SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Graph-geometric Message Passing via a Graph Convolution Transformer for FKP Regression

Huizhi ZHU[1], Wenxia XU[1,*], Jian HUANG[2] & Baocheng YU[1]

[1]*Hubei Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan* 430205*, China;*
[2]*School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan* 430074*, China*

Appendix A Non-Euclidean Rotation Representations

Definition 1. One rotation representation belong to a mainfold $\mathcal{M}$. We define a surjective rotation mapping $\varphi : \hat{\mathbf{x}} \in \mathcal{M} \rightarrow \varphi(\hat{\mathbf{x}}) \in SO(3)$. A representation mapping $\Phi : \mathbf{R} \in SO(3) \rightarrow \Phi(\mathbf{R}) \in \mathcal{M}, \varphi(\Phi) = \mathbf{R} \in SO(3)$.

Definition 2. The network Euclidean output space $\mathcal{X}$ to the representation manifold $\mathcal{M}$. We define a manifold mapping $\pi : \mathbf{x} \in \mathcal{X} \rightarrow \pi(\mathbf{x}) \in \mathcal{M}, \mathbb{R}^n \rightarrow \mathcal{M}$. $\hat{\mathbf{x}} = \pi(\mathbf{x})$ on the mainfold $\mathcal{M}$.

We discuss several non-Euclidean rotation representations, as below.

Quaternion representation. Quaternions are 4D unit vector rotations, $\mathbf{q} \in \mathcal{S}^3$, where $\mathbf{q}$ and $-\mathbf{q}$ resprent a same ratation. A network's prediction can only be output linearly for $\mathbf{x} \in \mathbb{R}^n$. The normalization step is often defined to be the related manifold mapping function, $\pi_q(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|$. The rotation mapping $\varphi : \mathbf{q} \rightarrow \mathbf{R}$, as follows:

$$\varphi(\mathbf{q}) = \begin{pmatrix} 2({q_0}^2+{q_1}^2)-1 & 2(q_1q_2-q_0q_3) & 2(q_1q_2+q_0q_3) \\ 2(q_1q_2+q_0q_3) & 2({q_0}^2+{q_2}^2)-1 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 2({q_0}^2+{q_3}^2)-1 \end{pmatrix} \tag{A1}$$

where $\mathbf{q}=(q_0, q_1, q_2, q_3)$. When the representation mapping $\Phi : \mathbf{R} \rightarrow \mathbf{q}$, as follows:

$$\begin{cases} q_0 = \sqrt{1 + R_{00} + R_{11} + R_{22}}/2 \\ q_1 = R_{21} - R_{12}/(4 * q_0) \\ q_2 = R_{02} - R_{20}/(4 * q_0) \\ q_3 = R_{10} - R_{01}/(4 * q_0) \end{cases} \tag{A2}$$

6D representation. 6D representation [1] essentially to Stiefel manifold $\mathcal{V}_2(\mathbb{R}^3)$ through Gram-Schmidt orthonormalization, $\pi_{6D}(\mathbf{x})$. The first two columns of a rotation matrix is two orthogonal unit 3D vectors $(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2)$. The rotation mapping φ_{6D} is obtained by adding the third column $\hat{\mathbf{c}}_3 = \hat{\mathbf{c}}_1 \times \hat{\mathbf{c}}_2$.

9D representation. SVD orthogonalization [3] as the manifold mapping π_{9D} to map a raw 9D network output $\mathbf{M}$ to a rotation matrix. More specific representation learning details will be discussed in Section 3.4.

10D representation. 10D representation [2] achieve $\boldsymbol{\theta} \in \mathbb{R}^{10}$ to $\mathbf{q} \in \mathcal{S}^3$ by manifold mapping π_{10D}, which computing smallest eigenvalue of $\mathbf{A}(\boldsymbol{\theta})$.

$$\pi_{10D} = \min_{\mathbf{q} \in \mathcal{S}^3} \mathbf{q}^\top \mathbf{A}(\boldsymbol{\theta}) \mathbf{q} \tag{A3}$$

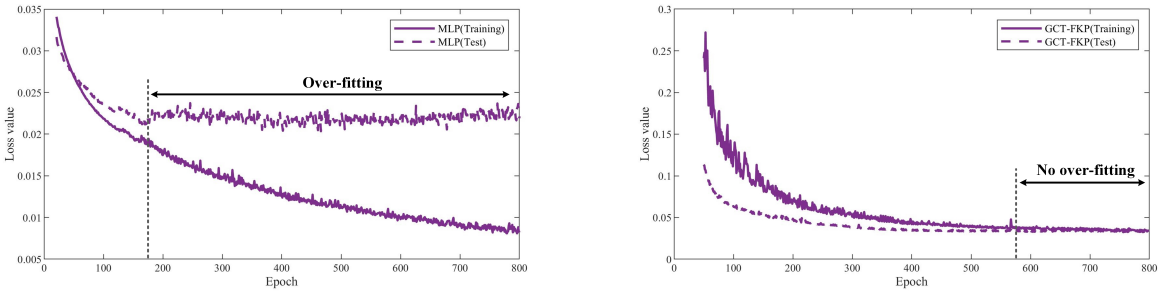
$$\mathbf{A}(\boldsymbol{\theta}) = \begin{pmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 \\ \theta_2 & \theta_5 & \theta_6 & \theta_7 \\ \theta_3 & \theta_6 & \theta_8 & \theta_9 \\ \theta_4 & \theta_7 & \theta_9 & \theta_{10} \end{pmatrix} \tag{A4}$$

since rotation mapping φ and representation mapping Φ are same as Eq.(A1) and Eq.(A2).

Appendix B Supplementary Experiment

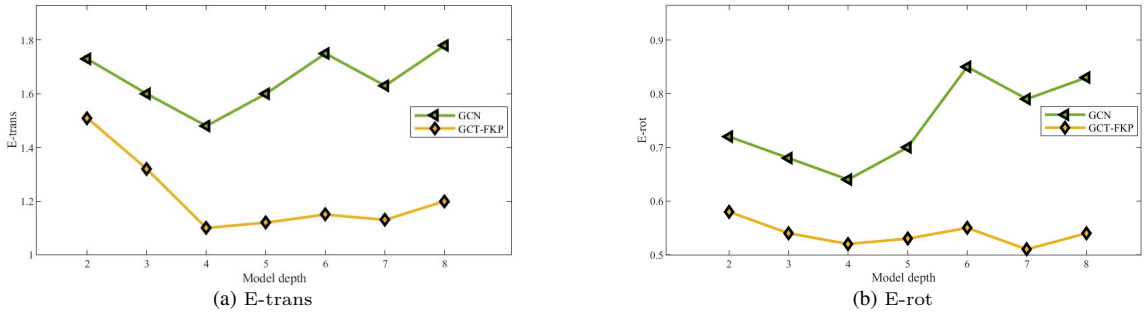
Over-fitting and over-soomthing. Figure B1 compares the training and test loss of GCT-FKP and MLP under similar parameters, whereas Figure B2 compares GCN and GCT-FKP at various model depths. The figures show that GCT-FKP substantially remedies the technical weaknesses of MLP, which is prone to overfitting, and GCN, which is sensitive to oversmoothing.

* Corresponding author (email: xuwenxia@wit.edu.cn)



(a) MLP method (b) GCT-FKP method

Figure B1 Results of the training and test loss comparison between MLP and GCT-FKP.



(a) E-trans (b) E-rot

Figure B2 Results of the E-trans and E-rot comparison between vanilla GCN and GCT-FKP w.r.t. model depth.

Model comparison. Table B1 provides a comparison of different models under similar conditions (including parameters, rotation representation, and loss functions, *etc.*).

Table B1 Model comparison for forward kinematics of the Gough-Stewart platform.

Methods	Translation Error (mm)				Rotation Error (deg)			
	Median↓	Mean↓	Acc(0.5mm)↑	Acc(1mm)↑	Median↓	Mean↓	Acc(0.5°)↑	Acc(1°)↑
CNN [4]	1.21	2.35	20.7	42.9	0.73	0.95	41.4	82.3
ResGCN [5]	0.87	1.14	32.3	61.8	0.64	0.82	43.6	84.5
GraFormer [6]	0.49	0.78	49.7	80.5	0.46	0.58	54.1	93.8
GCT-FKP (Ours)	**0.47**	**0.70**	**53.4**	**81.9**	**0.44**	**0.52**	**57.6**	**96.7**

Appendix C Dataset Generation Algorithm

Algorithm C1 Randomized generation dataset

Parameters: $l_{\min}, l_{\max}, \theta_{min}, \theta_{max}, \mathbf{\Omega}$;
Result: Dataset for the kinematics of Gough-Stewart platform;
1: $\mathbf{x} \sim \mathcal{U}(l_{\min}, l_{\max})$; ▷ Translation value $\mathbf{x} = (x\ y\ z)$;
2: $\mathbf{q}_{3D} \sim \mathcal{U}(\theta_{min}, \theta_{max})$; ▷ Rotation value (Euler-angles representation) $\mathbf{q}_{3D} = (\alpha\ \beta\ \gamma)$;
3: $\mathbf{R} \leftarrow \mathbf{R}_x(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\gamma) \in SO(3)$;
4: $\bar{\mathbf{l}}_j, \mathbf{l}_{oi} \leftarrow \mathcal{IK}(\mathbf{\Omega}, \mathbf{x}, \mathbf{R})$ ▷ Platform configuration information $\mathbf{\Omega} = (\mathbf{a}_i\ \mathbf{b}_i)$, Inverse kinematics solving function $\mathcal{IK}(\cdot)$;
5: $\mathbf{q}_{4D} \leftarrow \Phi(\mathbf{R})$; ▷ Rotation value (quaternion representation) $\mathbf{q}_{4D} = (q_0\ q_1\ q_2\ q_3)$;
6: $\mathbf{E} = \text{Edge}(\mathcal{V}, \mathcal{E}, \bar{\mathbf{L}}_j)$; ▷ Distance matrix (adjacent matrix with edge-length information) $\mathbf{E}$;
7: $\mathbf{e} = \text{Vec}(\mathbf{E})$; ▷ Vectorization operator $\text{Vec}(\cdot)$;
8: **return** $\text{Concat}(\mathbf{x}, \mathbf{q}_{4D}, \bar{\mathbf{l}}_j, \mathbf{e})$ or $\text{Concat}(\mathbf{x}, \mathbf{q}_{3D}, \bar{\mathbf{l}}_j, \mathbf{e})$; ▷ Get a dataset for the kinematics of the Gough-Stewart platform;

Dataset generation. To facilitate readers' understanding of the provided dataset, we offer a detailed explanation through pseudocode, as shown in Algorithm 1. Please note that the randomly generated motion-range space may include singular points.

References

1 Zhou Y, Barnes C, Lu J W, et al. On the continuity of rotation representations in neural networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, 5745–5753

2 Peretroukhin V, Giamou M, Greene W N, et al. A smooth representation of belief over so (3) for deep rotation learning with uncertainty. In: Proceedings of Robotics: Science and Systems (RSS), 2020
3 Levinson J, Esteves C, Chen K F, et al. An analysis of svd for deep rotation estimation. Adv Neural Inf Process Syst (NeurIPS), 2020, 33: 22554–22565
4 Zhu H Z, Xu W X, Wu J. Deep regression on quaternion: A forward kinematics neural network for study six-dof pose. In: Proceedings of IEEE International Conference on Cyborg and Bionic Systems (CBS), 2023, 432–437
5 Wang N Y, Zhang Y D, Li Z W, et al. Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proceedings of ECCV, 2018
6 Zhao W X, Wang W Q, Tian Y J. GraFormer: Graph-oriented transformer for 3D pose estimation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022, 20438–20447