Supplemental Material: Dynamic Hyperbolic Attention Network for Fine Hand-object Reconstruction

Zhiying Leng[1,2], Shun-Cheng Wu[2], Mahdi Saleh[2], Antonio Montanaro[3], Hao Yu[2], Yin Wang[1], Nassir Navab[2], Xiaohui Liang[1,4*], Federico Tombari[2]

[1] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China

[2] Computer Aided Medical Procedures, Technical University of Munich, Germany

[3] Politecnico di Torino, Italy

[4] Zhongguancun Laboratory, Beijing, China

{zhiyingleng,liang_xiaohui}@buaa.edu.cn, {shuncheng.wu,m.saleh}@tum.de, tombari@in.tum.de

Our main paper introduced a **D**ynamic **H**yperbolic **A**ttention **N**etwork (**DHANet**) for accurate hand-object reconstruction, which includes dynamic hyperbolic graph convolution (DHGC) and image-attention hyperbolic graph convolution (IHGC). In this supplemental document, we provide more detailed information about our method, including:

- Detailed architecture of our method in Section 7;
- More comparative experiments in Section 8;
- More reconstruction results in Section 9;

7. Detailed Architecture

7.1. Architecture for DHANet

Input:I, an 256*256 image			
Hand Branch		Object Branch	
Components	Architecture	Components	Architecture
Encoder **Output**: F^E_{sh}, F^E_{dh}, F_h	ResNet-18(I)	Encoder **Output**: F^E_{so}, F^E_{do}, F_o	ResNet-18(I)
Decoder **Output**: m_h	FC+MANO-layer(F_h)	Decoder **Output**: m_o	AtlasNet(F_o)
DHGC **Output**: $F^B_{h,mesh}$	DHGC(m_h)	DHGC **Output**: $F^B_{o,mesh}$	DHGC(m_o)
DHGC **Output**: $F^B_{h,mesh}$	DHGC($F^B_{h,mesh}$)	DHGC **Output**: $F^B_{o,mesh}$	DHGC($F^B_{o,mesh}$)
IHGC **Output**: $F^E_{o\to h}$	IHGC($F^B_{h,mesh}$, F^E_{sh}, F^E_{dh},F^E_{do}) MLPs	IHGC **Output**: $F^E_{h\to o}$	IHGC($F^B_{o,mesh}$, F^E_{so}, F^E_{do}, F^E_{dh}) MLPs
Decoder **Output**: m_h	FC+MANO-layer(F_h,$F^E_{o\to h}$)	Decoder **Output**: m_o	AtlasNet(F_o, $F^E_{h\to o}$)

Table 5. The architecture of our DHANet. "E" refers to Euclidean space. "B" refers to hyperbolic space.

In this section, we introduce the detailed network architecture for our DHANet. Given an input image I with size 256×256, DHANet reconstructs a hand mesh m_h of size 778×3, and an object mesh m_o of size 642×3. As listed in Table 5, our DHANet is a two-branch network, one for hand reconstruction and the other one for object reconstruction.

7.2. Architecture for DHGC

In this section, the detailed architecture of DHGC is statemented. DHGC consists of projection, graph construction, and hyperbolic graph convolution. Given a mesh in $l-1$-th layer, $v^{l-1,E}$, the l-th DHGC layer learns mesh features, $v^{l,B}$. The detailed implementation is listed in Table 6.

Components	Architecture
Exp-projection	geoopt.manifold.expmap0($v^{l-1,E}$)
Build graphs	geoopt_layers.KNN(k=20)
Transformation	poincare.MobiusLinear(3,64)
Activation	poincare.RadialNd(nn.LeakyReLU)
Aggregation	poincare.math.poincare_mean ()
Ouput:$v^{l,B}$	

Table 6. The detailed architecture of DHGC. We implemented DHGC based on the geoopt tool [2].

7.3. Architecture for IHGC

In this section, the detailed structure of IHGC is introduced. Given image features and mesh features, IHGC learns multi-modal features. Table 7 lists the detailed implementation of IHGC in the hand branch. The architecture of IHGC in the object branch is similar to the hand branch.

8. More comparative experiments

Comparison with spectral convolutions. Our DHANet is a spatial-based graph convolution defined in hyperbolic space. Nevertheless, we also itemize the comparison with a spectral-based graph convolution (Chebyshev graph convolution) in Table 8.

*corresponding author

Components	Architecture
Transformation for shallow image features	F^E_{sh}=nn.Conv1d(64, 32, kernel_size=1)
Exp-projection	F^E_{sh}=geoopt.manifold.expmap0()
Build graphs	f_2=geoopt_layers.KNN(k=20)
Features transformation	V=poincare.MobiusLinear(64,32)(concat(f_1,f_2))
Log-projection	V=geoopt.manifold.logmap0()
Transformation for deep image features of hand	F^E_{dh}=nn.Conv1d(64, 32, kernel_size=1)
Exp-projection	F^E_{dh}=geoopt.manifold.expmap0()
Build graphs	Q=geoopt_layers.KNN(k=20)
Log-projection	Q=geoopt.manifold.logmap0()
Transformation for deep image features of object	F^E_{do}=nn.Conv1d(64, 32, kernel_size=1)
Exp-projection	F^E_{do}=geoopt.manifold.expmap0()
Build graphs	K=geoopt_layers.KNN(k=20)
Log-projection	K=geoopt.manifold.logmap0()
Attention	att_weight=torch.matmul(Q, K) / d att_weight=softmax(att_weight) $F^E_{o\to h}$=torch.matmul(att_weight, V)
Aggregation	$F^E_{o\to h}$=torch.sum($F^E_{o\to h}$)
Ouput: $F^E_{o\to h}$	

Table 7. The detailed architecture of IHGC in the hand branch. We implemented IHGC based on the geoopt tool [2].

Method	Baseline [1]	Baseline+DGCNN	Baseline+ChebConv	ours
Hand error	28.1	25.9	27.6	**23.8**
Obj error	1579.2	1490.6	1489.8	**1236.0**

Table 8. Comparisons with different graph convolutions on FHB$^-$.

Comparsion on different datasets. Our method achieves the state-of-the-art result on FHB$^-$ dataset, while yeilds the lower accuracy than [1] on Obman dataset. The reason for that is dataset difference: Obman dataset is a synthetic dataset with complex and various backgrounds, while FHB$^-$ dataset is a real-world dataset with a simplistic kitchen environment background. This results in shallow image features learned on Obman containing more irrelevant background information, as shown in Figure 8. The more irrelevant features are, the more loose their projections are in hyperbolic space, and vice versa, as shown in Figure 8. Those loose shallow image features in hyperbolic space are unfavorable to searching the neighborhood between those features and mesh features. This causes our approach to work slightly worse in Obman dataset.

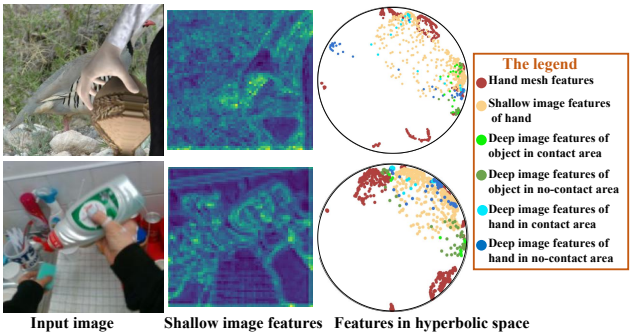

Figure 8. Feature differences between Obman dataset (the first row) and FHB$^-$ dataset (the second row).

9. Additional Results

As mentioned in Section 4.3 of the main paper, image features projected to hyperbolic space preserve the spatial relationships between hand and object in images, which is beneficial for modeling the hand-object interaction. We have shown some visualizations in Fig. 7 in the main paper. This supplemental material provides more visualizations for embeddings in hyperbolic space, as shown in Fig. 9. We observed that the light blue points are close to the light green points while other points are far away. This distribution is consistent with the spatial relationship between hand and object in the image.

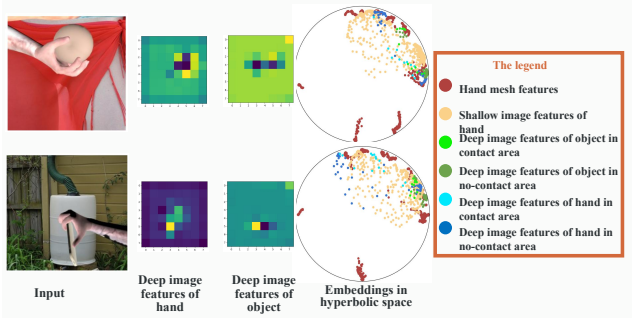

Figure 9. More visualizations for embeddings in hyperbolic space.

References

[1] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11807–11816, 2019. 2

[2] Max Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian optimization in pytorch, 2020. 1, 2