Supplementary material of ES6D

Ningkai Mo [1*] Wanshui Gan [1,2*] Naoto Yokoya [2,3] Shifeng Chen [1†]

[1] ShenZhen Key Lab of Computer Vision and Pattern Recognition,
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences,
[2] The University of Tokyo, [3]RIKEN
[nk.mo19941001, wanshuigan]@gmail.com, yokoya@k.u-tokyo.ac.jp, shifeng.chen@siat.ac.cn

The details of XYZNet As shown in Figure 1, the RGB patch and the XYZ map are concatenated as the input of XYZNet. $conv(c, k, s, p)$ means the 2D convolution with output channels c, kernel size k, stride s and padding p. $resblock(c, s)$ is the standard residual block of ResNet with output channels c and stride s. The output of $adaptive\_avgpool(h, w)$ or $adaptive\_maxpool(h, w)$ is of size $h * w$ for any input size and the number of output features is equal to the number of input planes.

The rules of grouping All the categories are illustrated in the main paper (section 3.4) with the toy model. The detailed grouping principle is as follows. All the symbols are defined as the same as the main paper.

- For all categories, the object centroid is the group g_0:

$$g_0 = \{(0, 0, 0)\}. \tag{1}$$

- For category 1, there are two continuous symmetric axis-angles, we can select one as group g_1:

$$g_1 = \{\boldsymbol{e}(\boldsymbol{a}_1)\}, \quad \boldsymbol{a}_1 \in AC_O. \tag{2}$$

- For category 2, there are two discrete symmetric axis-angles, we can select one as group g_1, but one axis is not enough to provide constraints for pose estimation, so we need to generate axes of number $|\boldsymbol{a_1}|$ as an auxiliary group g_2:

$$\begin{aligned} &g_1 = \{\boldsymbol{e}(\boldsymbol{a}_1)\}, \quad \boldsymbol{a}_1 \in A_O; \\ &g_2 = \{\boldsymbol{e}_{\text{base}}\} \cup \{R(\boldsymbol{a}_1)^n \, \boldsymbol{e}_{\text{base}}\}_{n=1}^{|\boldsymbol{a_1}|-1}, \end{aligned} \tag{3}$$

where $\boldsymbol{e}_{\text{base}}$ is a generated axis and is perpendicular to $\boldsymbol{e}(\boldsymbol{a}_1)$. The demonstration of grouping is shown in first row in Figure 9.

- For category 3, the object centroid or the g_0 is the only group.

- For category 4, there are two continuous symmetric axis-angles, which make up the group g_1:

$$g_1 = \{\boldsymbol{e}(\boldsymbol{a}_0), \boldsymbol{e}(\boldsymbol{a}_1)\}, \quad \boldsymbol{a}_0, \boldsymbol{a}_1 \in AC_O \wedge \boldsymbol{a}_0 \neq \boldsymbol{a}_1. \tag{4}$$

- For category 5, there is no continuous symmetric axis-angle. We need to group all discrete symmetric axis-angles. If the axis A could overlap with axis B after the specific angle around a symmetric axis, we regard A and B lie in the same group:

$$\begin{aligned} g_i =& \{\boldsymbol{e}(\widehat{\boldsymbol{a}})^i_{\text{base}}\} \cup \\ & \{\boldsymbol{e}(\boldsymbol{a}) | \boldsymbol{e}(\boldsymbol{a}) = R(\dot{\boldsymbol{a}})^n \boldsymbol{e}(\widehat{\boldsymbol{a}})^i_{\text{base}} \\ & \exists \dot{\boldsymbol{a}} \in A_o \wedge \exists n \in [1, \cdots, |\dot{\boldsymbol{a}}| - 1]\}, \end{aligned} \tag{5}$$

where $\boldsymbol{e}(\widehat{\boldsymbol{a}})^i_{\text{base}}$ is selected as the base of group g_i from the ungrouped axes of A_O. The demonstration of grouping is in the second row in Figure 9.

- For the asymmetry object:

$$\begin{aligned} &g_0 = \{(0, 0, 0)\}, \quad g_1 = \{(1, 0, 0)\}, \\ &g_2 = \{(0, 1, 0)\}, \quad g_3 = \{(0, 0, 1)\}. \end{aligned} \tag{6}$$

The details of grouped primitives in YCB-Video dataset [5] As shown in Figure 2, the first plot is the raw GP of objects in YCB-Video. For better performance, the GP is scaled by the object's shape and we trim the GP of the asymmetry object by the principal component of the object's shape, as shown in the second plot of Figure 2. We also check the validation of these processed GP by the numerical and visualization method, the results are illustrated in Figure 3.

More instances of category 2 and 5 in symmetry As stated in our main paper, symmetry objects can be divided

[*] The first two authors contributed equally and should be regarded as co-first authors.

[†] Corresponding author.

into five categories. There is only one instance per category except for category 2 and 5, so we display more instances of category 2 in Figure 4 and more instances of category 5 in Figure 5. The first instance in Figure 5 is a regular hexahedron, which has $6+12+8$ symmetry axes, and there are $2*6+1*12+1*8+1$ minima in the A(M)GPD landscape.

Complete experiment results We give the complete experiment results of YCB-Video dataset in Table 2. Besides, we also report the AUC of A(M)GPD in Table 1 for a more comprehensive evaluation. The ground truth masks are used as input in all methods in Table 1. As Table 1 illustrated, the proposed ES6D shows that training with the A(M)GPD loss can bring excellent accuracy on both symmetry and asymmetry objects.

Observation on two bad cases In the experiment with PoseCNN segment mask and ground truth mask, we can see that there is a large gap between the symmetric objects, *051_large_clamp* and *052_extra_large_clamp*, as listed in Table 2. We further investigate the reason for this phenomenon. We visualize the mask of *051_large_clamp* and *052_extra_large_clamp* in Figure 6. We can see that, semantic segmentation network failed in distinguishing two objects with the same appearance, this would raise the category level mistake and further lead to the low accuracy on pose estimation. Furthermore, one mask in semantic segmentation could include both *051_large_clamp* and *052_extra_large_clamp* part, since we crop the RGB image based on the mask result, the wrong mask prediction would lead to the extra cropped region. This would lead to a larger domain gap between the training dataset and the testing dataset. We conclude the above two reasons resulting in these two bad cases. On the other hand, in the experiment with the ground truth mask, ES6D achieves the best performance compared with PVN3D [2].

More visualization result on T-LESS dataset [3] We provide more visualization results in Figure 7 and Figure 8 for comparison. The green, red, and blue lines represent the ground truth pose, the result from A(M)GPD loss, and the result from ADD(S) loss, respectively. We can observe that the result from A(M)GPD loss is more accurate than the result from ADD(S) loss. The result from ADD(S) loss sometimes could be totally reversed due to the local minimal problem. The proposed A(M)GPD loss could effectively address this problem.

Comparison in YCB-Video dataset [5] with video demonstration For a visualization comparison with [2] and [4], we make the demonstration video as attached for reference. We can see that our result is more accurate and stable compared with [2] and [4].

References

[1] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 3

[2] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11632–11641, 2020. 2, 3

[3] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888. IEEE, 2017. 2

[4] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019. 2, 3

[5] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *Robotics: Science and Systems (RSS)*, 2018. 1, 2, 8

	DenseFusion (iterative)	PVN3D	XYZNet with ADD(S) loss	ES6D
002_master_chef_can	82.9	86.1	85.1	84.8
003_cracker_box	92.5	95.0	96.1	96.2
004_sugar_box	97.6	96.3	98.2	98.2
005_tomato_soup_can	90.7	90.9	93.5	94.0
006_mustard_bottle	96.9	96.6	98.4	98.6
007_tuna_fish_can	75.5	87.0	92.6	91.9
008_pudding_box	96.8	95.4	97.1	97.4
009_gelatin_box	98.2	95.6	98.5	98.7
010_potted_meat_can	86.8	88.3	89.1	88.2
011_banana	77.2	91.9	94.6	96.3
019_pitcher_base	97.4	96.7	97.9	98.0
021_bleach_cleanser	95.3	93.6	94.0	95.6
024_bowl	52.8	68.7	42.9	94.2
025_mug	94.1	94.6	92.9	93.0
035_power_drill	95.9	95.0	97.0	97.2
036_wood_block	94.0	80.2	91.8	92.7
037_scissors	81.9	92.6	80.4	65.2
040_large_marker	95.5	92.6	94.1	94.1
051_large_clamp	38.7	79.8	90.9	92.9
052_extra_large_clamp	36.8	59.1	90.4	92.6
061_foam_brick	63.1	91.7	94.6	92.7
ALL	83.7	89.1	91.9	**93.5**

Table 1. The comparison of 6D Pose (A(M)GPD)) on YCB-Video Dataset. Object names with bold typeface are symmetric.

			With PoseCNN segment mask						With GT segment mask			
	FFB6D [1]		DenseFusion [4] (per-pixel)		DenseFusion [4] (iterative)		Ours		PVN3D [2] (post process)		Ours	
	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)
002_master_chef_can	96.3	80.6	95.3	70.7	96.4	73.2	96.5	73.0	96.2	79.2	97.0	74.8
003_cracker_box	96.3	94.6	92.5	86.9	95.8	94.1	95.3	94.0	95.9	94.7	96.7	95.8
004_sugar_box	97.6	96.6	95.1	90.8	97.6	96.5	97.9	97.3	97.4	96.4	98.3	98.2
005_tomato_soup_can	95.6	89.6	93.8	84.7	94.5	85.5	97.3	90.4	96.6	88.5	97.3	91.9
006_mustard_bottle	97.8	97.0	95.8	90.9	97.3	94.7	98.2	97.9	97.4	96.3	98.5	98.4
007_tuna_fish_can	96.8	88.9	95.7	79.6	97.1	81.9	97.4	93.7	96.2	88.6	97.4	93.5
008_pudding_box	97.1	94.6	94.3	89.3	96.0	93.3	96.5	93.4	96.7	95.2	97.8	97.3
009_gelatin_box	98.1	96.9	97.2	95.8	98.0	96.7	97.7	96.5	97.8	96.2	98.9	98.9
010_potted_meat_can	94.7	88.1	89.3	79.6	90.7	83.6	92.5	84.6	93.6	88.3	93.4	86.8
011_banana	97.2	94.9	90.0	76.7	96.2	83.3	97.9	95.8	96.7	93.6	97.9	96.8
019_pitcher_base	97.6	96.9	93.6	87.1	97.5	96.9	97.8	97.7	97.1	96.5	97.9	97.8
021_bleach_cleanser	96.8	94.8	94.4	87.5	95.9	89.9	96.3	92.8	96.1	93.1	97.0	94.8
024_bowl	96.3	96.3	86.0	86.0	89.5	89.5	96.4	96.4	88.7	88.7	96.8	96.8
025_mug	97.3	94.2	95.3	83.8	96.7	88.9	97.3	95.0	97.5	95.5	97.5	94.5
035_power_drill	97.2	95.9	92.1	83.7	96.0	92.7	97.2	96.3	96.8	95.3	97.8	97.4
036_wood_block	92.6	92.6	89.5	89.5	92.8	92.8	94.4	94.4	91.5	91.5	96.0	96.0
037_scissors	97.7	95.7	90.1	77.4	92.0	77.9	87.1	61.5	96.9	93.5	89.6	71.5
040_large_marker	96.6	89.1	95.1	89.1	97.6	93.0	97.8	90.6	96.7	91.8	98.3	90.7
051_large_clamp	96.8	96.8	71.5	71.5	72.5	72.5	61.0	61.0	94.4	94.4	97.5	97.5
052_extra_large_clamp	96.0	96.0	70.2	70.2	69.9	69.9	59.6	59.6	91.1	91.1	96.8	96.8
061_foam_brick	97.3	97.3	92.2	92.2	92.0	92.0	96.6	96.6	96.8	96.8	96.9	96.9
ALL	**96.6**	**92.7**	91.2	82.9	93.2	86.1	**93.6**	**89.0**	95.7	91.9	**97.1**	**93.2**

Table 2. The comparison of 6D Pose (ADD-S, ADD(S)) on YCB-Video Dataset. Object names with bold typeface are symmetric, and the number with bold typeface means the best result.

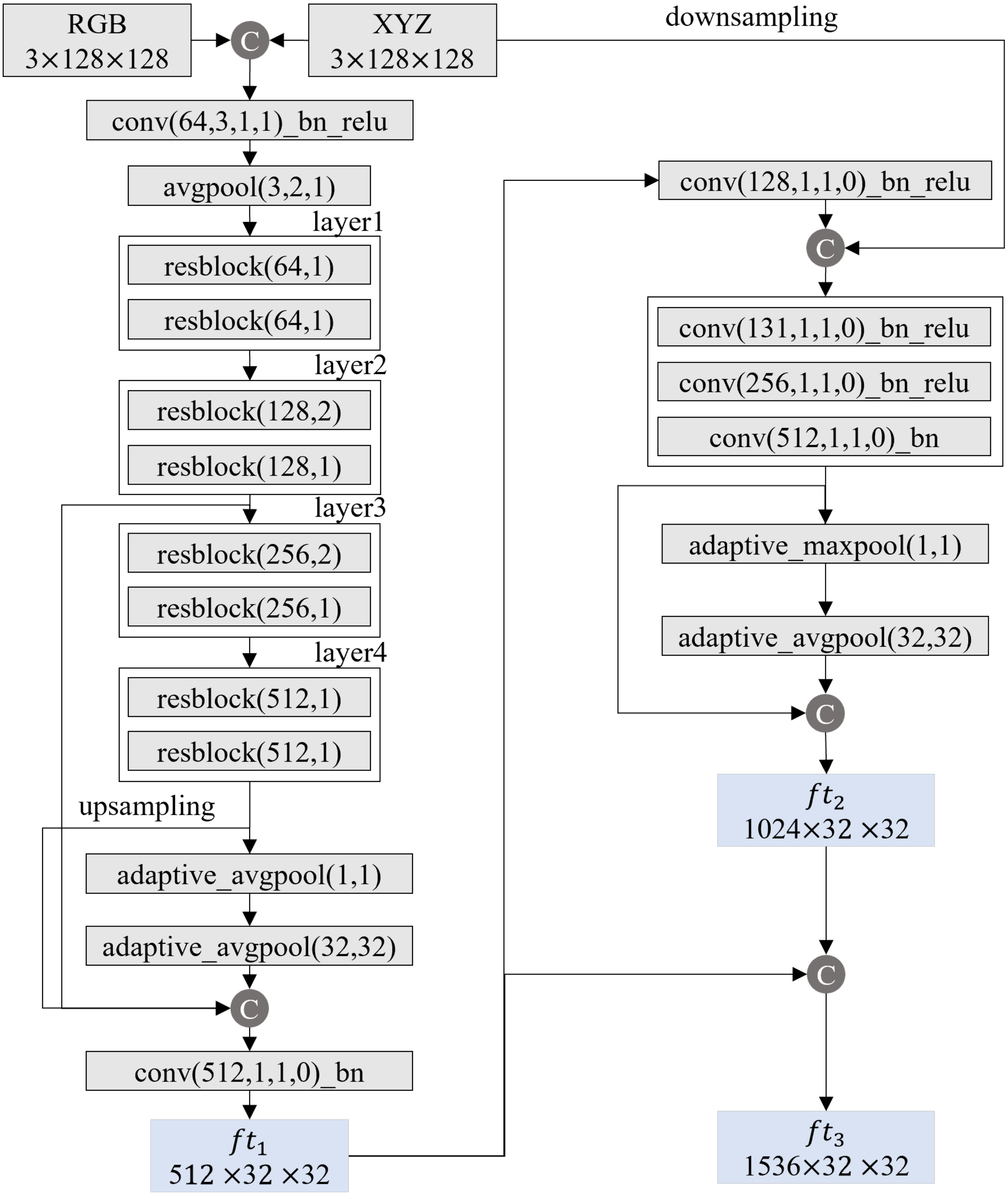


Figure 1. **The details of XYZNet**.

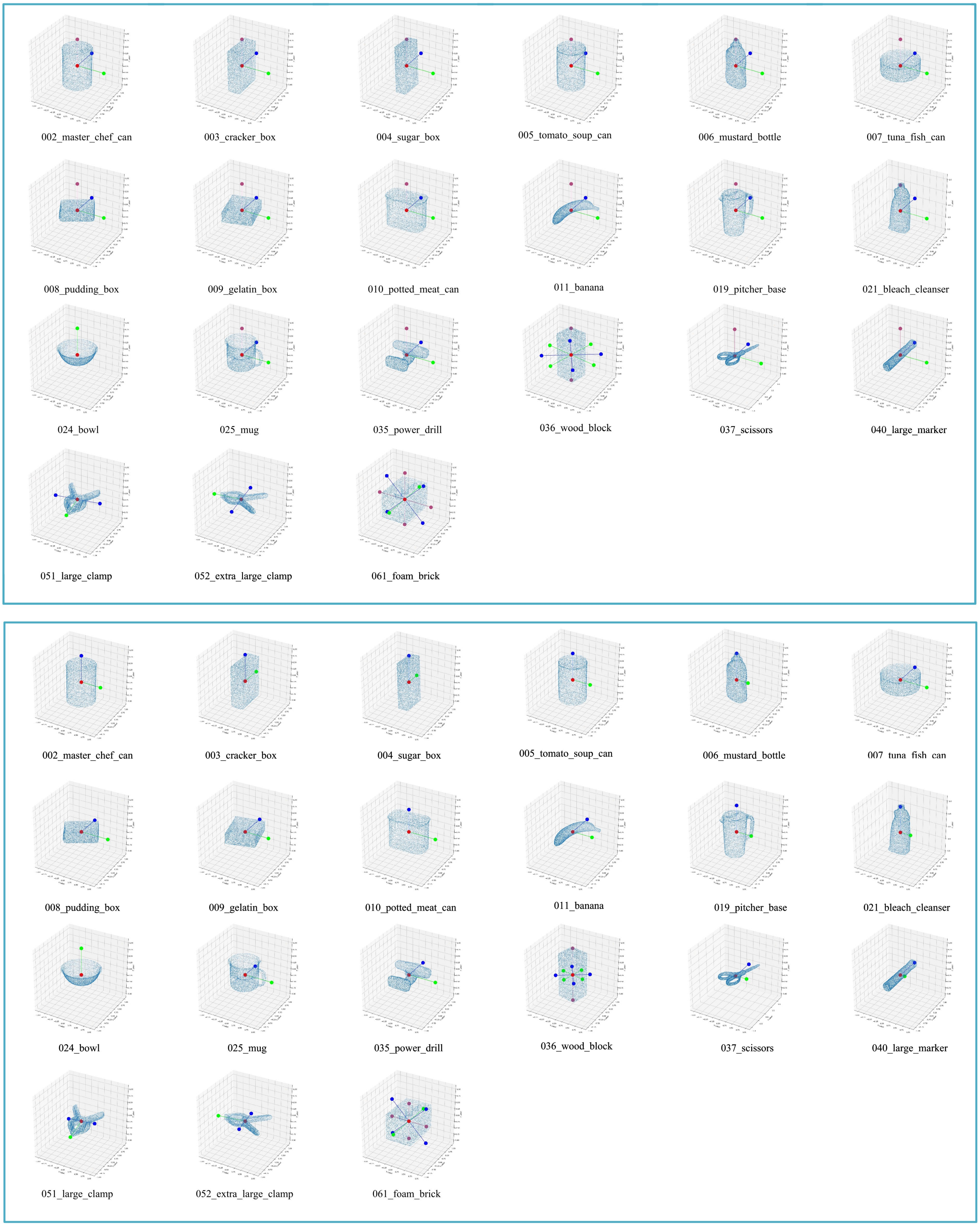


Figure 2. **The details of grouped primitives in YCB-Video dataset**. The first plot is the raw GP of objects, and the second plot is the processed GP of objects.

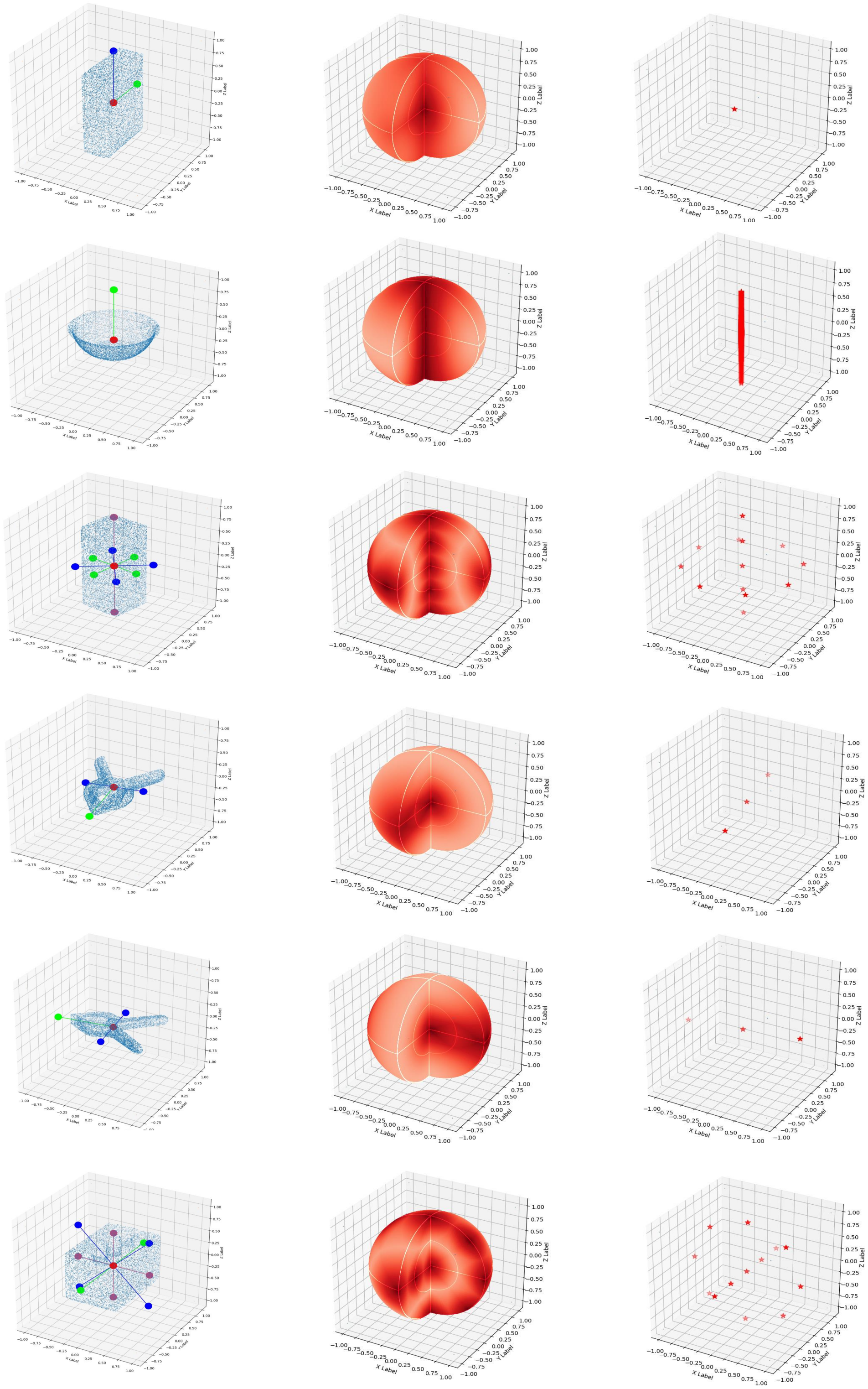


Figure 3. **The validation of processed grouped primitives in YCB-Video dataset**. For each object, the first column presents the grouped primitives. The second shows the A(M)GPD landscape in the rotation space, where the darker color represents the smaller value of A(M)GPD. The third column reveals the minima in each landscape. Best viewed in color.

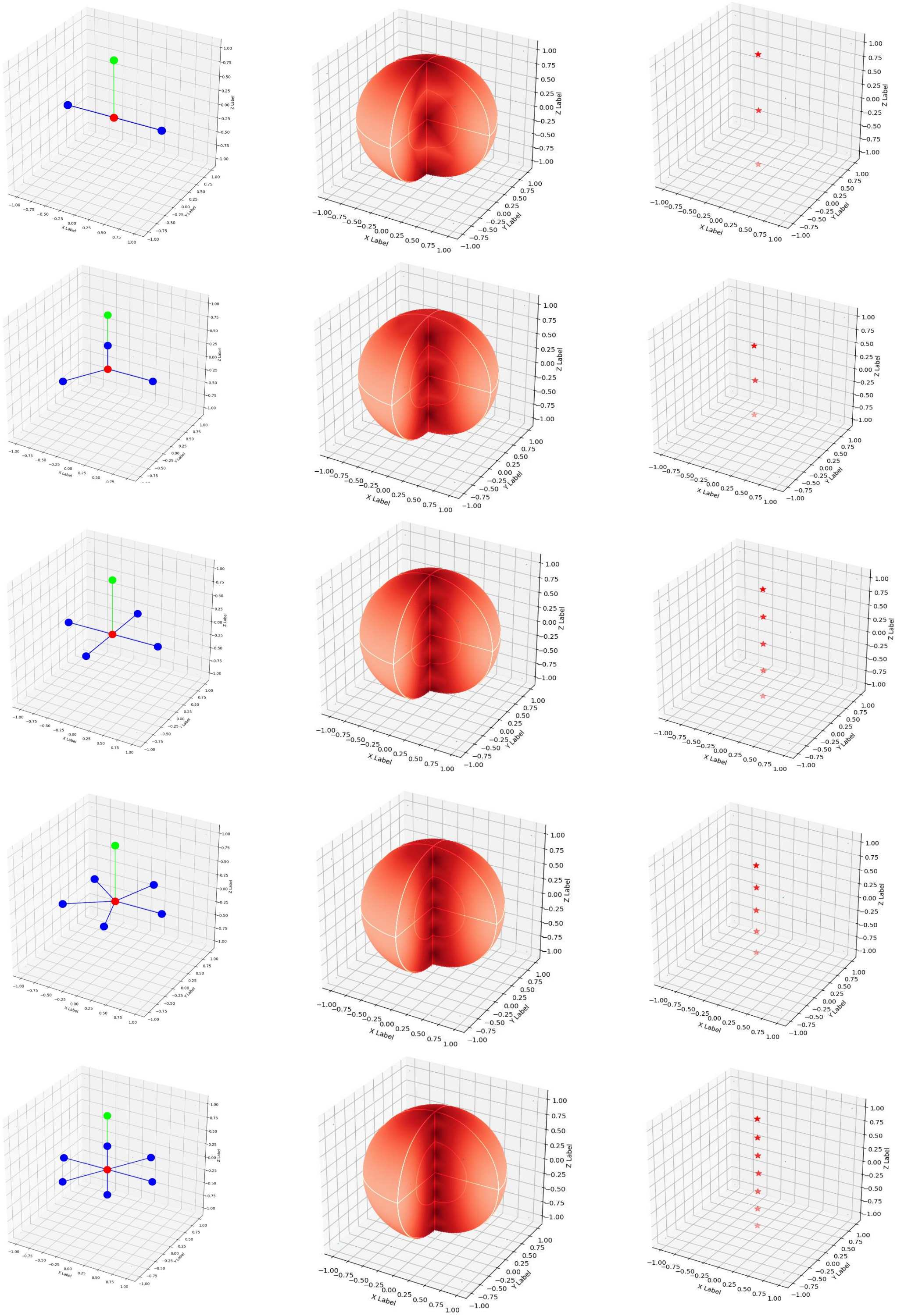


Figure 4. **More instances of category 2**. For each example, the first column presents the grouped primitives. The second shows the A(M)GPD landscape in the rotation space, where the darker color represents the smaller value of A(M)GPD. The third column reveals the minima in each landscape. Best viewed in color.

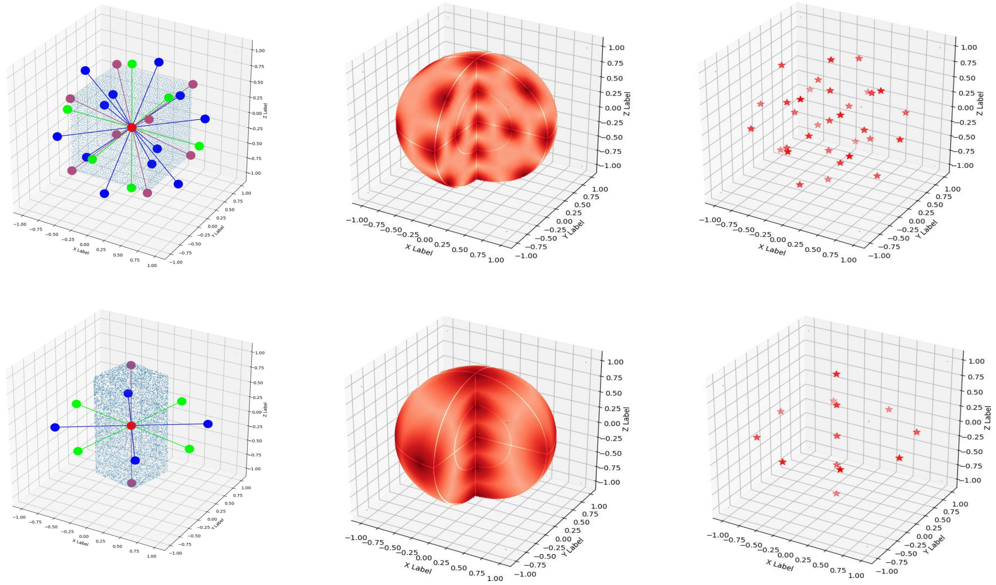


Figure 5. **More instances of category 5**. For each example, the first column presents the grouped primitives. The second shows the A(M)GPD landscape in the rotation space, where the darker color represents the smaller value of A(M)GPD. The third column reveals the minima in each landscape. Best viewed in color.

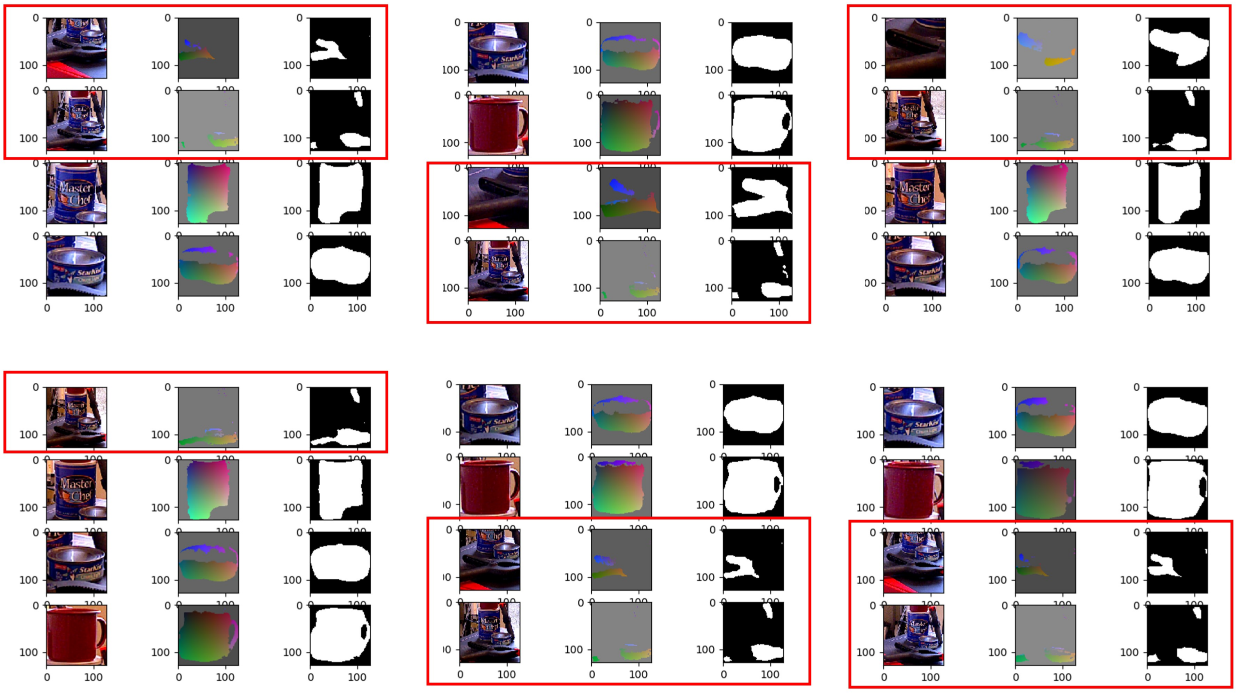


Figure 6. **Visualization for *051_large_clamp* and *052_extra_large_clamp* on the YCB-Video testing dataset**. The *051_large_clamp* and *052_extra_large_clamp* are marked with the red rectangle. The mask result comes from [5].

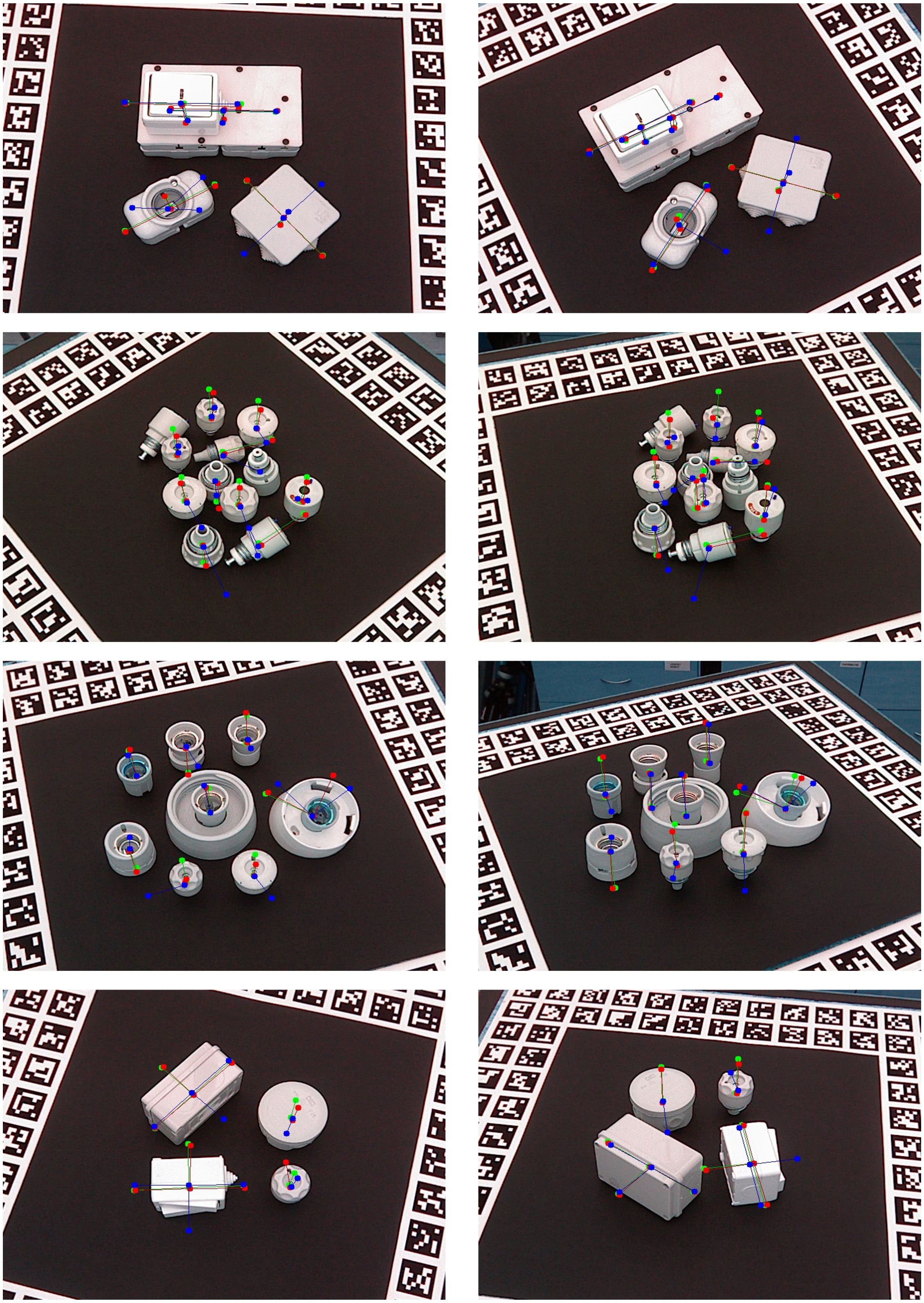


Figure 7. **Visualization on the T-LESS dataset with different training loss**. The green, red, and blue lines represent the ground truth pose, the result from A(M)GPD loss, and the result from ADD(S) loss, respectively.

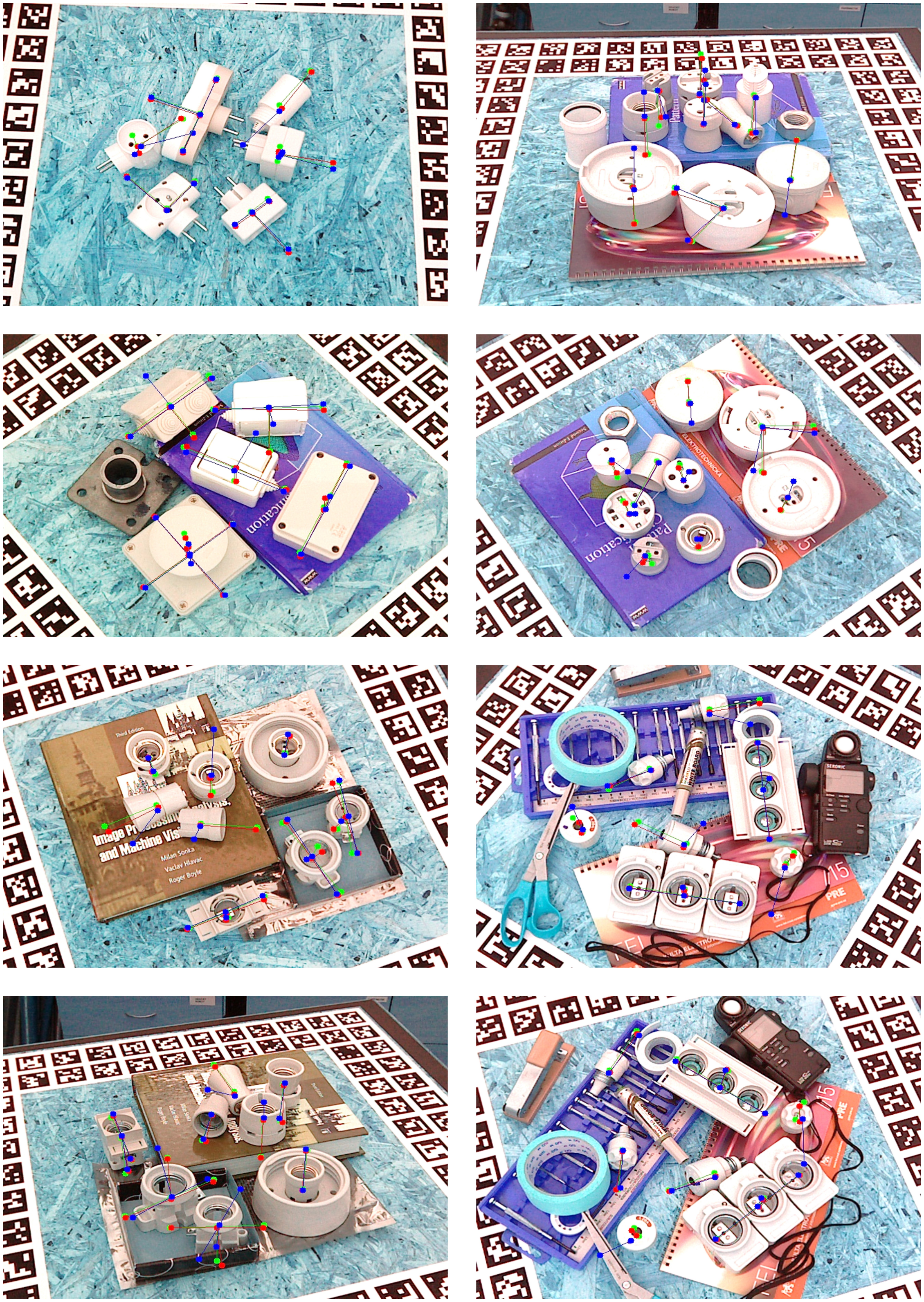
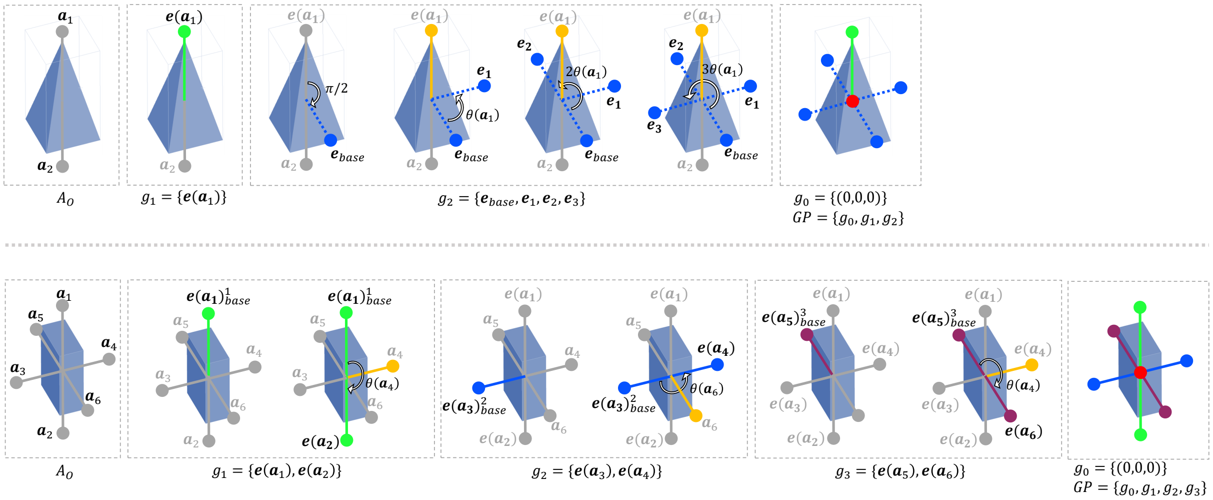


Figure 8. **Visualization on the T-LESS dataset with different training loss**. The green, red, and blue lines represent the ground truth pose, the result from A(M)GPD loss, and the result from ADD(S) loss, respectively.



A_O $g_1 = \{e(a_1)\}$ $g_2 = \{e_{base}, e_1, e_2, e_3\}$ $g_0 = \{(0,0,0)\}$ $GP = \{g_0, g_1, g_2\}$

A_O $g_1 = \{e(a_1), e(a_2)\}$ $g_2 = \{e(a_3), e(a_4)\}$ $g_3 = \{e(a_5), e(a_6)\}$ $g_0 = \{(0,0,0)\}$ $GP = \{g_0, g_1, g_2, g_3\}$

Figure 9. **Demonstration of grouping**. The first row shows the grouping operation for category 2, and the second row is for category 5.