ShapeMatcher: Self-Supervised Joint Shape Canonicalization, Segmentation, Retrieval and Deformation

Yan Di[1*], Chenyangguang Zhang[2*], Chaowei Wang[3*], Ruida Zhang[2], Guangyao Zhai[1], Yanyan Li[1], Bowen Fu[2], Xiangyang Ji[2] and Shan Gao†[3]
[1]Technical University of Munich, [2]Tsinghua University, [3]Northwestern Polytechnical University
shangbuhuan13@gmail.com, gaoshan@nwpu.edu.cn

1. Network Architecture

The network structure of ShapeMatcher is illustrated in Fig. 1, containing the structure illustration of the partial and the full branches (Dual-branch structure, (a)), and of the deformation module (b). In (a), taking the full branch as an example, the input point cloud goes through the VNT simple encoder to generate the invariant features. The segmentation network, region centroid extraction network, and feature extraction network all take these features as inputs, each established with a 4-layer MLP. The network of the partial branch shares the exactly same architecture with the full branch. Upon retrieving the most similar CAD model, in (b), the invariant features from the two branches are concatenated and fed into the influence vector extraction network to generate neural cage influence vectors. Based on the influence vectors and the offset of the region centroid, the neural cage is controlled to generate the final deformed mesh.

2. Training and Testing Details

As Sec. 3.5 in the main manuscript illustrates, the joint training procedure for ShapeMatcher contains three stages. First, we train the full branch for gaining *Canonicalization* and *Segmentation* capability. In the first stage, the used loss function is defined as

$$\mathcal{L}^{(1)} = \mathcal{L}_{can} + \mathcal{L}_{seg}. \tag{1}$$

During this stage, the target input of ShapeMatcher S_{tgt} is a full point cloud with a random pose, augmented by S^c_{tgt} lying in the affine-invariant space by random translations $T_{rand} \in [-0.1, 0.1]$ and random rotations $R_{rand} \in [-1, 1]$ on three Eulerian angles respectively. We adopt the supplementary reconstruction branch to predict $\hat{S^c}_{tgt}$ to construct the $\mathcal{L}_{can}$. This stage lasts 200 epochs adopting the Adam optimizer with the learning rate of 0.0006, taking 7 hours on a single NVIDIA 3090.

Second, the partial-full consistency losses $\mathcal{L}_{ccan}$, $\mathcal{L}_{ccen}$ and $\mathcal{L}_{cseg}$ are considered for the partial branch to learn *Canonicalization* and *Segmentation* processes. Specifically, in this stage, the weights of the full branch is frozen. The input of the full branch S_{full} is also generated by S^c_{full} by adding random translations and rotations. Meanwhile, the input of the partial branch $S_{partial}$ is obtained via the truncation mask $U_{f2p} \in \mathbb{R}^N$ by $S_{partial} = S_{full}U_{f2p}$. This truncation operation ensures the consistent partial-full correspondence of S_{full} and $S_{partial}$. In this second stage, the loss function for gradient update of the partial branch is written as

$$\mathcal{L}^{(2)} = \mathcal{L}_{can} + \mathcal{L}_{seg} + 5\mathcal{L}_{ccan} + 2\mathcal{L}_{ccen} + 2\mathcal{L}_{cseg}. \tag{2}$$

The accentuated consistency terms bring two benefits. First, they enforce the concordance of the established affine-invariant space by the partial branch to the space of full branch, enabling the subsequent $\mathbf{R}\&\mathbf{D}$ process taking place in geometrically aligned spaces. Second, they assist the network to better understand semantic-consistent segmentation for the partial branch. They resist the defective influence for segmentation brought by the missing parts in the partial input. During the implementation, this stage lasts another 200 epoches adopting the Adam optimizer with the learning rate of 0.0006, taking 6 hours on a single NVIDIA 3090.

After the two stages, the *Canonicalization* and *Segmentation* processes of both the full and the partial branches are well learned. Thus, in the final stage, we freeze the two modules and train the $\mathbf{R}\&\mathbf{D}$ process. During this stage, the input of the full branch is a randomly selected full source model S_{src}, and the partial input is a randomly selected partial target model S_{tgt}. The utilized $\mathbf{R}\&\mathbf{D}$ losses are

$$\mathcal{L}^{(3)} = \mathcal{L}_{retrieval} + \mathcal{L}_{deform}. \tag{3}$$

During the implementation, this stage lasts 200 epoches

*Authors with equal contributions. † Corresponding author.

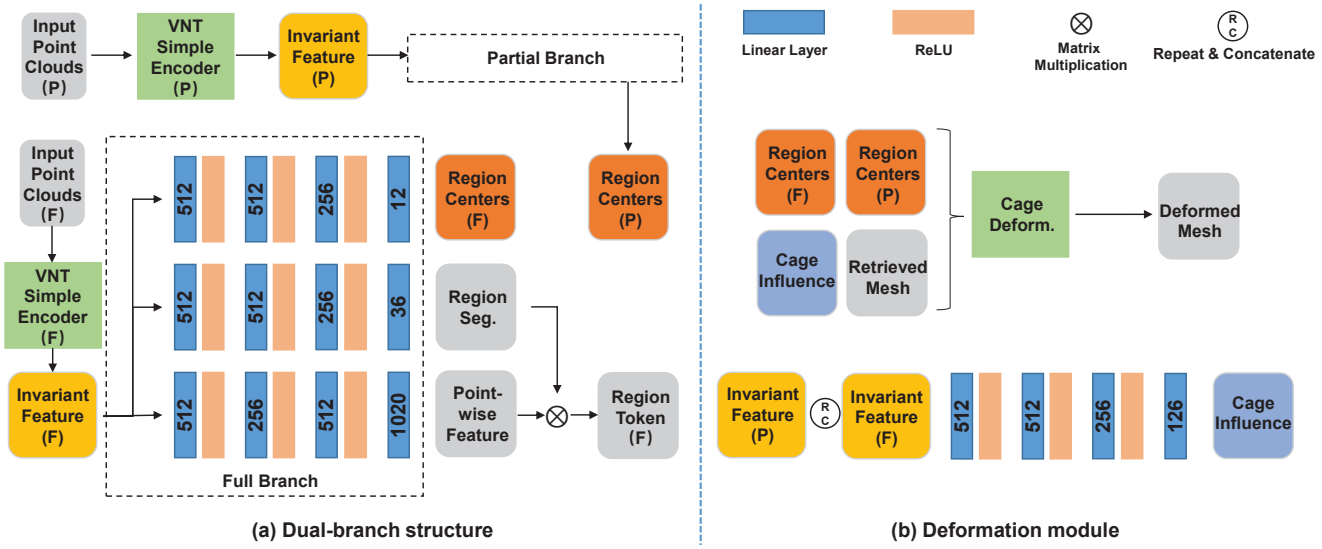

Figure 1. **The network structure of ShapeMatcher.**

Method	Chair	Table	Cabinet	Average
Uy *et al.* [6]	0.466	0.712	**0.146**	0.441
U-RED [3]	1.774	2.875	0.457	1.702
Ours	**0.370**	**0.491**	0.230	**0.364**

Table 1. Inference time analysis.

adopting the Adam optimizer with the learning rate of 0.0006, taking 6 hours on a single NVIDIA 3090.

Finally, in the testing, the partial branch generates the retrieval tokens of the input partial target S_{tgt}, after the *Canonicalization* and *Segmentation* processes. Similarly, the full branch encodes the retrieval tokens of each source model S_{src}. After the region-weighted comparison, the source shape S_r with the smallest distance score is identified as the best retrieval. Consequently, the deformation module deforms S_r to the final output S_{src}^{dfm} which closely matches the target input.

3. Run-time Analysis

Tab. 1 shows the average R&D time (seconds) on a single NVIDIA 3090 on three categories with the database capacity of 494 (Chair), 793 (Table), and 123 (Cabinet). Results exhibit that ShapeMatcher possesses a faster inference speed which attributes to the lightweight network design. Moreover, the training time of ShapeMatcher is 19 hours in total, which is detailed in the three stages in Sec. 2.

4. Visual Illustration of Segmentation

Fig. 2 shows some segmentation results of ShapeMatcher. Our segmentation ensures semantic consistency

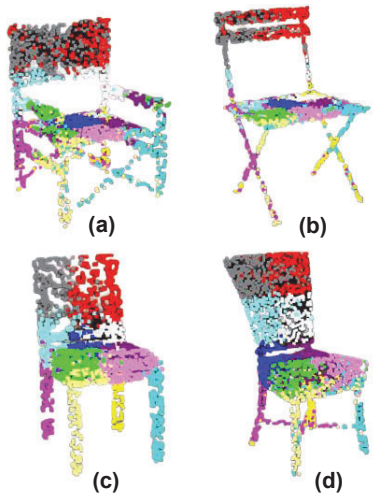

Figure 2. Visual illustration of segmentation results.

across instances, which further enables the accurate region-weighted retrieval and the part center guided deformation.

5. Additional Qualitative Results on Synthetic Cases

We provide additional qualitative results on synthetic PartNet [5] to further exhibit the superiority of ShapeMatcher against baselines Uy *et al.* [6] and U-RED [3]. Fig. 3, Fig. 4 and Fig. 5 present the scenarios when the occlusion rates of the input targets are set to 50%, 25% and 0% (full target inputs). Results show that ShapeMatcher consistently outperforms all baselines with the most resembled retrieved shape and tightly matched deformation.

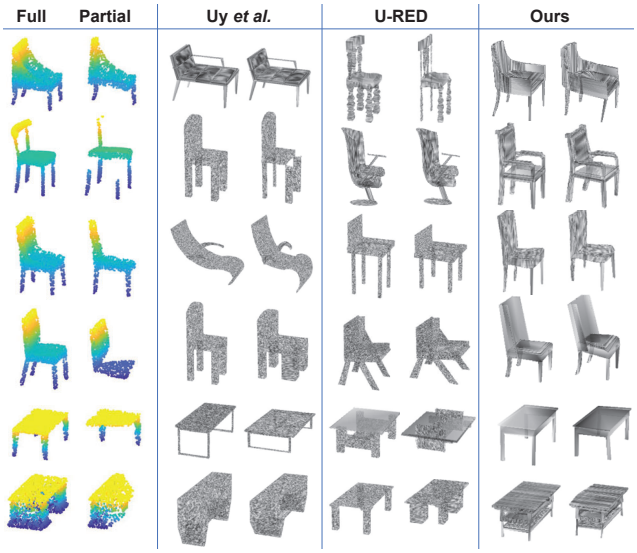

Figure 3. Additional qualitative results on PartNet [5] with the occlusion rate of 50%.

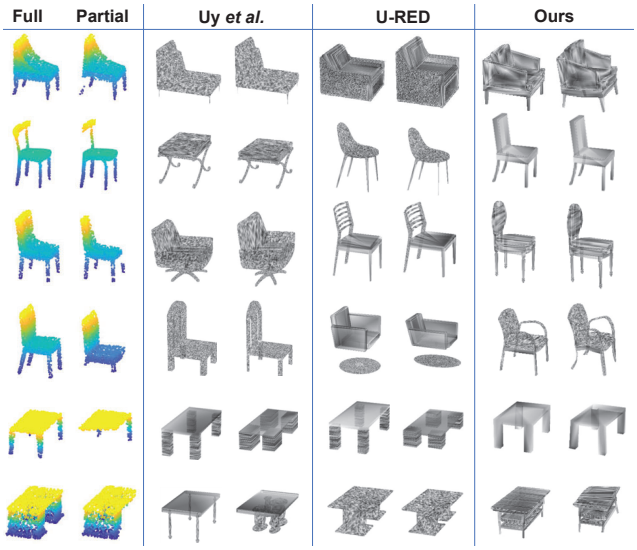

Figure 4. Additional qualitative results on PartNet [5] with the occlusion rate of 25%.

6. Qualitative Results on Real-world Cases

To fully investigate the performance of ShapeMatcher handling real-world cases, we provide qualitative results on Scan2CAD [1] against baselines Uy *et al.* [6] and U-RED [3]. This visualization is made by rendering the R&D shape results to the input image. It is revealed that ShapeMatcher generates the most resembled R&D shapes given occluded and noisy real-world scans under arbitrary poses.

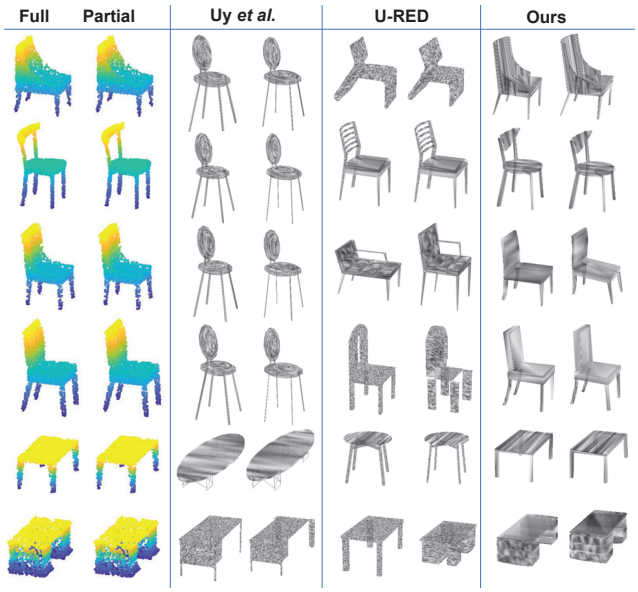

Figure 5. Additional qualitative results on PartNet [5] with full input targets.

Method	Chair	Table	Cabinet	Average
Uy *et al.* + can.	1.430	2.866	1.003	2.111
U-RED + can.	0.722	0.326	1.284	0.570
Ours + can.	**0.201**	**0.152**	**0.514**	**0.202**

Method	$\|S\| = 100$	$\|S\| = 200$	$\|S\| = 400$	$\|S\| = 800$
Uy *et al.*	5.308	4.581	4.700	4.159
Ours	**0.300**	**0.273**	**0.238**	**0.142**

7. Evaluation on Canonical Shapes.

We also provide results on PartNet with **Canonical Shapes** as input. In this way, we mainly compare the retrieval and deformation performance of ShapeMatcher with other competitors. In real applications, the objects can be canonicalized with off-the-shelf pose estimation methods [2, 4, 7], which however, usually lead to low-quality results due to the partial observations of objects.

8. Analysis of Database Diversity

We compare with Uy *et al.* [6] (M.) on the effect of database size on PartNet. We use the same database split with [6]. It is clear that generally larger database $\rightarrow$ better results.

9. Limitations and Failure Cases.

Currently, due to the adopting of the *Retrieval* and *Deformation* procedure, the performance of ShapeMatcher is limited with the access of pre-established database. Moreover, ShapeMatcher gives failure cases under extreme oc-

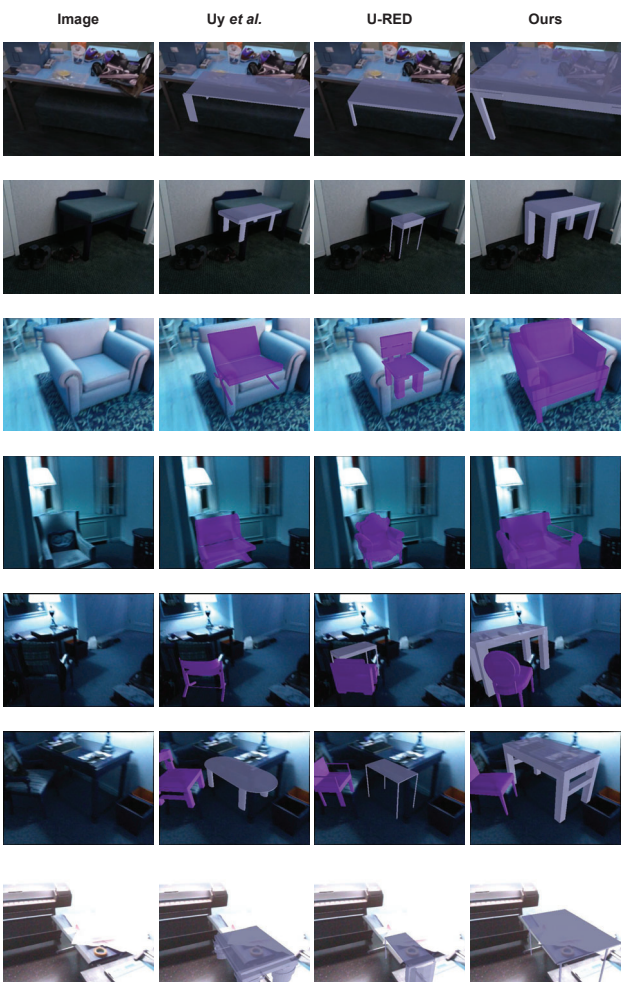

Figure 6. Qualitative results on Scan2CAD [1].

Figure 7. Additional qualitative results on Scan2CAD [1].

clusion conditions, as Fig. 8 shows. When the important structures of the input target shape are heavily occluded, it is hard for our method to complement it, so as the baseline methods Uy *et al.* [6] and U-RED [3].

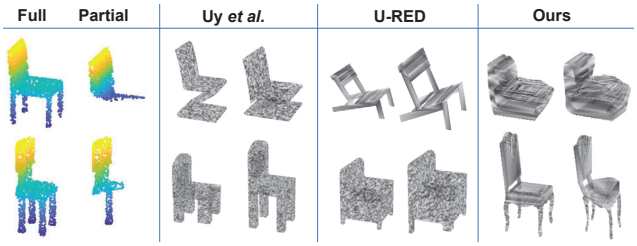

Figure 8. Failure cases on PartNet [5] with the occlusion rate of 50%.

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 2614–2623, 2019. 3, 4

[2] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. So-pose: Exploiting self-occlusion for direct 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12396–12405, 2021. 3

[3] Yan Di, Chenyangguang Zhang, Ruida Zhang, Fabian Manhardt, Yongzhi Su, Jason Rambach, Didier Stricker, Xiangyang Ji, and Federico Tombari. U-red: Unsupervised 3d shape retrieval and deformation for partial point clouds. *arXiv preprint arXiv:2308.06383*, 2023. 2, 3, 4

[4] Yan Di, Ruida Zhang, Zhiqiang Lou, Fabian Manhardt, Xiangyang Ji, Nassir Navab, and Federico Tombari. Gpv-pose: Category-level object pose estimation via geometry-guided point-wise voting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6781–6791, 2022. 3

[5] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tri-

pathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 2, 3, 4

[6] Mikaela Angelina Uy, Vladimir G Kim, Minhyuk Sung, Noam Aigerman, Siddhartha Chaudhuri, and Leonidas J Guibas. Joint learning of 3d shape retrieval and deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11722, 2021. 2, 3, 4

[7] Ruida Zhang, Yan Di, Zhiqiang Lou, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Rbp-pose: Residual bounding box projection for category-level pose estimation. In *European Conference on Computer Vision*, pages 655–672. Springer, 2022. 3