合肥工業大學 HEFEI UNIVERSITY OF TECHNOLOGY 计算机与信息学院 SCHOOL OF COMPUTER SCIENCE AND INFORMATION ENGINEERING 人工智能学院 SCHOOL OF ARTIFICIAL INTELLIGENCE

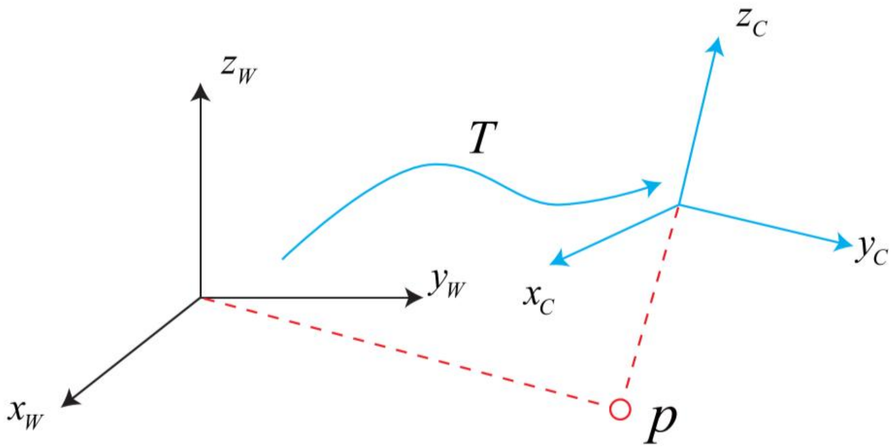
Generalizable and Actionable Parts Pose Estimation with Symmetry Annotation-Free Learning Strategy

ICML 2026

Wenxiao Chen* [1] , Xueyu Yuan* [1] , Liu Liu [1] , Di Wu [2] , Dan Guo [1]
[1]Hefei University of Technology [2]University of Science and Technology of China

CONTENT

6D Pose : thetransformation from object coordinate to world coordinate .



- **6D : 3D Rotation + 3D Translation**

$$a' = Ra + t.$$

- **SO(3) is the space of all valid 3D rotations :**

$$SO(n) = \{\boldsymbol{R} \in \mathbb{R}^{n\times n} | \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}, \det(\boldsymbol{R}) = 1\}.$$

- **Homogeneous coordinates :**

$$\begin{bmatrix} \boldsymbol{a}' \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ 1 \end{bmatrix} \triangleq \boldsymbol{T} \begin{bmatrix} \boldsymbol{a} \\ 1 \end{bmatrix}.$$

express rotation and translation in a unified linear matrix form.

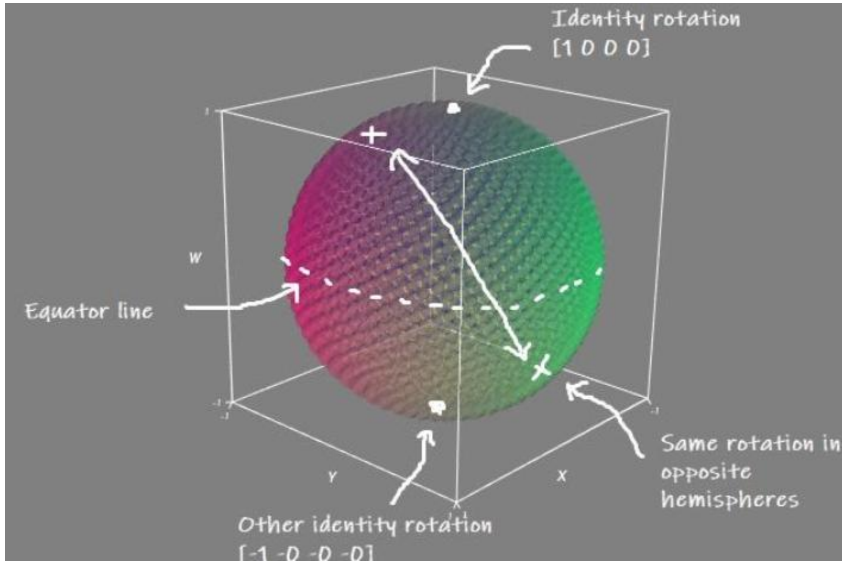
- **SE(3) is the space of all 3D rigid transformations :**

$$SE(3) = \left\{ \boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} | \boldsymbol{R} \in SO(3), \boldsymbol{t} \in \mathbb{R}^3 \right\}.$$

In 6D pose estimation, a pose can be represented by a unit quaternion q : $\boldsymbol{q} = q_0 + q_1 i + q_2 j + q_3 k,$

$$\boldsymbol{q} = [s, \boldsymbol{v}], \quad s = q_0 \in \mathbb{R}, \boldsymbol{v} = [q_1, q_2, q_3]^T \in \mathbb{R}^3,$$

Unit quaternions lie on a 4D unit hypersphere, where antipodal points q and -q represent the same 3D rotation.

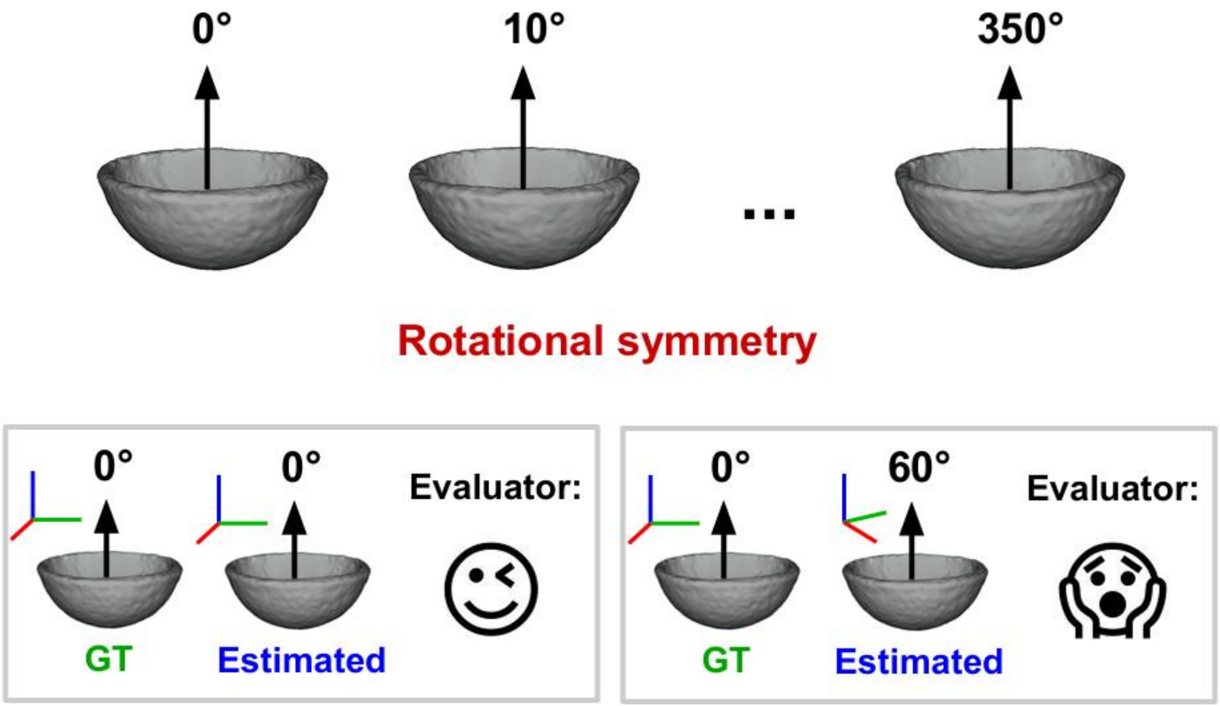


- **Advantage:**

1. Quaternions are easy to compose through quaternion multiplication.
2. Quaternions are easy to apply for rotating 3D points or vectors.
3. Quaternions support smooth interpolation using Spherical Linear Interpolation.
4. Quaternions avoid the gimbal lock problem, but they need normalization and are less intuitive to manipulate directly.

Therefore, in our method, we use quaternions to represent 3D rotation. Compared with rotation matrices, quaternions provide a more compact and singularity-free representation, requiring only a single four-dimensional vector to describe rotation. This makes them more suitable for candidate generation, refinement, and aggregation in our 6D pose estimation framework.

For symmetric objects or parts, one visual observation may correspond to multiple valid 6D poses.



Due to object symmetry, multiple poses may be visually indistinguishable.

- **Main problem:**

1. Multi-solution ambiguity

A single ground-truth pose is not enough, because multiple equivalent poses can describe the same object state.

2. Rotation ambiguity

Rotating an object around its symmetry axis may produce almost the same appearance.

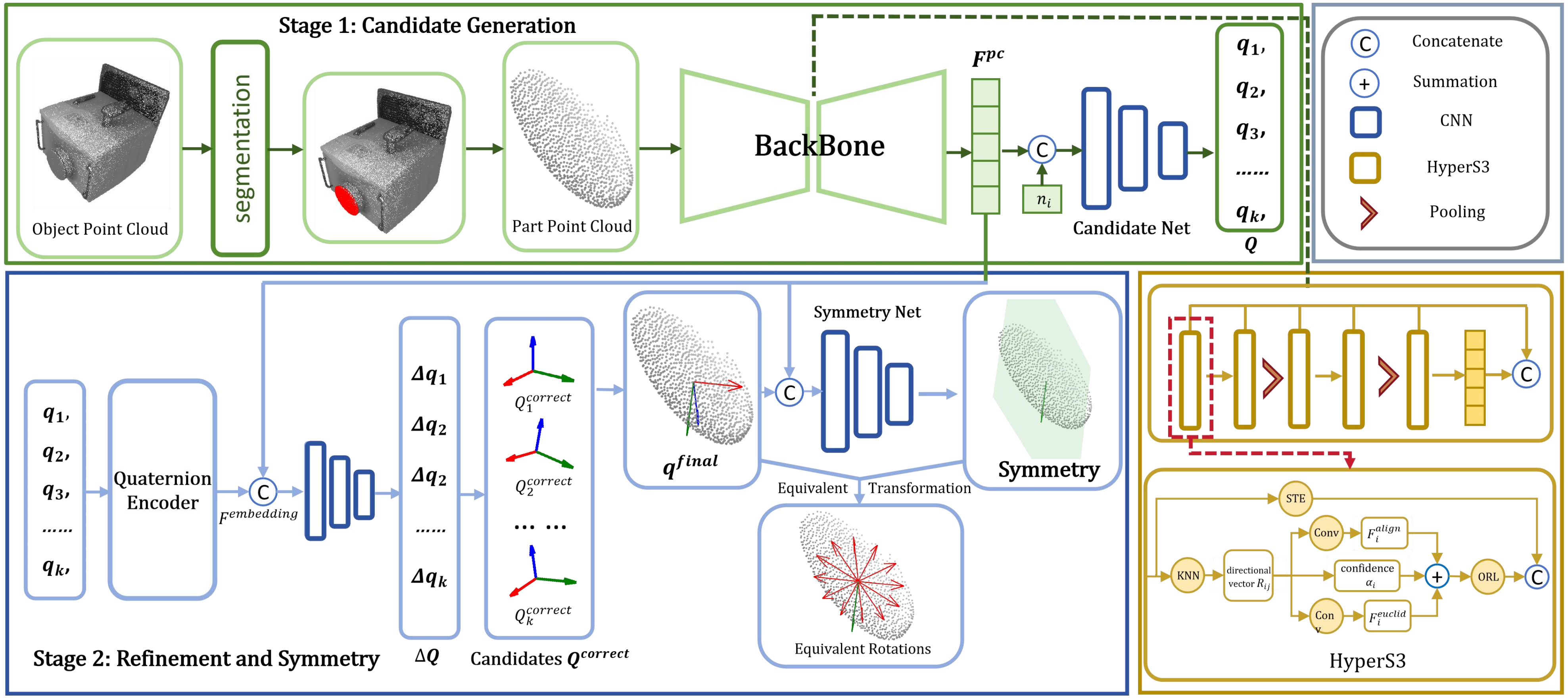
3. Mirror ambiguity

Reflection across a symmetry plane may also lead to indistinguishable visual observations.

4. Annotation dependency

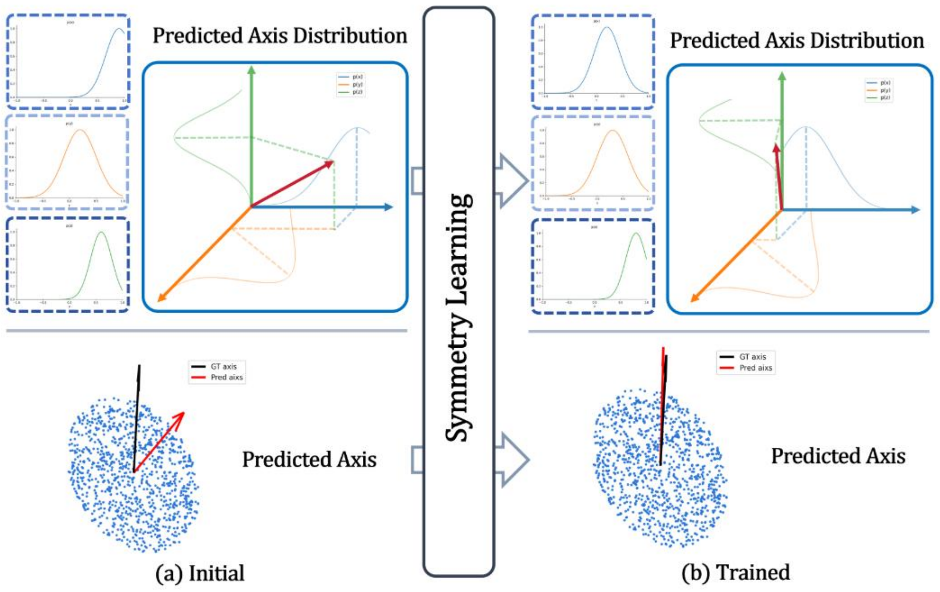
To address this, we aim to model the equivalent pose set without relying on explicit symmetry annotations.

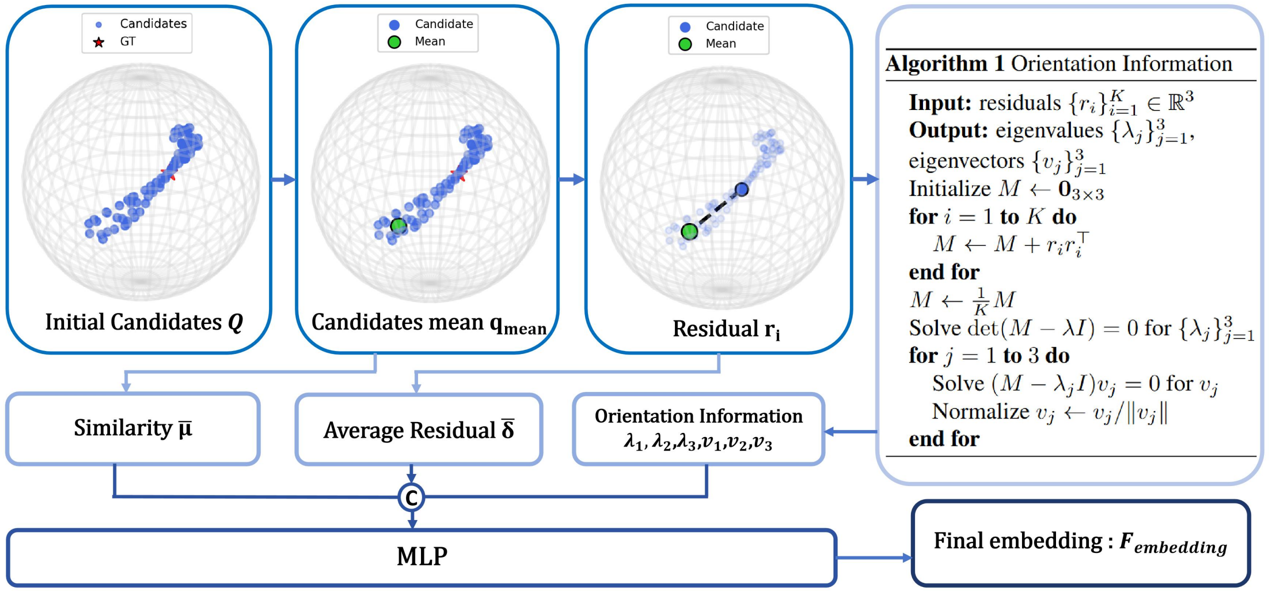
03 Framework ——Overall pipeline



Overview of our framework. First, we construct a backbone with our designed S^3 hyperspherical (HyperS3) layer to extract point cloud feature. Then, we generate quaternion candidates and refine each on the hyperspherical manifold of quaternio S^3. To better cope with the multi-hypothesis caused by symmetry, we additionally design a self-adaptive network to estimat the symmetry axes or planes, based on which we generate the corresponding equivalent solutions later. Finally, we aggrega all refined candidates and apply an additional refinement step, yielding q_{final}.

04 Method ——Symmetry and Candidates





Process of the self-adaptive symmetry learning process. The figure depicts the evolution of the framework from an initial state (a) to a trained state (b). The top panels show the implicit probability distribution along x, y, z axes.

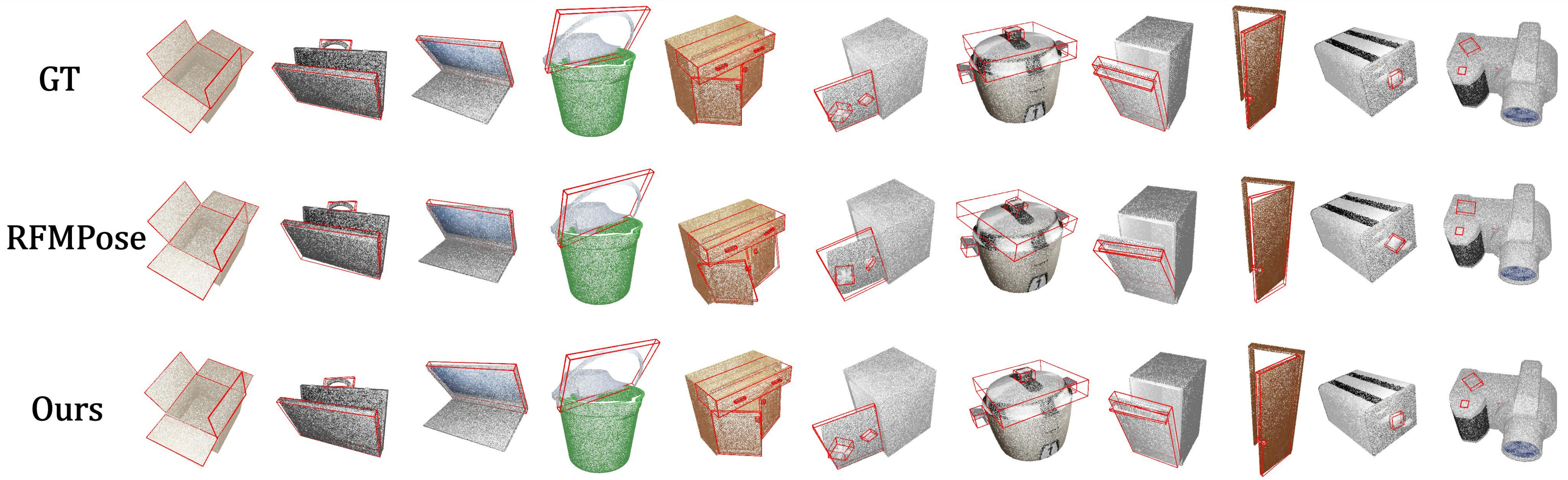
The illustration of the transformation process from the initial candidates Q to the feature embedding. The right demonstrates the extraction of orientation information. The left shows the extraction of other geometric information.

05 Result ——Main result

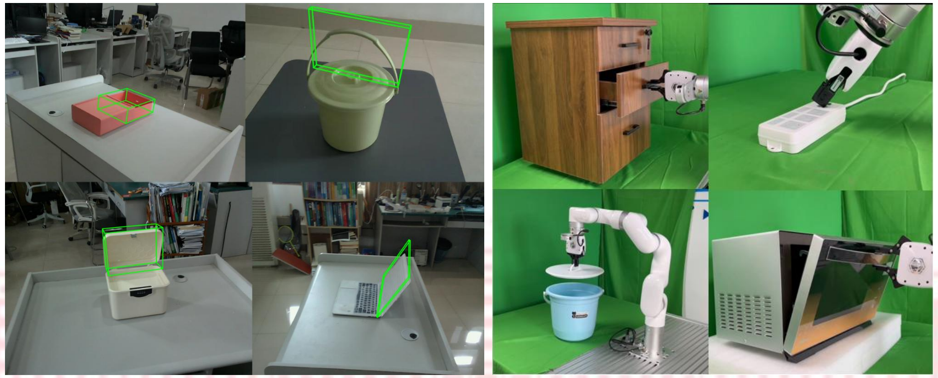
		Method	Sd.Dw	Sd.Ld	Sd.Bn	Rd.F.Hl	Hg.Kb	Ln.F.Hl	Hg.Hl	Hg.Dr	Hg.Ld	Avg
Seen	Rot.(°)↓	GAPartNet	5.03	7.82	2.41	12.74	**4.74**	10.39	10.15	8.46	7.72	7.71
		GASEM	9.30	4.45	9.11	11.96	5.31	17.11	9.42	8.44	6.93	9.11
		GenPose++	1.95	19.46	1.97	29.94	44.23	20.94	9.76	9.83	4.62	15.86
		RFMPose	1.53	21.11	1.87	31.94	57.97	15.13	8.05	11.45	4.20	17.03
		DFGAP	1.79	3.58	2.44	5.12	9.03	7.71	**6.04**	5.63	8.21	5.51
		SAFAG(Ours)	**1.44**	**0.46**	**0.69**	**3.56**	8.72	**3.07**	7.77	**1.00**	**2.39**	**3.23**
	Trans.(cm)↓	GAPartNet	0.075	0.032	0.009	0.078	0.010	0.015	0.039	0.038	0.038	0.037
		GASEM	0.069	0.023	0.005	0.051	0.005	0.015	**0.031**	0.047	0.080	0.036
		GenPose++	**0.024**	0.039	0.014	0.018	0.021	0.072	0.072	0.048	0.029	0.035
		RFMPose	0.046	0.049	0.031	0.035	0.069	0.103	0.087	0.079	0.042	0.060
		DFGAP	0.037	0.014	0.008	0.037	0.005	0.011	0.024	0.025	0.019	0.020
		SAFAG(Ours)	0.032	**0.012**	**0.002**	**0.004**	**0.003**	**0.010**	0.056	**0.014**	**0.018**	**0.016**
Unseen	Rot.(°)↓	GAPartNet	14.62	29.17	9.21	19.89	16.89	36.54	64.31	38.57	19.18	27.59
		GASEM	20.72	24.04	8.28	27.41	**12.86**	33.85	57.66	22.62	57.66	29.45
		GenPose++	14.73	26.51	39.90	10.38	86.45	16.61	18.11	16.00	56.26	31.66
		RFMPose	6.33	26.81	53.22	8.20	76.94	16.10	28.25	17.85	66.78	33.39
		DFGAP	4.15	10.28	5.98	4.33	19.17	20.03	**16.69**	12.87	13.24	11.86
		SAFAG(Ours)	**3.74**	**0.41**	**3.12**	**3.94**	28.98	**15.83**	33.35	**3.40**	**4.71**	**10.83**
	Trans.(cm)↓	GAPartNet	0.318	0.076	0.042	0.091	0.038	0.164	0.539	0.131	0.415	0.201
		GASEM	0.165	0.385	0.014	0.052	0.019	0.226	0.261	0.087	0.345	0.172
		GenPose++	0.133	0.024	0.037	0.021	0.255	0.033	**0.079**	0.054	0.094	0.081
		RFMPose	0.133	0.049	0.062	0.051	0.129	0.113	0.123	0.081	0.202	0.105
		DFGAP	0.063	0.035	0.012	0.053	**0.076**	0.126	0.163	0.033	0.192	0.084
		SAFAG(Ours)	**0.058**	**0.012**	**0.004**	**0.004**	0.025	**0.022**	0.399	**0.028**	**0.046**	**0.066**

Results of GAParts Pose estimation in terms of per-part-class Rotation error and translation error. We use degree error (noted as °) and distance error (noted as cm) as metrics. Ln.=Line. F.=Fixed. Rd.=Round. Hg.=Hinge. Hl.=Handle. Sd.=Slider. Ld.=Lid. Bn.=Button. Dw.=Drawer. Dr.=Door. Kb.=Knob. Rot.=Rotation. Trans.=Translation.

05 Result ——Qualitative and Real-world



Qualitative results on GAParts pose estimation. We compare our method with RFMPose. More qualitative results can be found in supplementary materials.



Thank You For Your Patience to listen!

合肥工业大学

HEFEI UNIVERSITY OF TECHNOLOGY