Practical Final Report: Graph-OMO and Puppet Hand

Hongbo Chen
Technische Universität München
Boltzmannstraße 15, 85748 Garching bei Munchen
hongbo.chen@tum.de

1. INTRODUCTION

In this work, we propose two hypotheses: Graph-OMO and Puppet Hand.

Graph-OMO is a new architecture based on the 1st stage Network(OMO) of OMOMO [3], which leverages Graph Convolution Network to make OMO can handle inputs with variable number of objects without retraining. The detailed architecture can be seen in Fig. 2.

Puppet Hand is new data correction method, which constructs a rigid-body hand according to the beta parameters of MANO hand [6] in Physical Simulator and can completely solve the fingers' penetration problem that happens frequently in hand MoCap data by leveraging rigid body collision in Physical Simulator.

2. RELATED WORK

OMOMO [3] achieves the task of generating coherent human motion sequence by taking a single object motion sequence as input. In this work, we modify OMOMO's 1st stage to achieve the task of generating hands positions sequence of multiple objects interaction.

HIMO [4] proposed a SOTA human multiple objects interaction dataset. We train and validate our model on this dataset.

Diffusion Model [1] is an effective generation model, which is used in this work to generate hand root position sequences given objects' motion sequences as conditions.

Graph Convolution Network [2] is an architecture that can handle Graph with different number of vertices and edges. This attribute is needed for this work to make it possible to process variable number of objects as input.

Attention [8] is an important block in NN architectures. Not only applied in OMOMO's original architecture to learn cross-frame correlation, Attention is also adapted to Graph-OMO as an aggregation method in this work.

MANO hand [6] and **SMPLX model** [5] are well-used parametric models for modeling hands and human. As MANO is integrated into SMPLX, we can obtain hands' joints layout for Puppet Hand construction by giving specific beta parameters to SMPLX layer [5]

MuJoCo [7] is an open-source physical simulation platform for machine learning and other possible application. In this work, we leverage the rigid body collision in simulation to solve the finger penetration problem in Mocap data.

3. METHOD: Graph-OMO

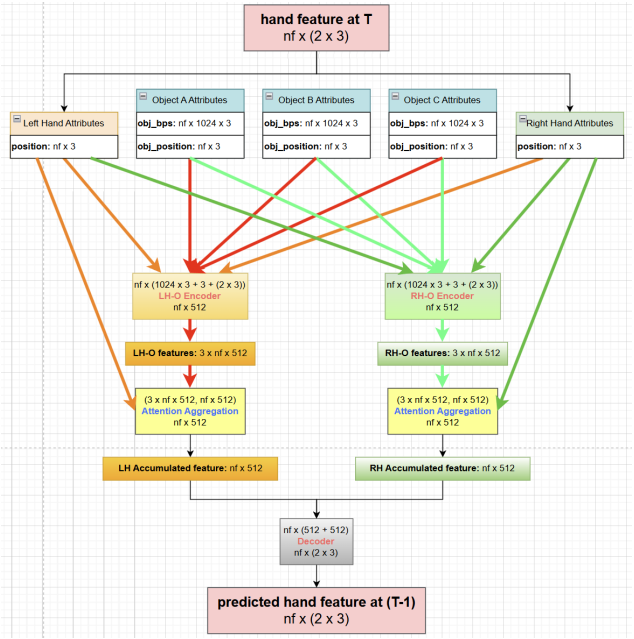

Figure 1. This picture shows the message passing procedure and the forward step of Graph-OMO

3.1. Data Representation

$G_\theta(x_t, c)$ is Graph-OMO which takes x_t and c as input and gives $\hat{x}_{t-1}$ the prediction of noised input at diffusion step $t-1$. x_t is noised input at diffusion step t with shape $(nf, 2*3)$ and represents root joint position sequence of both hands. x_t can be divided into left hand input x_t^{left} and right hand input x_t^{right}, each with shape

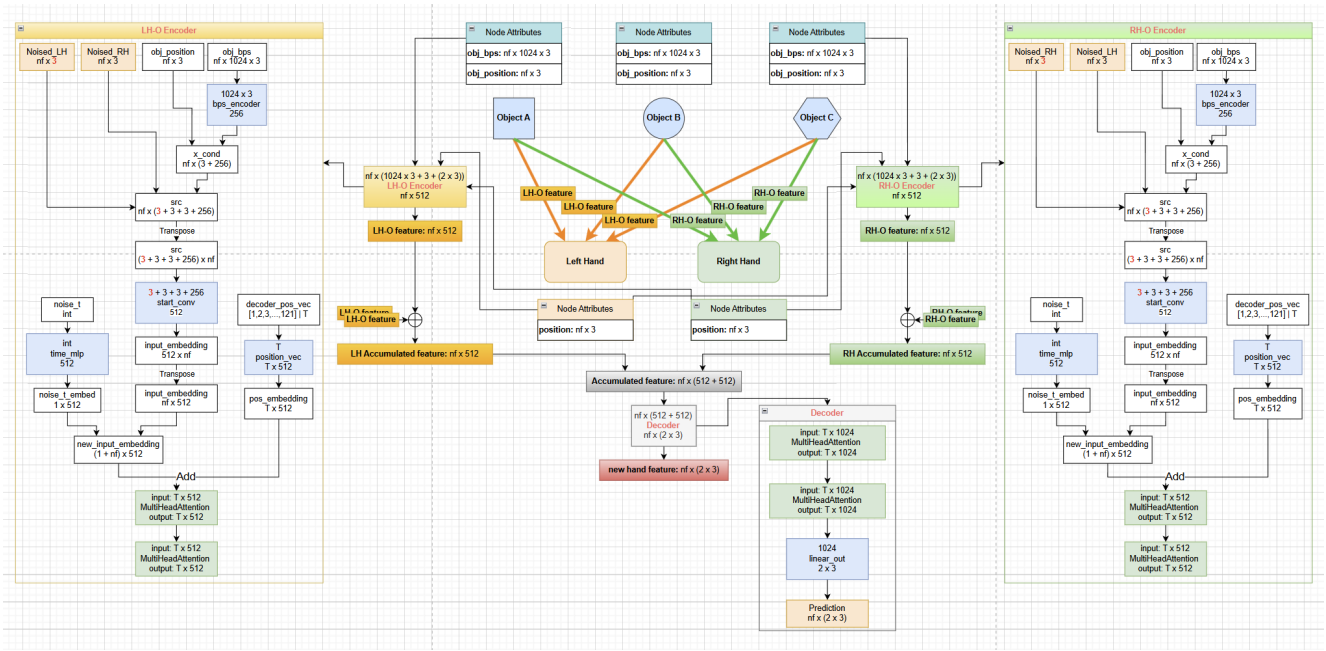

Figure 2. This figure shows the detailed architecture of Graph-OMO. There are object nodes and hand nodes. Each edge connects a hand and an object. At first, each edge feature is computed by edge encoder. Then edge features are aggregated to left hand and right hand separately. Finally, we concatenate the aggregated features of left hand and right hand, and send to decoder to obtain the prediction.

$(nf, 3)$. $c = [c_1, c_2, ..., c_n]$ is the conditon of G_θ. $c_i \in c$ with shape $(nf, 3 + 1024 * 3)$ is the feature of one object. d_v is the dimension of encoded feature at each frame. v_i^{left} is the encoded feature on the edge between object i and left hand computed by left hand encoder E_θ^{left} with shape (nf, d_v). f_{left} is the aggregated feature from $v^{left} = [v_1^{left}, v_2^{left}, ..., v_n^{left}]$ with shape (nf, d_v). f is the concatenated feature of f_{left} and f_{right} with shape $(nf, 2 * d_v)$. D_θ is a decoder in G_θ.

3.2. Pipeline

Diffusion Step: Diffusion model has forward process and reverse process. x_t is obtained by Eq. (1) in forward process [1]. In reverse process, we use G_θ to recover x_{t-1} given x_t and use the gradient of $Loss(x_{t-1}, G_\theta(x_t, c))$ to update the weights in G_θ.

$$x_t = \sqrt{a_t} x_0 + \sqrt{1 - a_t}\epsilon \tag{1}$$

Graph Construct: Interaction between a pair of hands and variable number of objects can be represented by graph data structure. In this work, we assign each object as a node with attribute c_i and right hand and left hand as separate nodes with attributes x_t^{left} and x_t^{right}. Edges is connected from left and right hand to all the objects.

Network Setup: Inspired by human's natural attribute that the motor control of left hand and right hand are charged separately by right brain and left brain. The architecture of this work has two separate edge feature encoders: $E_\theta^{left}(x_t, c_i)$ and $E_\theta^{right}(x_t, c_i)$. A decoder D_θ processes the concatenated features of both hands to output the prediction.

Message Passing: In message passing progress [2], which is executed in the forward process $\hat{x}_{t-1} = G_\theta(x_t, c)$, $E_\theta^{left}(x_t, c_i)$ process all the edges to get edge feature v_i^{left}. Then edge features of left hand are aggregated by specific aggregation function to obtain aggregated feature f_{left}. Same procedure goes for right hand to obtain f_{right}. Finally, we send concatenated feature f to decoder D_θ to obtain the predicted denoised input from x_t. Equations below describes concrete steps. Visualization of message passing can be seen in Fig. 1.

$$v_i^{left} = E_\theta^{left}(x_t, c_i) \tag{2}$$
$$v^{left} = [v_1^{left}, v_2^{left}, ..., v_n^{left}] \tag{3}$$
$$v_i^{right} = E_\theta^{right}(x_t, c_i) \tag{4}$$
$$v^{right} = [v_1^{right}, v_2^{right}, ..., v_n^{right}] \tag{5}$$
$$f_{left} = Aggre(v^{left}) \tag{6}$$
$$f_{right} = Aggre(v^{right}) \tag{7}$$
$$\hat{x}_{t-1} = D_\theta(concat([f_{left}, f_{right}])) \tag{8}$$

3.3. Aggregation

In Graph-OMO, we tried different aggregation strategies for gathering the edge features.

Sum aggregation: $Aggre(v) = \sum_{i=1}^{n} v_i$ (seen in Fig. 11) is used as baseline strategy.

Max aggregation: $Aggre(v) = \max(v_1, v_2, ..., v_n)$ is slightly better than sum aggregation $Aggre(v) = \sum_{i=1}^{n} v_i$ (seen in Fig. 11).

Attention aggregation (seen in Eqs. (9) and (10)), which computes the correlation of hand feature and edge feature of each frame, brings the best performance (seen in Fig. 10). Moreover, Attention aggregation uses left hand input feature x_t^{left} as prompt for $E_\theta^{left}(x_t, c_i)$ and right hand input feature x_t^{right} as prompt for $E_\theta^{right}(x_t, c_i)$. This design makes the left hand edge encoder $E_\theta^{left}(x_t, c_i)$ focus on left hand feature and the right hand edge encoder $E_\theta^{right}(x_t, c_i)$ focus on right hand feature.

$$Aggre(v^{left}, x_t^{left}) = softmax(\frac{T_\theta(x_t^{left})^T v^{left}}{\sqrt{d_v}})v^{left} \tag{9}$$

$$Aggre(v^{right}, x_t^{right}) = softmax(\frac{T_\theta(x_t^{right})^T v^{right}}{\sqrt{d_v}})v^{right} \tag{10}$$

4. METHOD: Puppet Hand

4.1. Key Idea

As the name "Puppet Hand" means, firstly we construct a rigid-body hand according to the joints' layout of Mano hand [6] given the beta parameters (seen in Fig. 3). Then we connect the joints of rigid-body hand with respective joints of MANO hand by virtual spring tendons(seen in Fig. 4). Setting rigid-body hand as active in rigid body collision simulation with objects and joints of MANO hand as inactive in simulation, we then use Mocap data's hand position sequence to update joints position of MANO hand and the rigid-body hand is driven by the MANO hand joints. Finally let the simulation run through the whole sequence, we obtain the joints position sequence of rigid-body hand as newly fixed data without penetration with objects.

4.2. Simulation Setup

Since Physical Simulation is a discrete approximation of the real continuous world, large number of computation steps per second is necessary for a stable simulation. In this work, we set simulation step as 1000 steps per second and the Mocap data sequence is played 30 frames per second for driving the rigid-body hand.

Since MuJoCo [7] cannot directly handle concave objects, we use CoACD [9] to divide a concave object into multiple convex sub-parts, then take all the sub-parts of this object as a group to join the simulation.

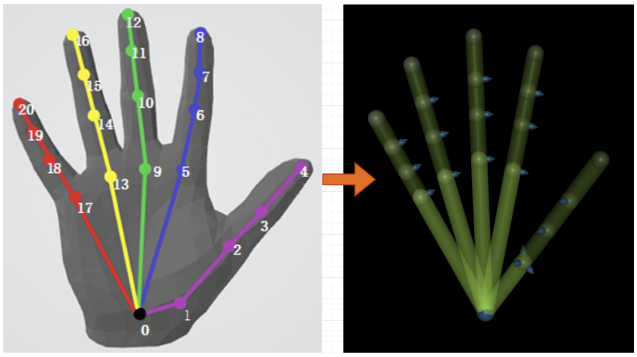

Figure 3. Left hand in this picture is MANO hand joints layout given certain beta parameters. Right hand in this picture is the rigid-body hand constructed to have the same skeleton as the left MANO hand. Each joint of rigid-body hand has certain degree of freedom according their indices

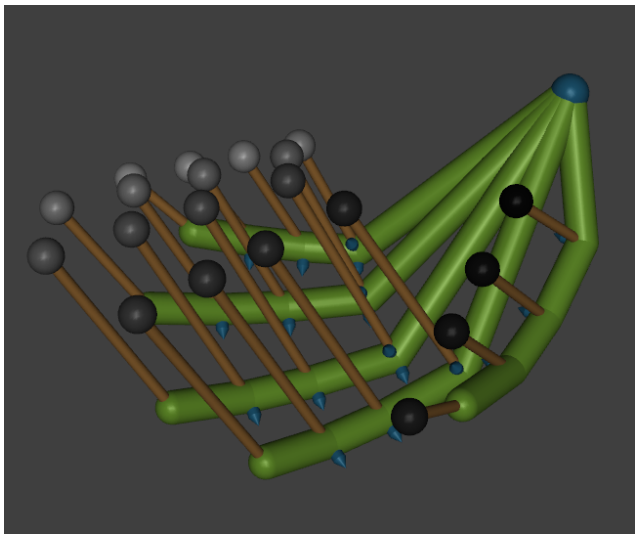

Figure 4. In this Figure, the green hand is the rigid-body hand, the spheres with gray scale degree are the joints of MANO hand, the orange lines represent the spring tendons that connect each joint of rigid-body hand with respective joint of MANO hand. The blue sphere denotes the root joint of both rigid-body hand and MANO hand, which means the root joint of both hands have a hard connection.

5. DATASET

We use HIMO [4], a HOI dataset for multiple objects interaction. For all the conducted experiemnts, we shuffle all the sequences in the dataset and split them with the ratio 8:1:1 for train, validation and test, then we process the data with sphere bps to capture objects' geometry feature and sliding window to get fix frame size.

For sphere bps sampling, which is used in OMOMO [3] to extract geometry feature for learning, we visualize the sampled result and find out that the feature distribution is

chaotic (seen in Fig. 5) and the sampled points more inside the sphere tend to have more incoherent and rapid variation compared to the sample points more outside. Therefore we propose a different sphere sampling strategy which rather sample points in a sphere volume, but sample points on the surface of the sphere evenly by using Fibonacci Grid. As Fig. 6 shows, the visualization is more well-organized and keep the needed features for learning. The experiment(seen in Fig. 7) shows new sampling strategy achieves similar performance compared with the original strategy.

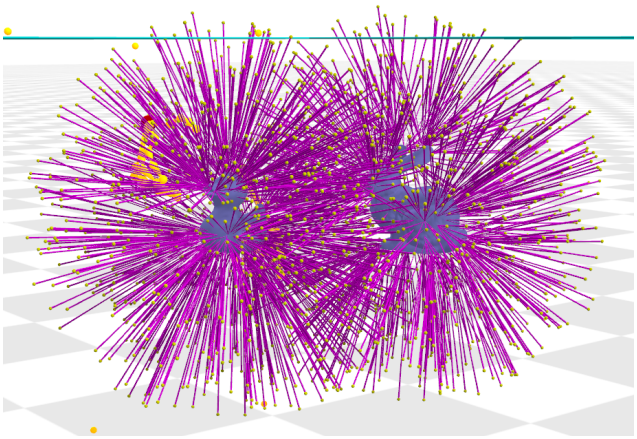

Figure 5. This figure shows OMOMO's original sampling strategy

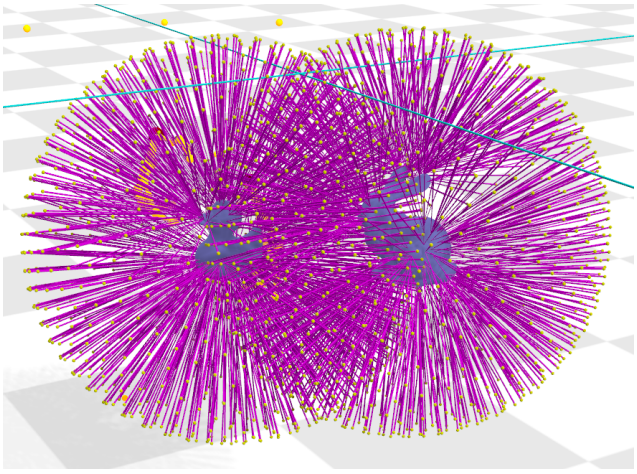

Figure 6. This figure shows new proposed sampling strategy. Similar to old strategy, yellow points are the sampled points, purple lines are the directional distance from sampled points to the closest point sampled on object's mesh surface

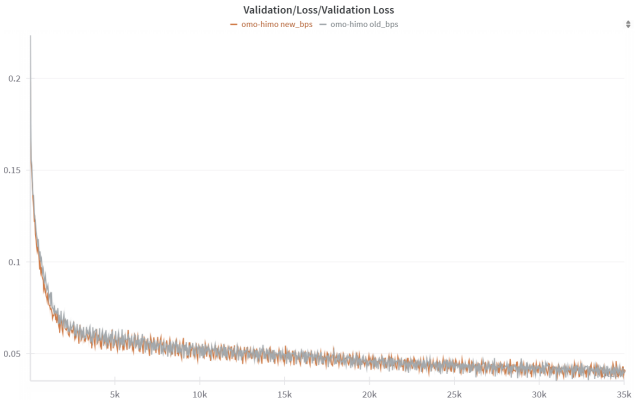

Figure 7. This figure shows new sampling strategy achieves same performance as the old strategy in validation

6. EXPERIMENT: Graph-OMO

6.1. Baseline Model

We obtain the baseline mode by modifying OMOMO's 1st stage model [3] to take more objects as input (shown in Fig. 8). This baseline model is named OMO-HIMO. Shown in Fig. 9, we didn't find a good metric to evaluate the performance of trained model. In this case, we use validation curve during training as a criterion.

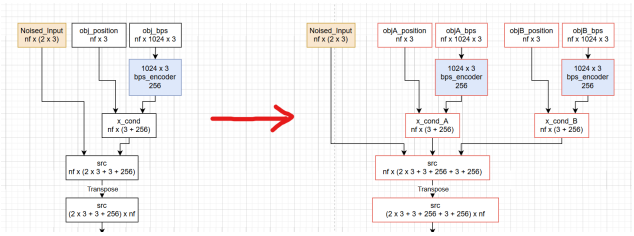

Figure 8. This figure shows how to modify OMOMO's 1st stage network for multiple objects tasks

6.2. Training Progress

Parameters: We conduct the training of different models with the same parameters: learning rate 2e-5, batch size 128 and gradient accumulation every 2 steps.

Mix Sample Training: As HIMO [4] dataset contains samples of 2 objects and 3 objects, to guarantee the parallel computing of training framework, we divide the dataset into 2-objects dataset and 3-objects dataset and use 2-objects dataloader and 3-objects dataloader alternatively during training.

Validation Curves: The performance of models is determined by the validation curve during training. In Fig. 10, we compared OMO-HIMO(baseline), Graph-OMO with sum aggregation and Graph-omo with attention aggrega-

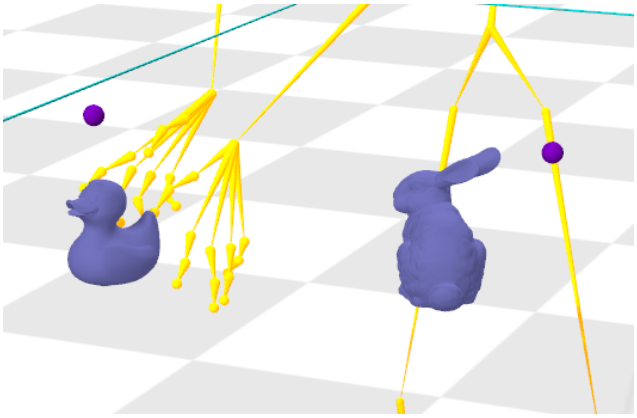

Figure 9. In this picture, two purple points denotes the predicted hand positions and the yellow skeleton is the ground truth. The duck in this picture is the only moved object. The prediction and the ground truth both make sense.

tion. Though the validation curves, we found that Grap-OMO outperforms the baseline model. Furthermore, attention aggregation brings further improvement compared with sum aggregation. In Fig. 11, we compared the validation curve of sum aggregation and max aggregation and found that max aggregation is slightly better than sum aggregation. The training curves of the finest Graph-OMO can be seen in Fig. 12.

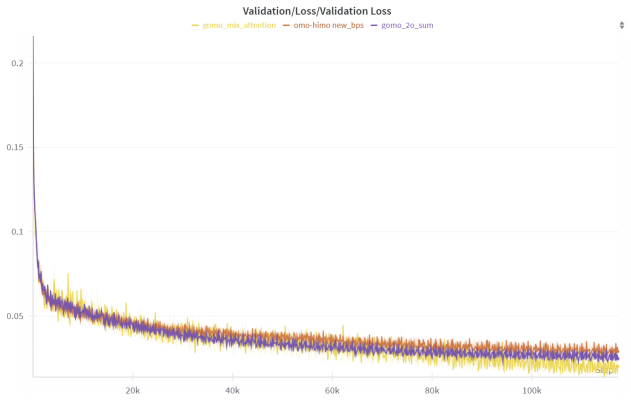

Figure 10. In this picture, the orange validation curve belongs to baseline model OMO-HIMO, the purple validation curve belongs to Graph-OMO with sum aggregation, the yellow validation curve belongs to Graph-OMO with attention aggregation

7. EXPERIMENT: Puppet Hand

Fig. 13 Fig. 14 and Fig. 15 show Puppet Hand can effectively correct penetration problem for different kinds of Mocap sequences. Since joint positions are embedded in the rigid-body hand, zero-penetration between joints and objects is guaranteed by physical simulator.

Figure 11. In this picture, the green validation curve belongs to Graph-OMO with sum aggregation, the red validation curve belongs to Graph-OMO with max aggregation

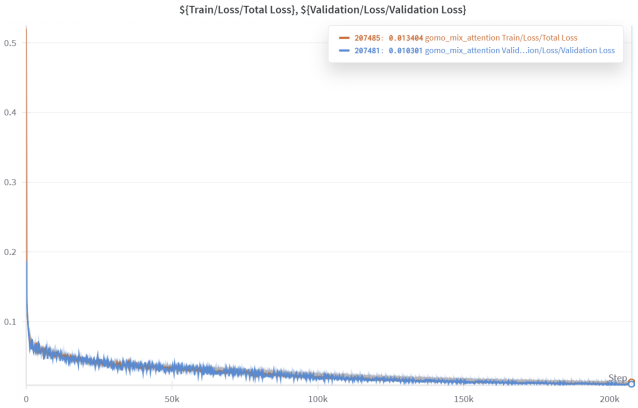

Figure 12. In this picture, we have train and validation curves of Graph-OMO model with attention aggregation in mix sample training. As more training steps accomplished, both beautiful curves have less vibration and don't show overfitting

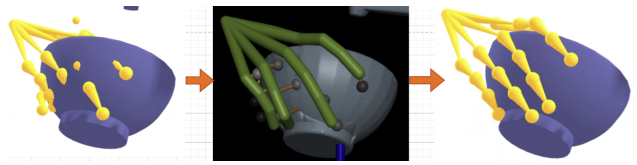

Figure 13. In this picture, left is the original data with penetration, middle is running puppet hand in simulation, right is corrected data.

8. CONCLUSION

Contribution: In this work we proposed Graph-OMO, a new architecture, the effectiveness of which has been observed through the validation curve in training progress. We also proposed Puppet hand, an effective method for fixing penetration in Mocap data.

Future work and Limitation: As OMOMO [3] used

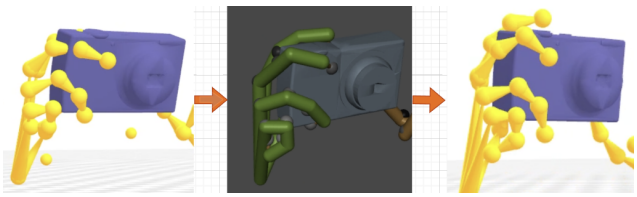

Figure 14. Same setting as Fig. 13

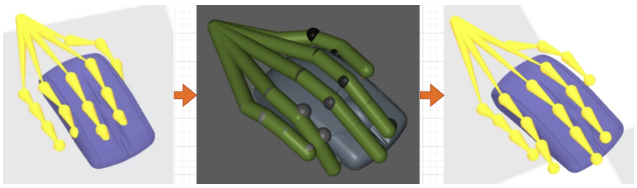

Figure 15. Same setting as Fig. 13

the same architecture for 1st stage task and 2nd stage task, Graph-OMO should also be capable of handling input with more complex features, such as orientation of hands and hand grasping degrees. Puppet hand as a correction method, can only handle the data with basically correct hand root pose capture. If the capture data has a big deviation or incoherence from the ground truth, then additional fix method is needed.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 1, 2

[2] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017. 1, 2

[3] Jiaman Li, Jiajun Wu, and C. Karen Liu. Object motion guided human motion synthesis, 2023. 1, 3, 4, 5

[4] Xintao Lv, Liang Xu, Yichao Yan, Xin Jin, Congsheng Xu, Shuwen Wu, Yifan Liu, Lincheng Li, Mengxiao Bi, Wenjun Zeng, and Xiaokang Yang. Himo: A new benchmark for full-body human interacting with multiple objects, 2024. 1, 3, 4

[5] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 1

[6] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. 1, 3

[7] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. 1, 3

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 1

[9] Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics*, 41(4):1–18, July 2022. 3