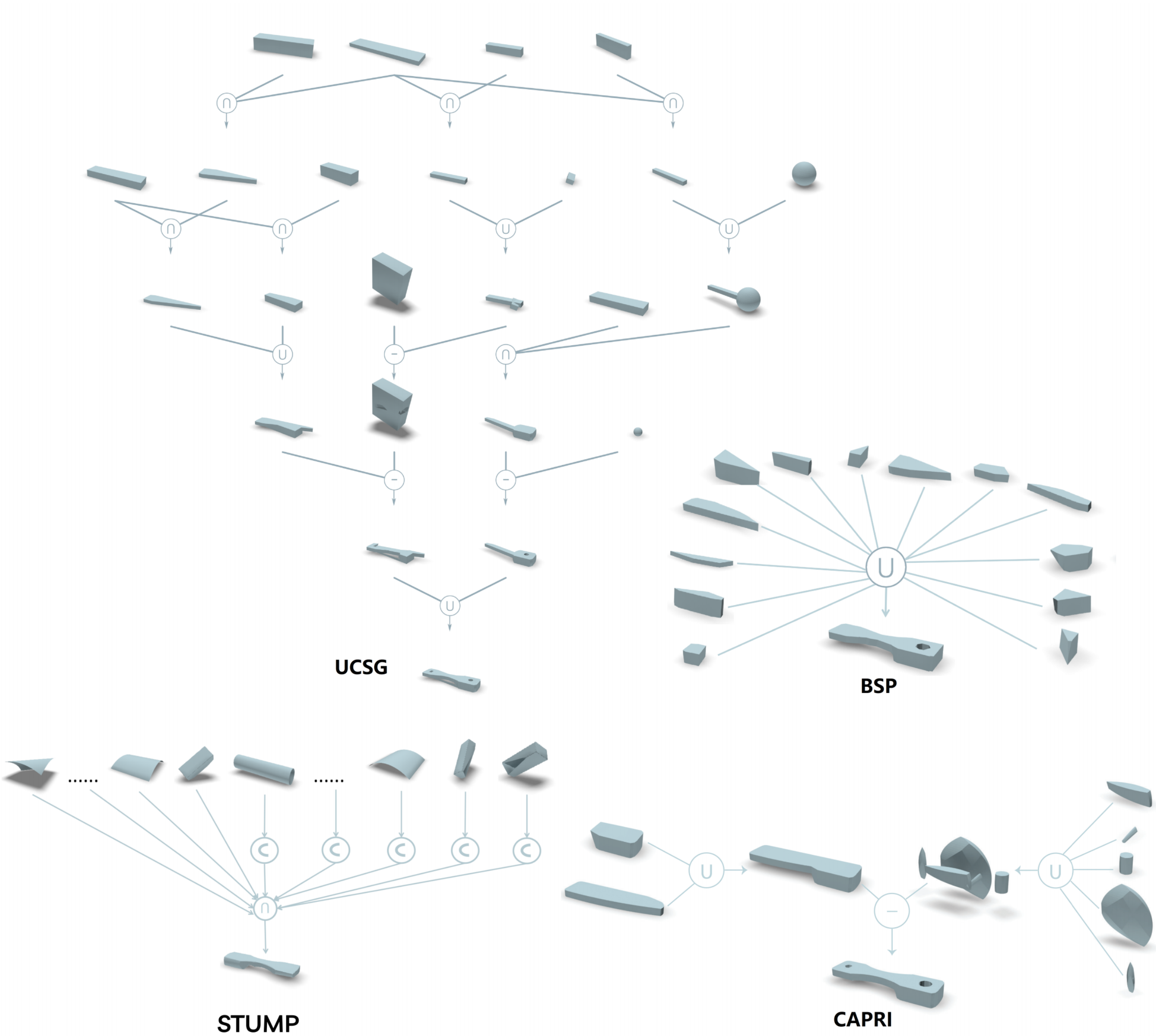

Figure 8. An example of CSG tree visual comparison on ABC.

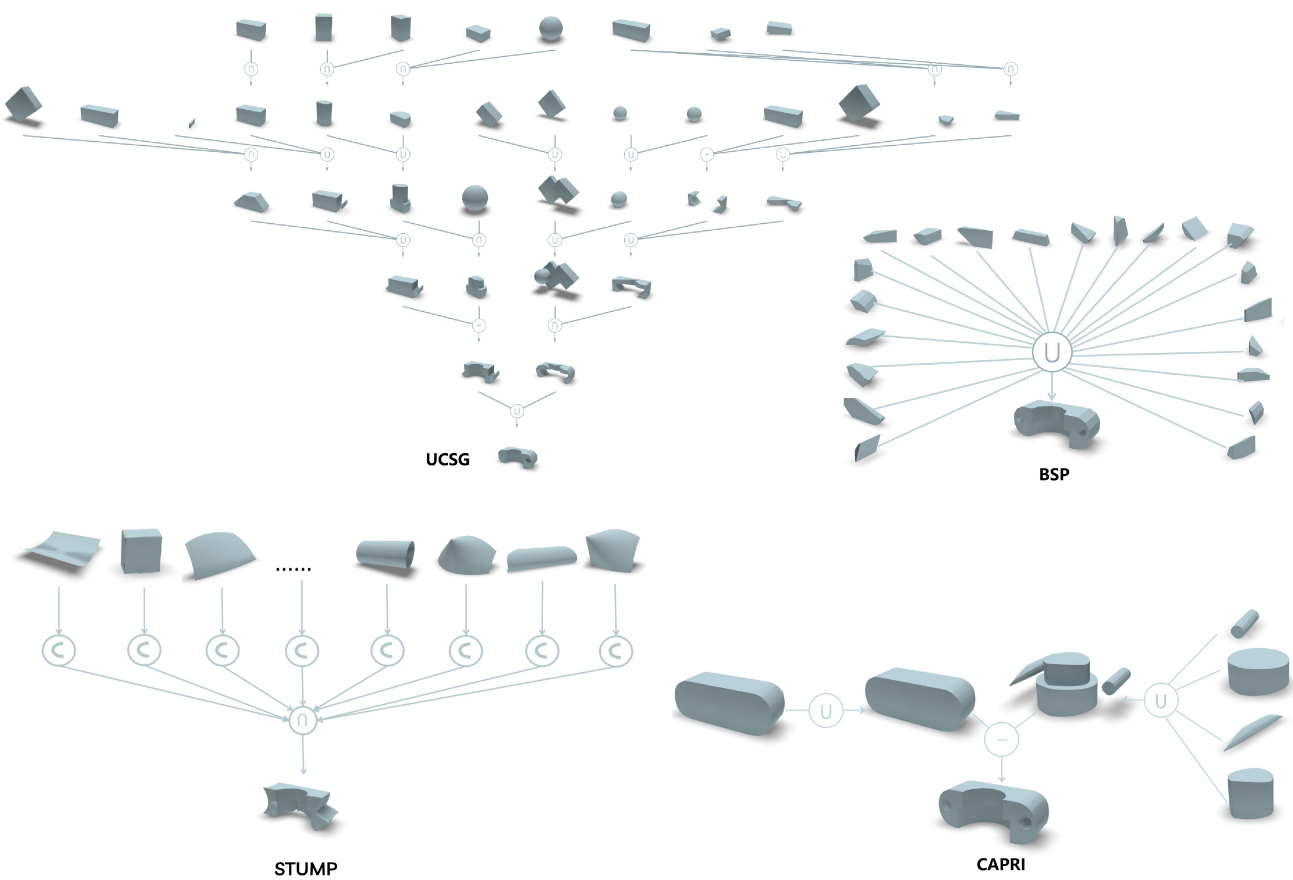

Figure 9. An example of CSG tree visual comparison on ABC.

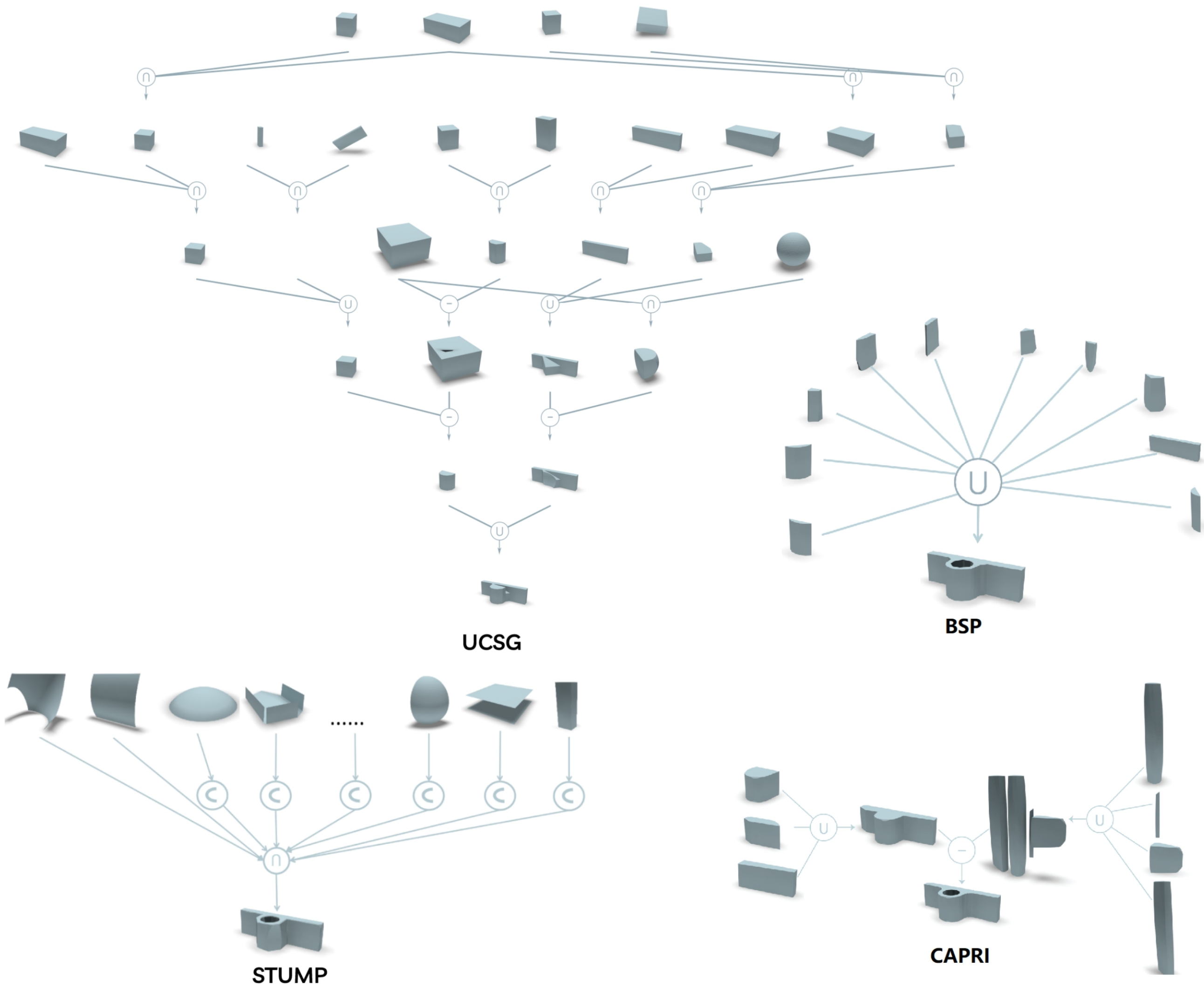

Figure 10. An example of CSG tree visual comparison on ABC.

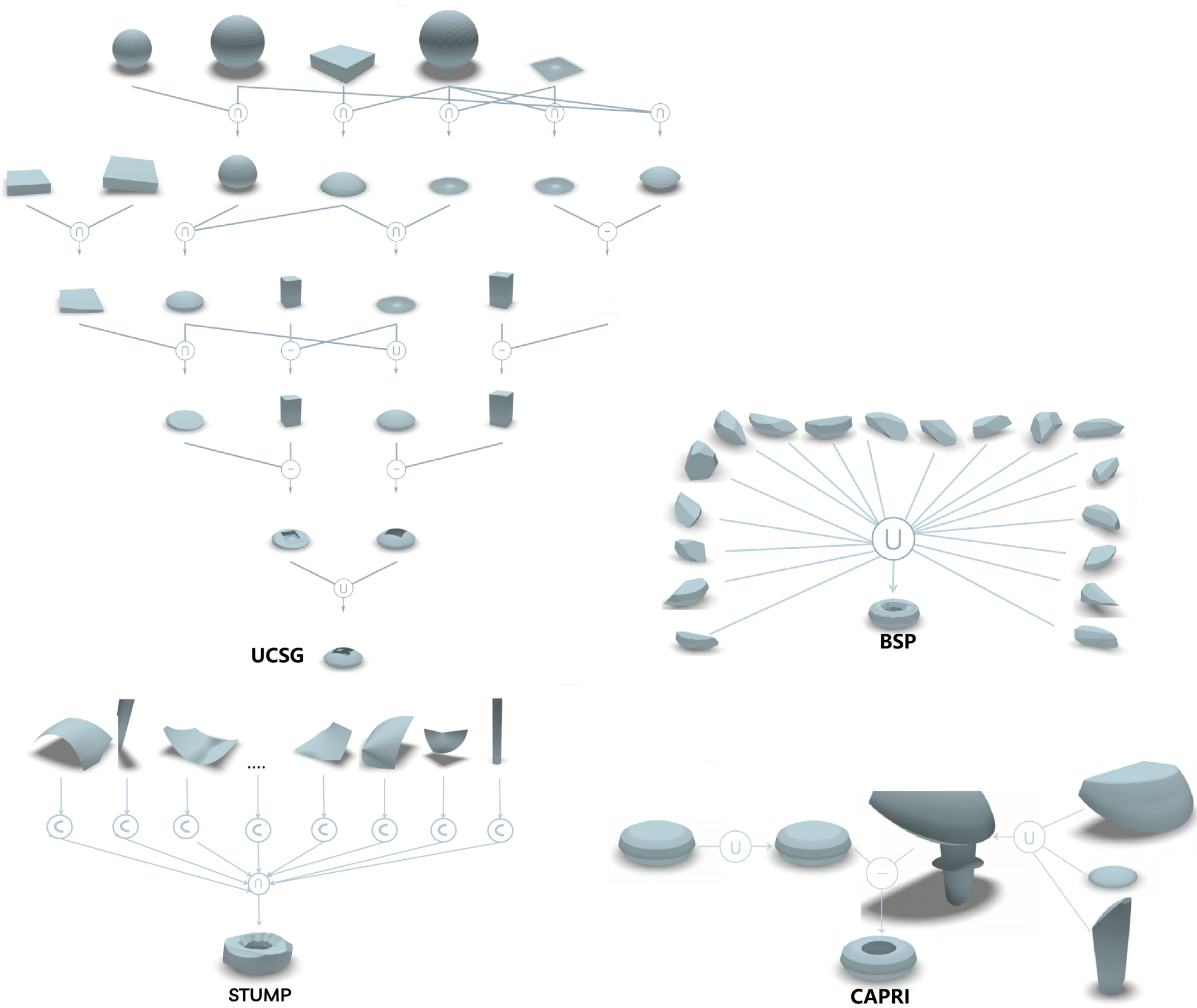

Figure 11. An example of CSG tree visual comparison on ABC.

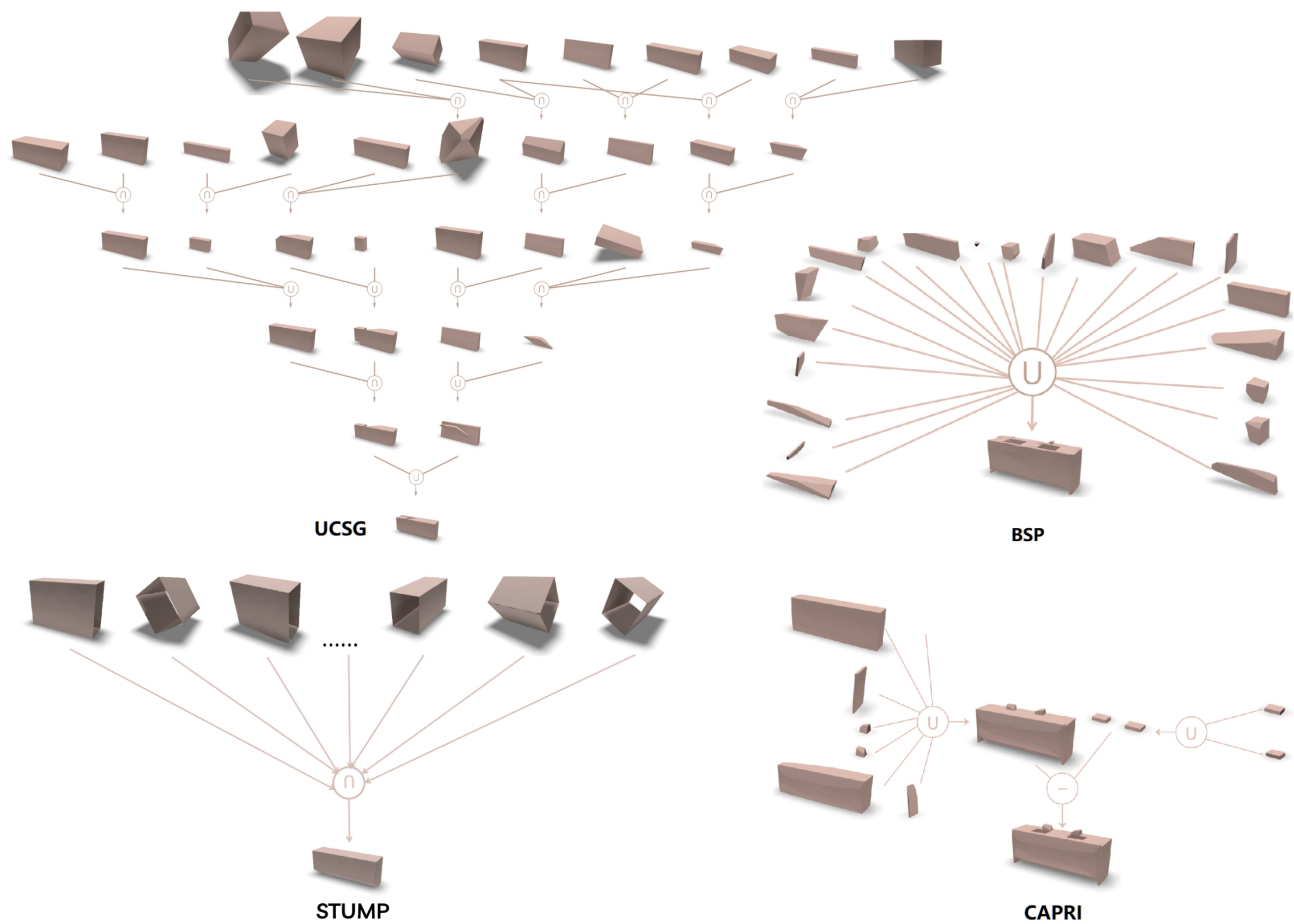

Figure 12. An example of CSG tree visual comparison on ShapeNet.

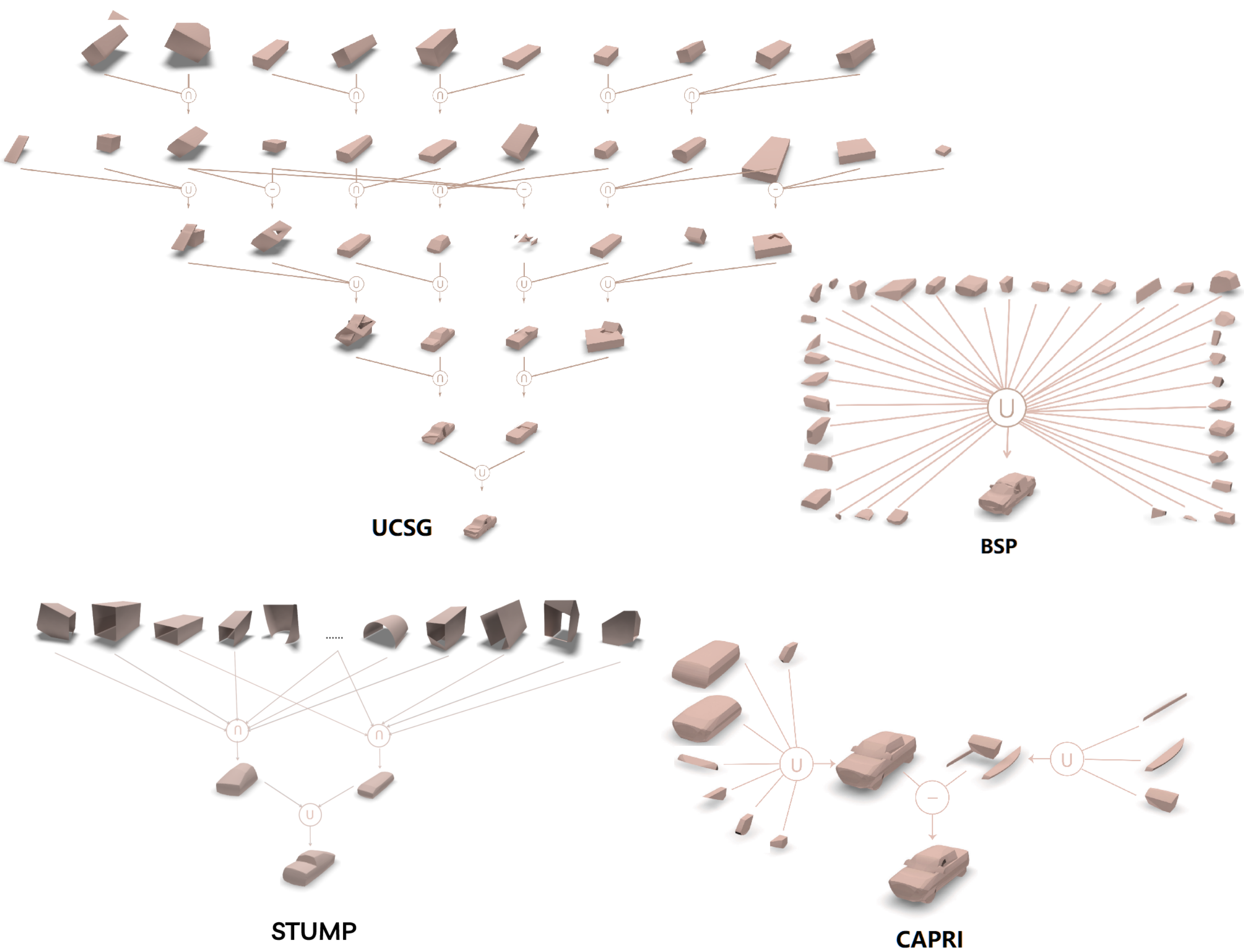

Figure 13. An example of CSG tree visual comparison on ShapeNet.

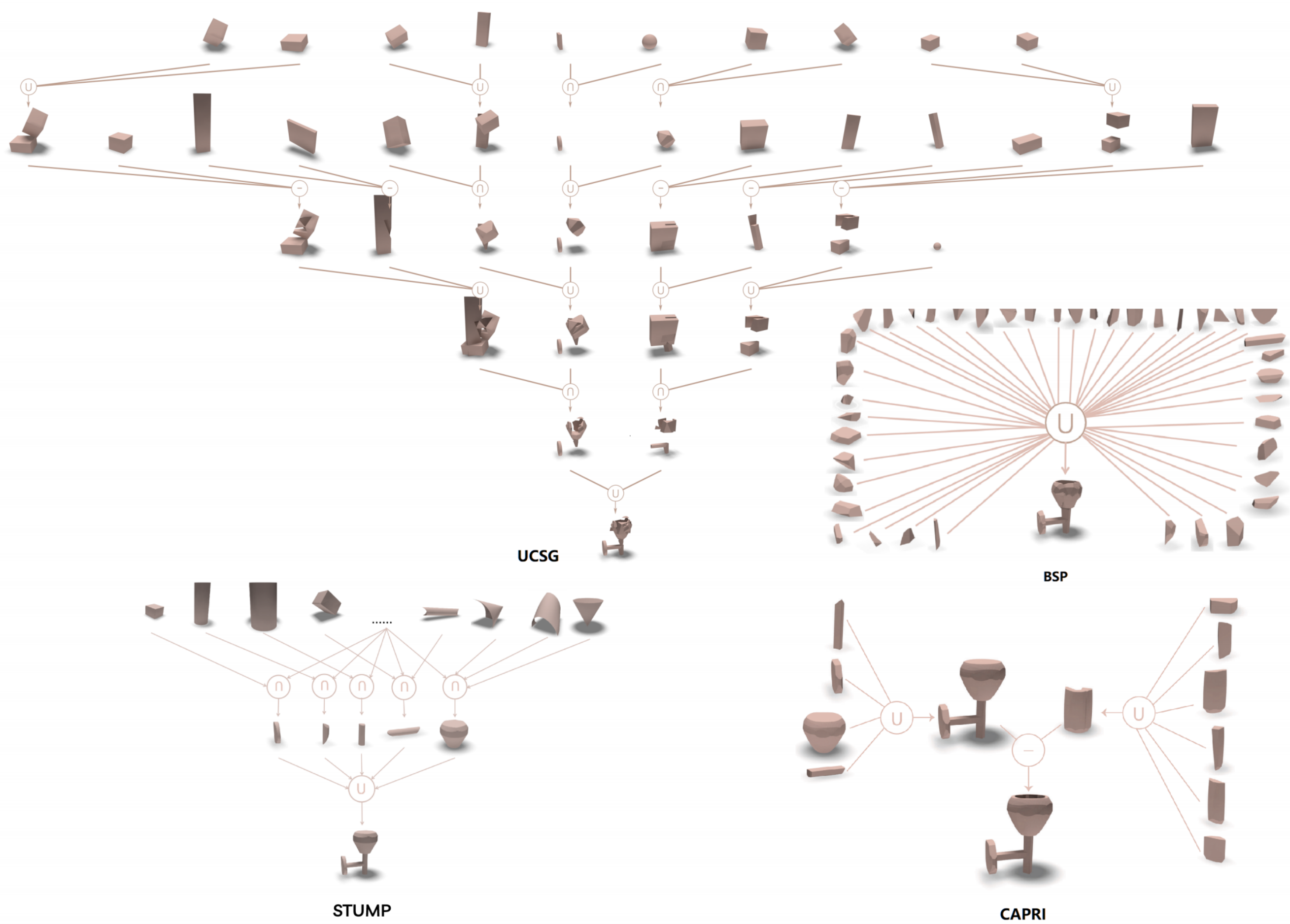

Figure 14. An example of CSG tree visual comparison on ShapeNet.

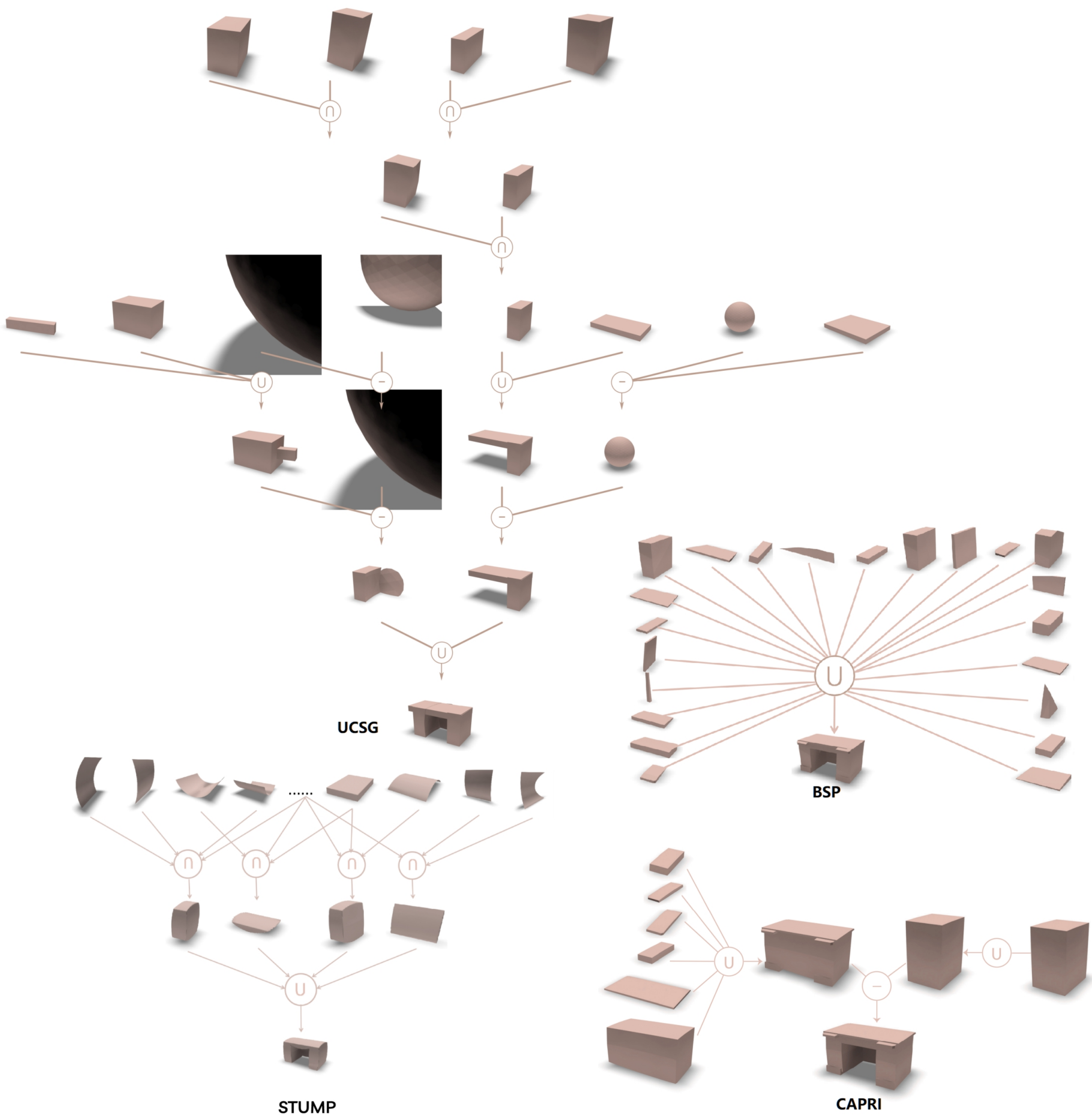

Figure 15. An example of CSG tree visual comparison on ShapeNet.

References

[1] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes vis binary space partitioning. 2020. 1

[2] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. 2019. 1

[3] Jianmin Zheng Daxuan Ren, Jianfei Cai, Haiyong Jiang Jiatong Li, Zhongang Cai, Junzhe Zhang, Liang Pan, Mingyuan Zhang, Haiyu Zhao, and Shuai Yi. CSG-Stump: A learning friendly csg-like representation for interpretable shape parsing. *ICCV*, 2021. 1

[4] Kacper Kania, Maciej Zięba, and Tomasz Kajdanowicz. Ucsg-net–unsupervised discovering of constructive solid geometry tree. *arXiv preprint arXiv:2006.09102*, 2020. 1

[5] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020. 1