

INTERNAL



工作汇报

第四周

2024年1月26日



第一部分：浙大论文研读报告

Kinematic Motion Retargeting via Neural Latent Optimization for Learning Sign Language

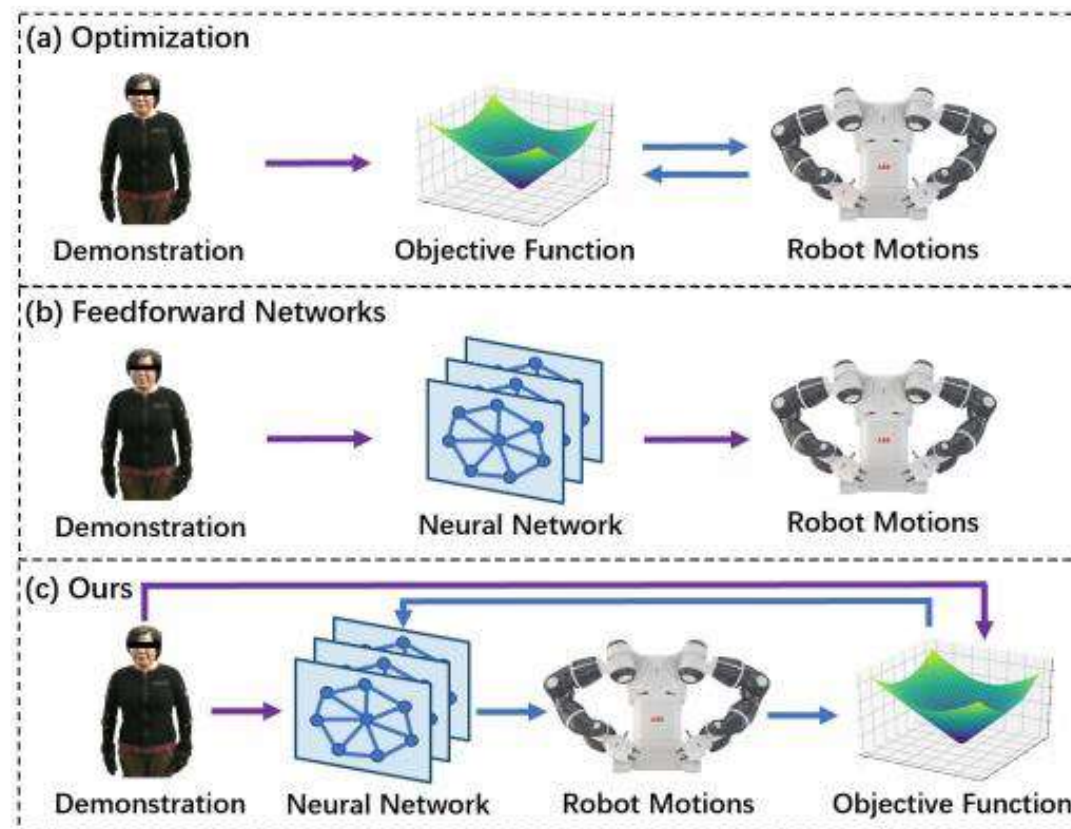
2024年1月26日

浙大 Kinematic Motion Retargeting via Neural Latent Optimization for Learning Sign Language

方法对比和引入

- ①传统的**基于优化**的方法**耗时且严重依赖于良好的初始化**,
 - ②最近使用**前馈神经网络**的研究**对看不见的运动泛化效果较差**。
- 二者都**忽略了人类骨骼和机器人结构中的拓扑信息**。

一种新的**神经潜在优化方法**来解决这些问题。潜在优化利用一个**解码器来建立潜在空间和机器人运动空间之间的映射**。之后,通过**搜索最优潜向量**,即可获得满足机器人约束的重定向结果。除了潜在优化之外,神经初始化还利用**编码器提供更好的初始化**,以实现更快更好的优化收敛。人类骨骼和机器人结构都被**建模为图**,以便更好地利用拓扑信息。



传统的基于优化的方法和使用前馈神经网络的方法在机器人运动重定向方面有不同的原理和优缺点。

传统的基于优化的方法：

****原理：****

1. ****优化问题****：基于优化的方法通常将运动重定向问题建模为一个优化问题，其中目标是最小化某个代价函数。代价函数通常包括与期望运动轨迹的差异、运动平滑性、能量消耗等方面的项。
2. ****数学优化****：通过数学优化技术（例如梯度下降），优化算法尝试调整机器人关节角度或轨迹，以最小化代价函数。

****优缺点：****

- ****优点****：这种方法可以提供较好的数学保证，尤其是在良好初始化的情况下，可以收敛到全局最优解。
- ****缺点****：但是，它可能对初始条件非常敏感，需要良好的初始化。此外，优化问题的求解通常比较耗时，不太适用于实时性要求高的应用。

使用前馈神经网络的方法

****原理：****

1. ****神经网络****：使用前馈神经网络，通常是递归神经网络（RNN）或长短期记忆网络（LSTM），将历史运动观测映射到未来的运动预测。
2. ****端到端学习****：通过端到端学习，神经网络直接学习从输入（历史运动观测）到输出（未来运动预测）的映射。

****优缺点：****

- ****优点****：神经网络可以学习非常复杂的映射关系，对于非线性、高度非凸的问题有较好的适应性。它们在一些情况下可以实现更好的泛化，对于看不见的运动也能做出预测。
- ****缺点****：由于神经网络的黑盒性质，它们可能在未见过的情况下产生不确定的预测。此外，神经网络的训练需要大量的数据，且可能对数据的分布变化敏感，因此泛化效果不如优化方法在一些场景中稳定。

为什么会出现这些缺点：

- ****传统优化方法****：敏感于初始条件是因为它们可能陷入局部最优解，而不同的初始化可能导致不同的局部最优解。
- ****神经网络方法****：泛化差可能是因为模型过于复杂，在训练数据之外的空间中泛化能力较差。此外，神经网络对数据的分布变化敏感，可能在测试时遇到不同于训练数据的分布。

综合来看，选择哪种方法通常取决于具体的任务需求、数据特性以及对实时性和泛化能力的要求。在一些场景中，**混合使用传统方法和神经网络方法**可能是一种有效的策略。

浙大 Kinematic Motion Retargeting via Neural Latent Optimization for Learning Sign Language

输入——数据

训练集 → 3个场景———“商业”、“火车站”和“西湖”
61个手语序列
9691帧

测试集 → 2个场景———“医院”和“介绍”
25个手语序列
4156帧

浙大

输入——图

\mathbf{x}_i : 节点i的节点特征

\mathbf{x}_j : 邻居节点j的节点特征

\mathbf{e}_{ji} : 节点j到节点i的边特征

人体骨架图

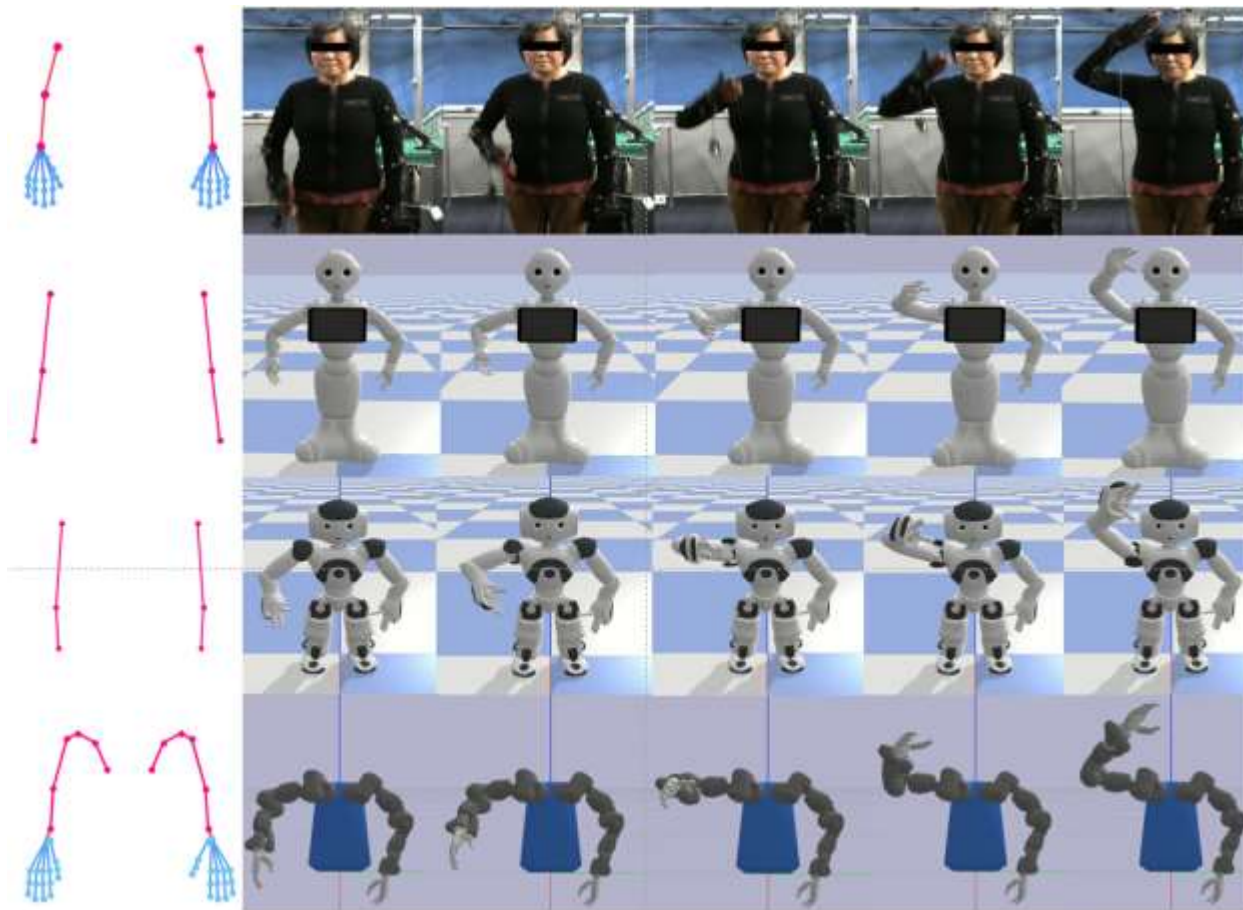
节点特征是**人体关节的位置和朝向（旋转矩阵）**

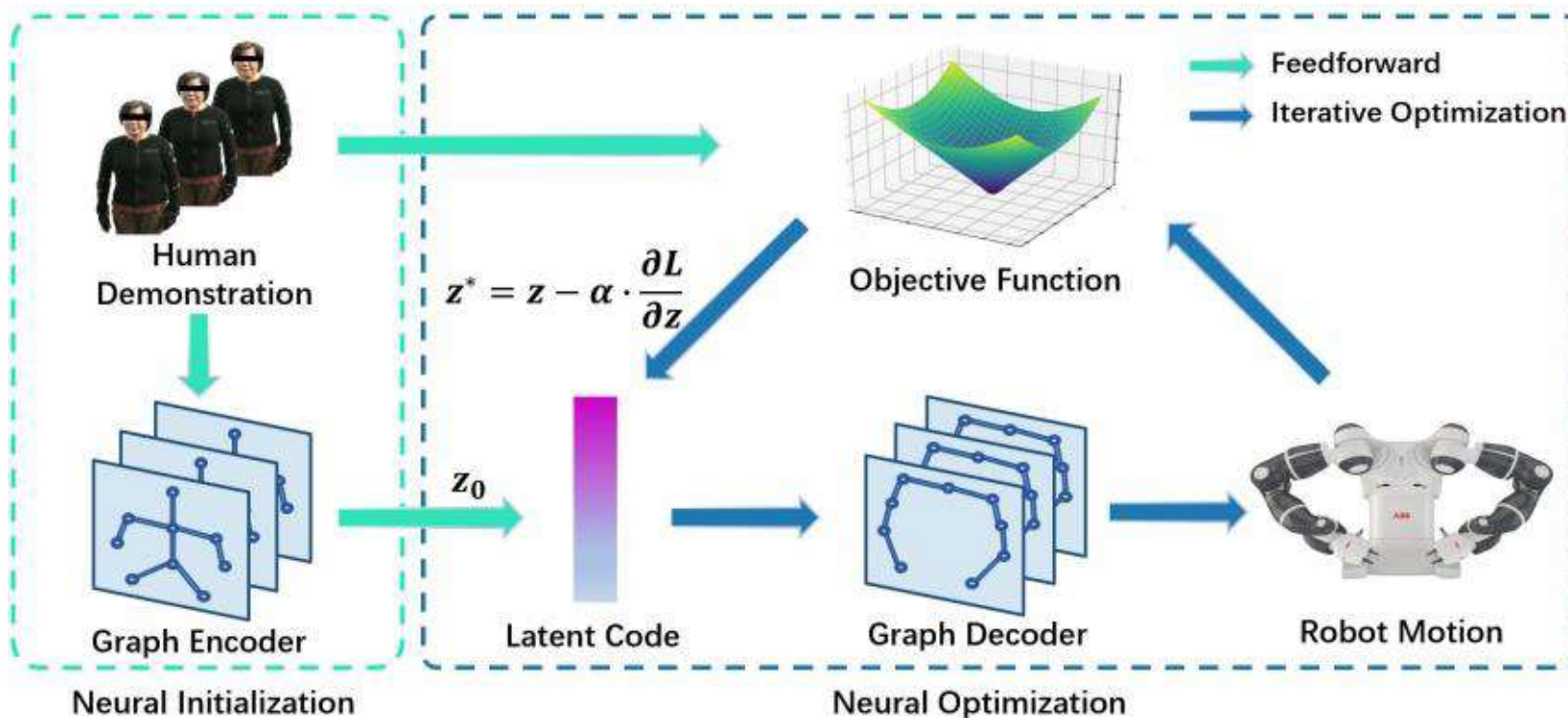
边特征是**人体关节位置之间的相对偏移**

机器人结构图

节点特征是对应机器人**关节的旋转角度**

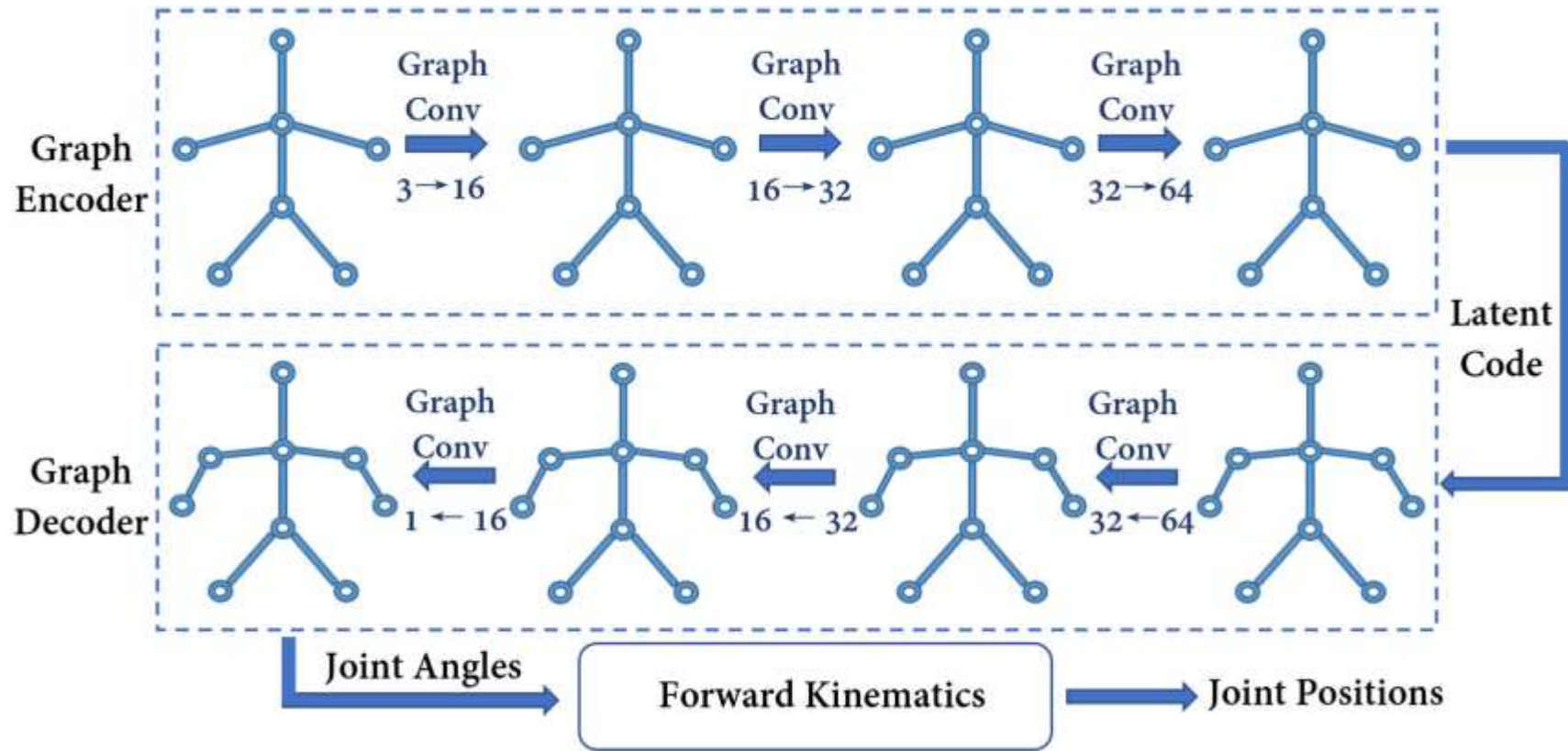
边特征是初始状态两个关节坐标系之间的**相对位移和旋转**





编码器以人的演示作为输入，生成一个更好的初始值 z_0 ，它应该接近最优的潜在码 z^* ，解码相应的机器人运动；计算目标函数的损失；通过梯度下降优化潜码。

浙大 模型细节



图编码器由**三层图卷积**组成，使用**线性层**将图编码器的输出特征映射到潜在代码。潜伏码与机器人关节的上下限拼接后，被馈送到图解码器中。

图解码器也由**三层图卷积**组成，通过 **\tanh 激活函数**，解码器的输出特征被限制在 -1 到 1 的范围内。然后将它们线性地重新映射到每个机器人关节的上限和下限。

正运动学模块：为了对比人和机器人之间动作的差异形成自监督，根据**关节角**计算各个**关节的位置和朝向**

浙大 损失函数

$$L = \lambda_{ee}L_{ee} + \lambda_{ori}L_{ori} + \lambda_{elb}L_{elb} \\ + \lambda_{fin}L_{fin} + \lambda_{col}L_{col}$$

末端执行器损失 L_{ee} : 末端执行器损失鼓励机器人匹配人类演示的末端执行器位置。

方向损失 L_{ori} : 方向损失是通过比较人和机器的末端执行器方向的差异来计算的。

肘关节丢失 L_{elb} : 鼓励其他关节的运动与演示相似，通过比较从肘关节到手腕的归一化向量的差异来计算

手指丢失 L_{fin} : 鼓励机器人模仿人类的手指动作，比较从掌指关节到指尖的向量，通过手指长度归一化

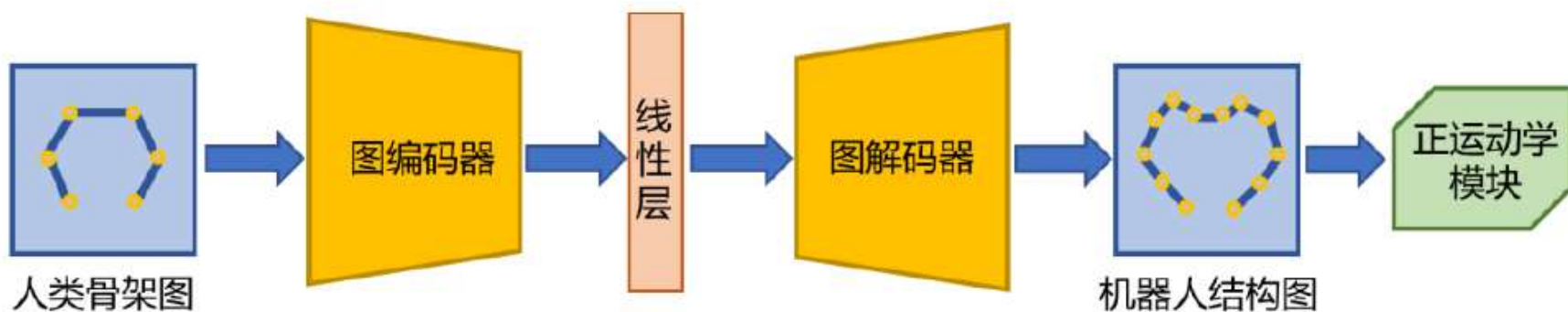
碰撞损失 L_{col} : 用于惩罚导致碰撞的机器人运动。将机器人手臂的链接建模为胶囊，并计算胶囊对之间的距离。如果这个距离小于不发生碰撞的最小距离，则计算损失。

λ_{ee} , λ_{ori} , λ_{elb} , λ_{fin} and λ_{col} 分别为它们的权值，其大小分别为1000、100、100、100和1000。

为适应其他任务，应该重新设计末端执行器损失和手指损失。

浙大

输入输出



输入： 演示者各个关节数据构成的图，包括关节的位置，四元数形式表示的关节旋转

图编码器： 从输入的图数据中提取每个关节的特征，得到隐空间中当前人类动作状态的编码 Z_h 。

线性层 L 将人类动作状态隐空间中的编码映射到机器人动作状态的隐空间 Z_r

图解码器： 利用 z_r 恢复出机器人的关节角状态

输出： 将机器人关节角状态输入到正运动学模块，得到机器人各个关节在世界坐标系下的位置和旋转

Dynamic Movement Primitive based Motion Retargeting for Dual-Arm Sign Language Motions

2021, ICRA, Zhejiang University

基于动态动作原语的双臂手语动作重定位

DMP 方法主要流程：

1. **学习动作和 DMP 参数化：**使用 DMP，研究者首先通过**演示或运动捕捉等手段**，获取人类执行手语动作的**运动数据**。通过学习过程，**DMP 的参数**（包括运动方程中的**权重、力场**等参数）被调整
2. **运动重定向：**在运动重定向时，DMP 被用于生成与原始手语动作**相似的运动轨迹**。将已学得 DMP 模型应用于目标机械手臂，以生成对应于手语动作的**关节角度序列**。
3. **双臂协同：**DMP 被分别用于每只手臂，以确保它们协同合作，以生成一个整体一致的双臂动作

优缺点：

优点：**实时性比较高**，DMP **具有适应性**，能够适应不同的环境和初始条件。这使得手语动作可以在不同的情境下执行，而无需重新手动调整关节角度

缺点：所有涉及到两只手相对位置的手势，均有比较大的误差

C-3PO: Cyclic-Three-Phase Optimization for Human-Robot Motion Retargeting based on Reinforcement Learning

201CRA, Taewoo Kim

机器人运动重定向的循环三相优化框架方法主要流程：

- 1.在第一阶段，使用 NTU-DB 和机器人参考运动训练骨架和机器人运动的潜流形。
- 2.在第二阶段，使用模拟器和奖励函数进行定量学习。
- 3.第三阶段，通过基于 dt 的微调对策略进行优化

优缺点：

优点：基于新提出的循环路径和滤波路径，定义了扩展的潜在状态和改进的奖励函数，这显示出比之前的工作更高的性能。在一个编码器-解码器网络中，实验证明了该模型可以在非马尔可夫环境中使用 MC 方法充分地执行人机运动重定向。

缺点：因为 20 年以前还没有很规范的数据集，故在骨骼与机器人之间的坐标对齐过程中为了进行适当的对齐，我们做了三个假设。

Towards a Natural Motion Generator: a Pipeline to Control a Humanoid based on Motion Data

19 IROS

基于优化源三维运动数据与目标人形机器人之间映射的高效运动重定向主要流程：

1. 运动重定向的准备
2. 运动传递的优化
3. 逆运动学计算关节轨迹
4. 关节轨迹的后处理

优缺点：

优点：提出了一种无梯度优化方法，用于提出的运动重定向方法：坐标下降贝叶斯优化，它优雅地混合了全局和局部优化方法。

缺点：此方法只适合从少拓扑结构到多拓扑结构的映射，由 human space 到 robot space 再使用 inverse kinematic 就会有诸如碰撞，解角超出限度等诸多问题。



第二部分：训练结果分析

Inspire Hand + Schunk Hand +Schunk Hand(none finger loss)+Schunk Hand(only finger loss)

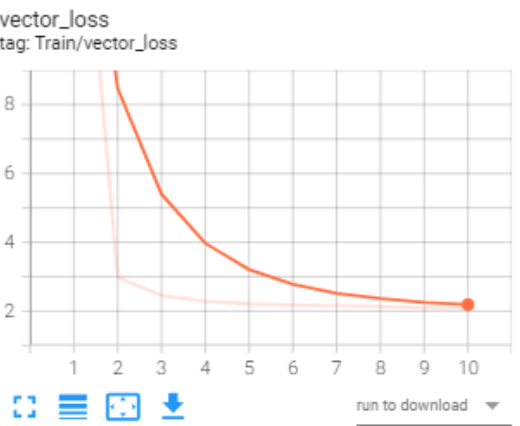
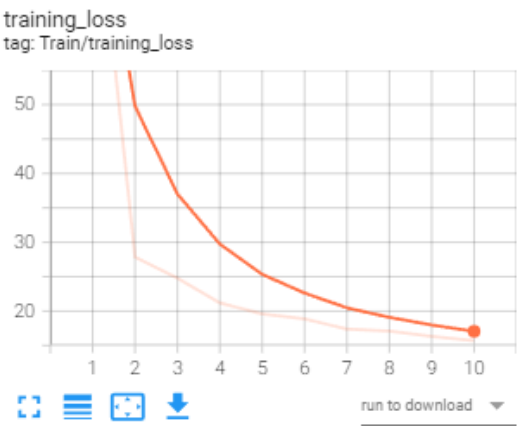
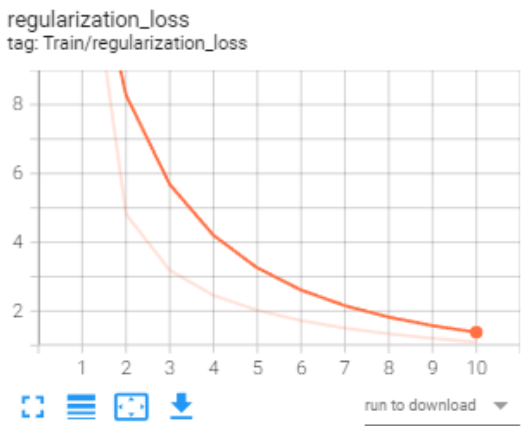
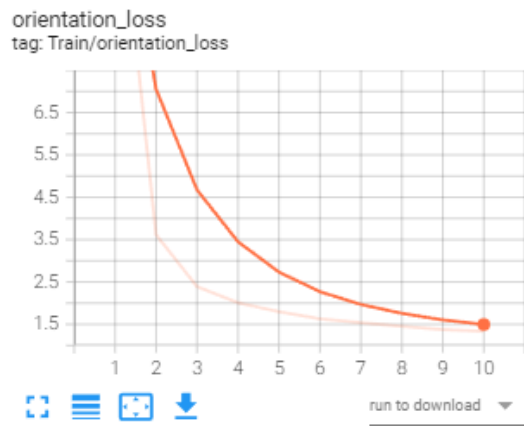
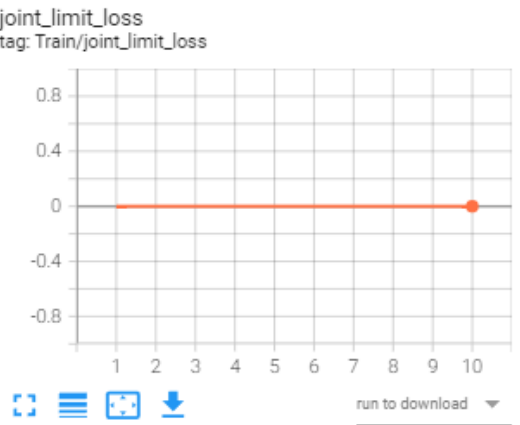
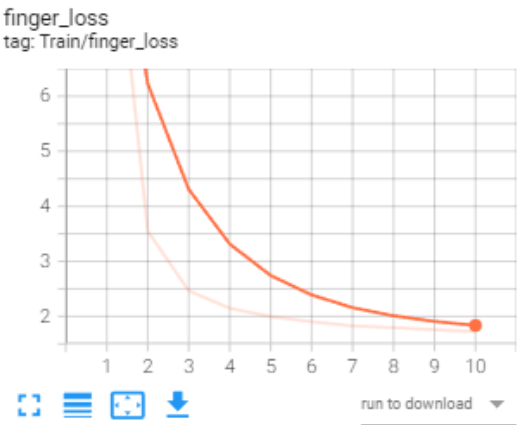
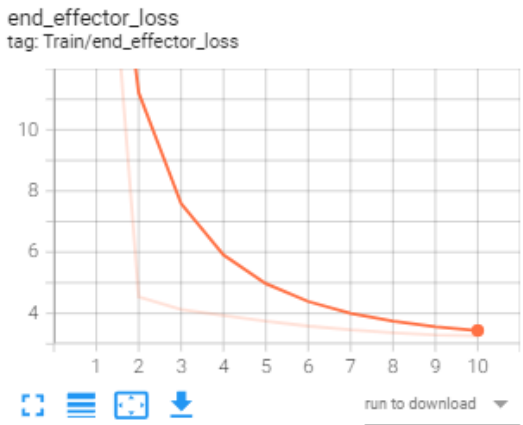
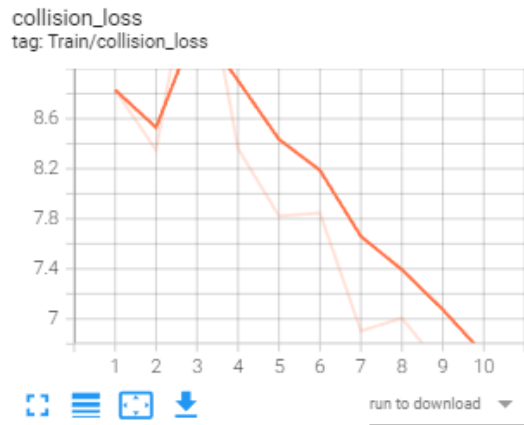
2024年1月26日

运行结果分析

Inspire手 (完整代码)

Train

Train



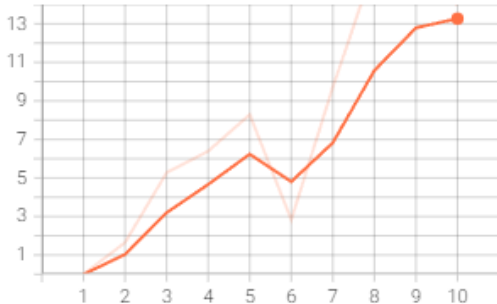
运行结果分析

Inspire手 (完整代码)

Test

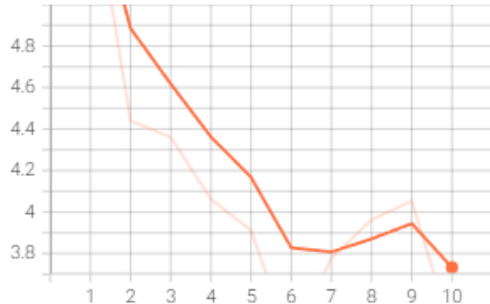
Test

collision_loss
tag: Test/collision_loss



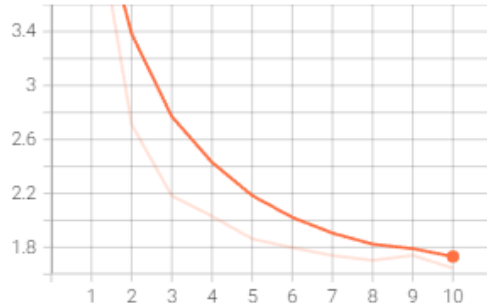
run to download

end_effector_loss
tag: Test/end_effector_loss



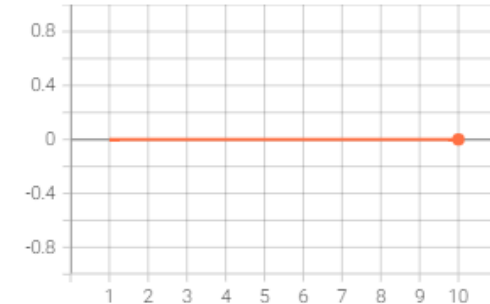
run to download

finger_loss
tag: Test/finger_loss



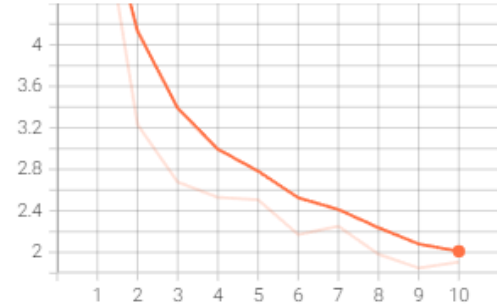
run to download

joint_limit_loss
tag: Test/joint_limit_loss



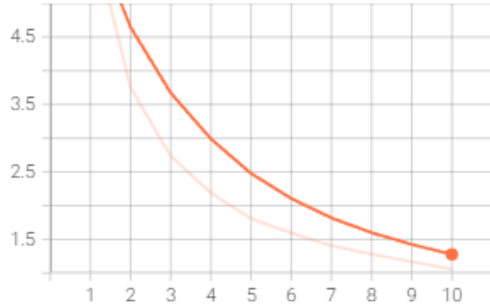
run to download

orientation_loss
tag: Test/orientation_loss



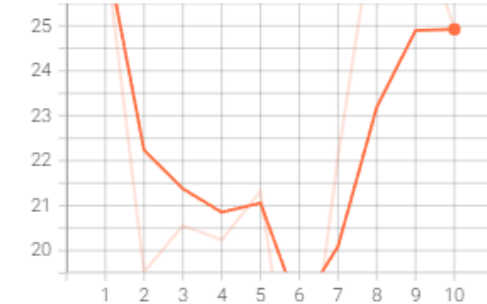
run to download

regularization_loss
tag: Test/regularization_loss



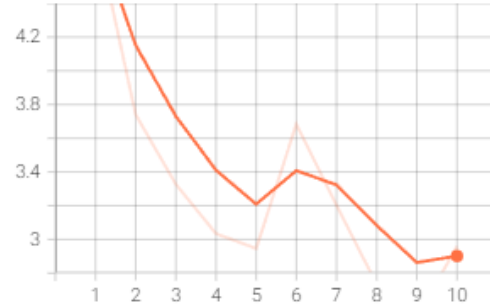
run to download

testing_loss
tag: Test/testing_loss



run to download

vector_loss
tag: Test/vector_loss



run to download

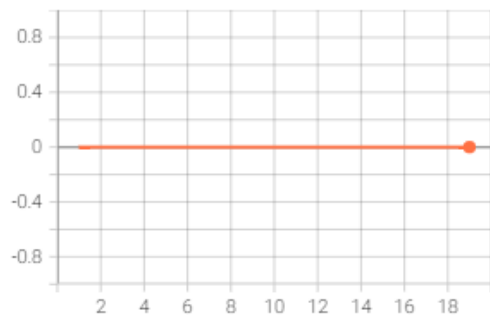
运行结果分析

Schunk手 (完整代码)

Train

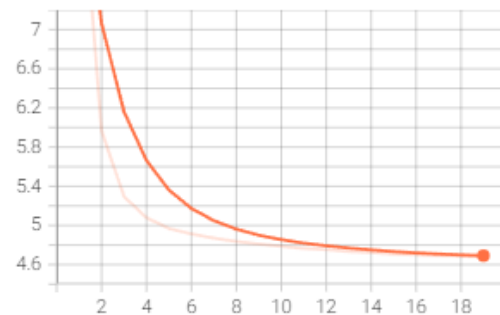
Train

collision_loss
tag: Train/collision_loss



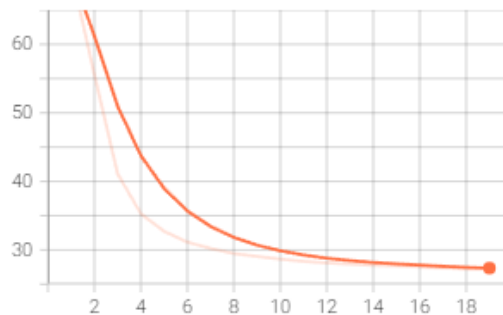
run to download

end_effector_loss
tag: Train/end_effector_loss



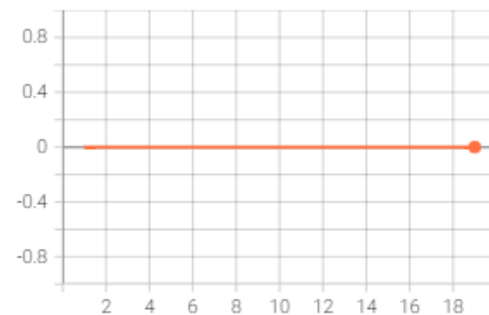
run to download

finger_loss
tag: Train/finger_loss



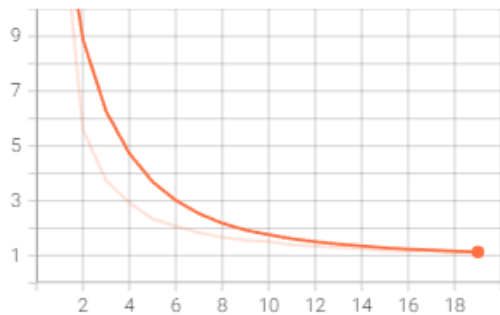
run to download

joint_limit_loss
tag: Train/joint_limit_loss



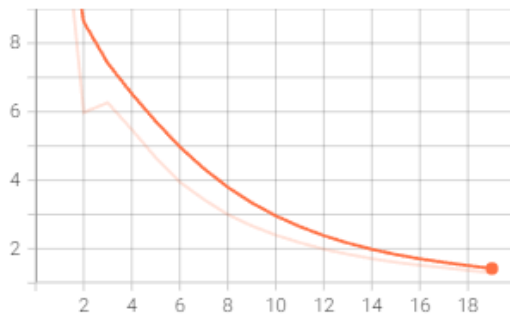
run to download

orientation_loss
tag: Train/orientation_loss



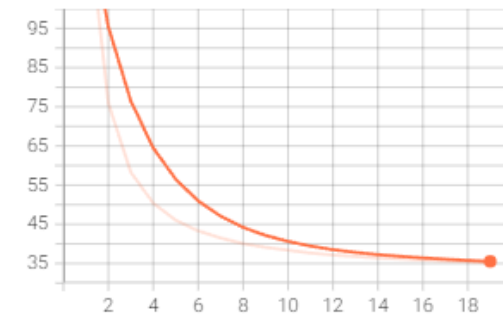
run to download

regularization_loss
tag: Train/regularization_loss



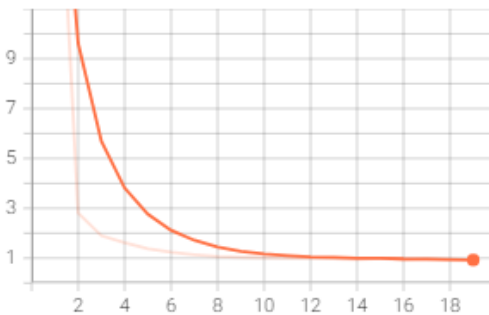
run to download

training_loss
tag: Train/training_loss



run to download

vector_loss
tag: Train/vector_loss



run to download

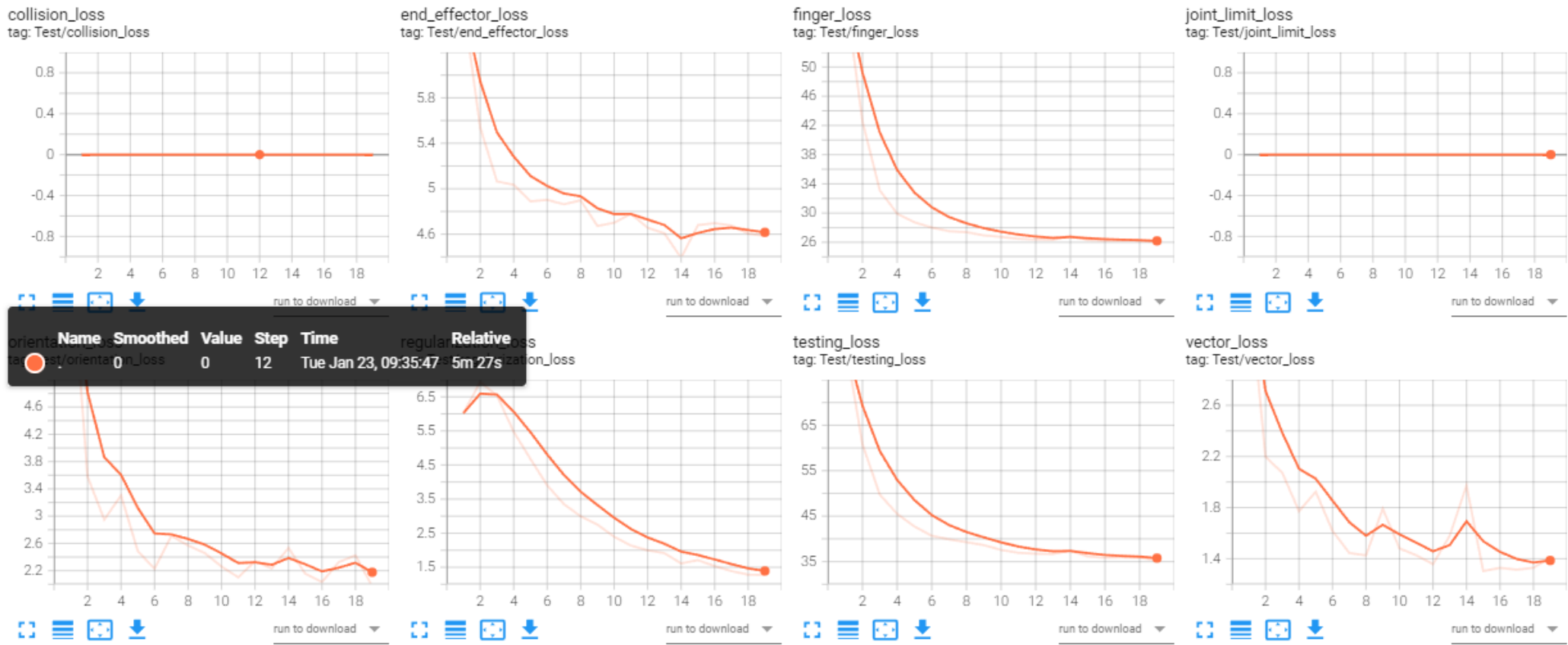
运行结果分析

Schunk手（完整代码）

Test

Test

8



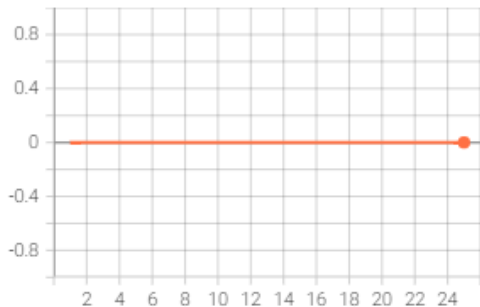
运行结果分析

Schunk手（代码去除Finger_loss和collison_loss）

Train

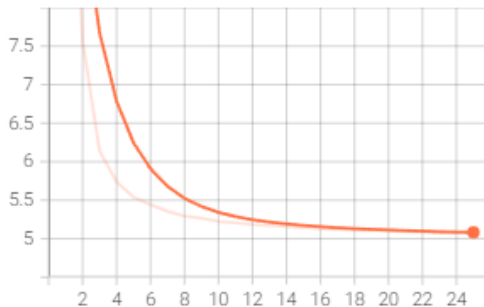
Train

collision_loss
tag: Train/collision_loss



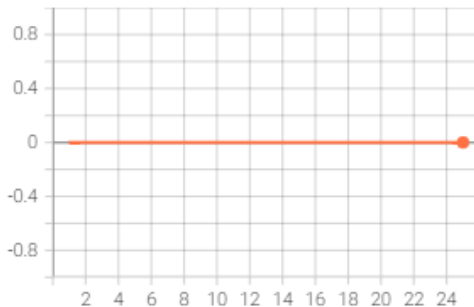
run to download

end_effector_loss
tag: Train/end_effector_loss



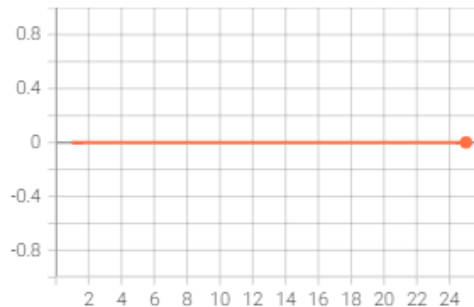
run to download

finger_loss
tag: Train/finger_loss



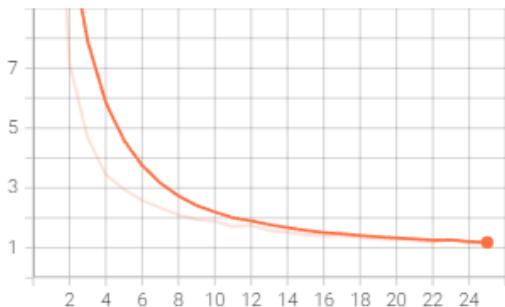
run to download

joint_limit_loss
tag: Train/joint_limit_loss



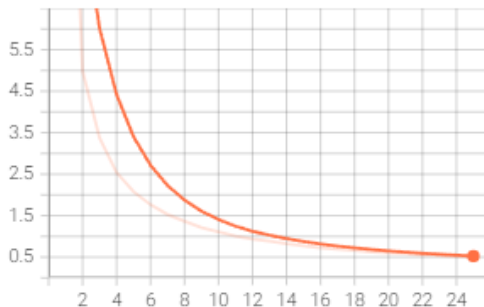
run to download

orientation_loss
tag: Train/orientation_loss



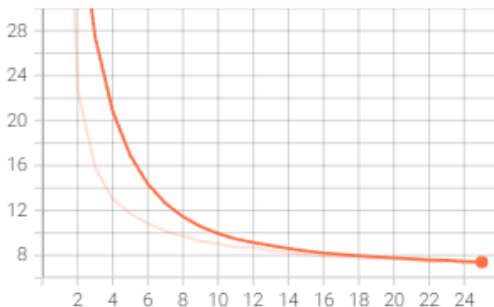
run to download

regularization_loss
tag: Train/regularization_loss



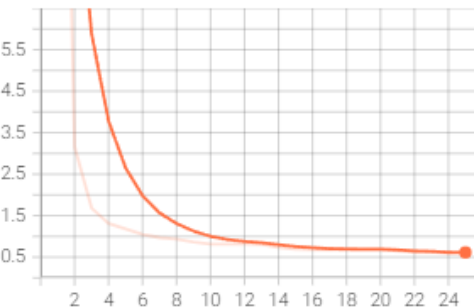
run to download

training_loss
tag: Train/training_loss



run to download

vector_loss
tag: Train/vector_loss



run to download

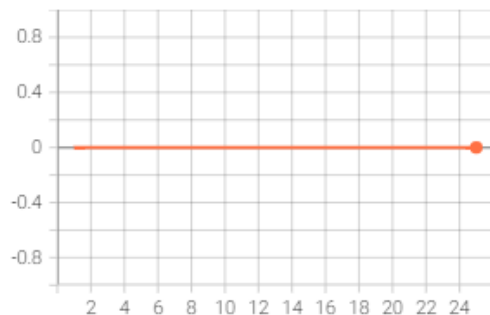
运行结果分析

Schunk手 (代码去除Finger_loss和collison_loss)

Test

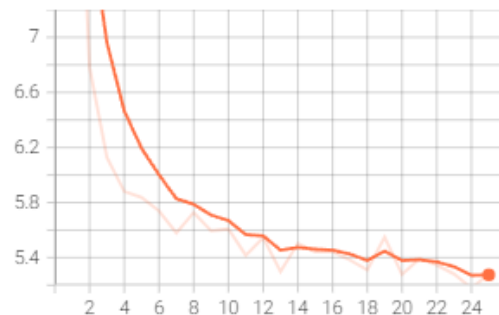
Test

collision_loss
tag: Test/collision_loss



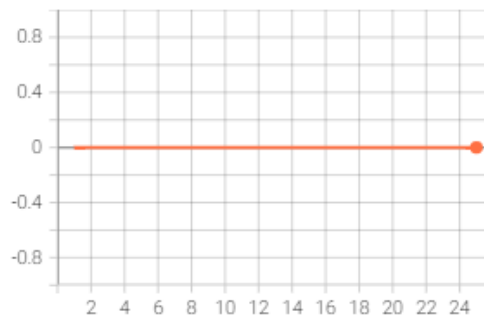
run to download

end_effector_loss
tag: Test/end_effector_loss



run to download

finger_loss
tag: Test/finger_loss



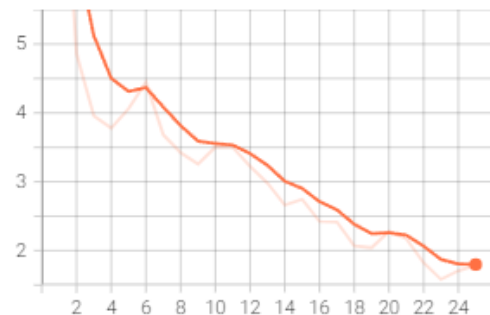
run to download

joint_limit_loss
tag: Test/joint_limit_loss



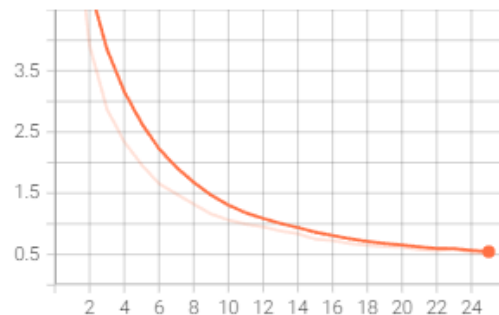
run to download

orientation_loss
tag: Test/orientation_loss



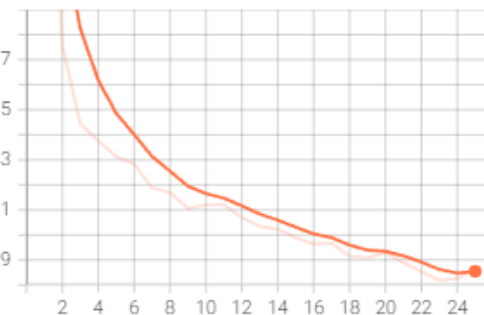
run to download

regularization_loss
tag: Test/regularization_loss



run to download

testing_loss
tag: Test/testing_loss



run to download

vector_loss
tag: Test/vector_loss



run to download

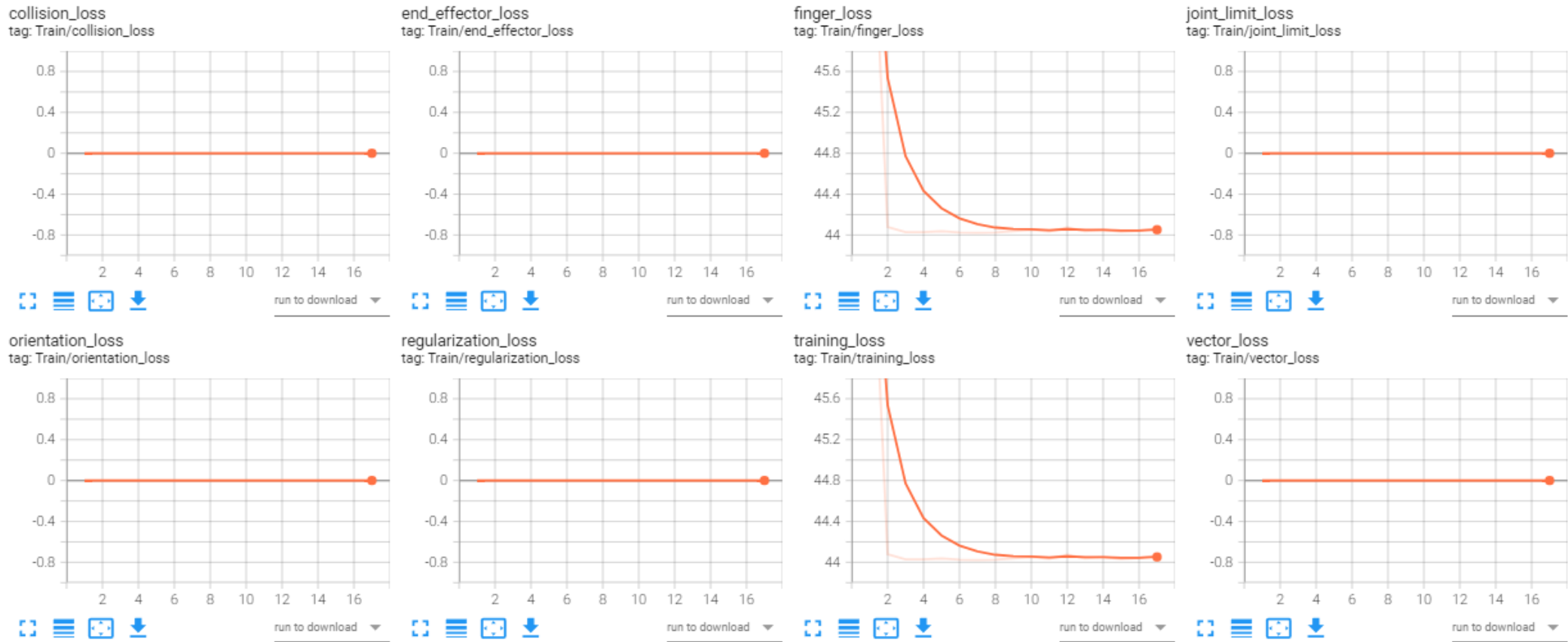


运行结果分析

Schunk手 (Only Finger_loss)

Train

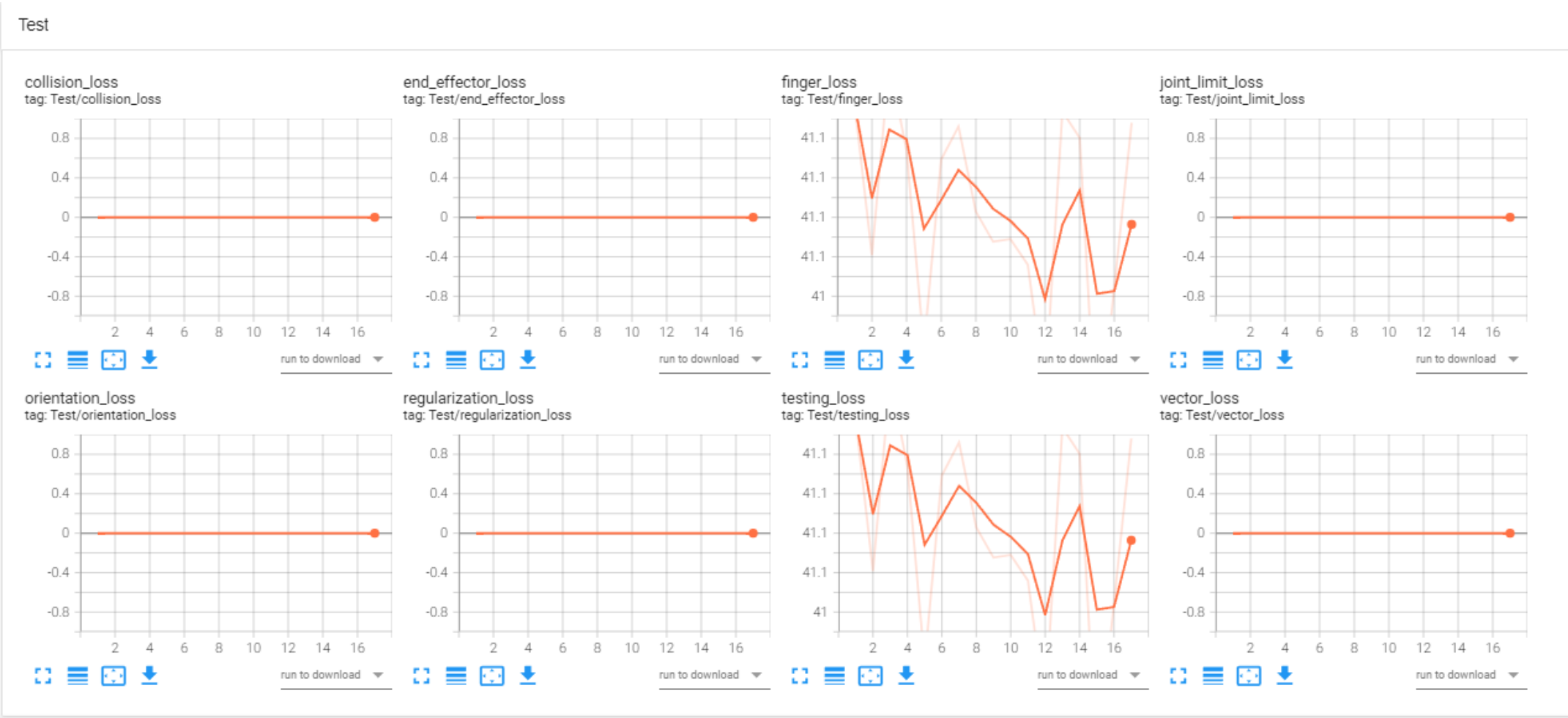
Train



运行结果分析

Schunk手 (Only Finger_loss)

Test





第三部分：相关文件的修改和当前问题分析

2024年1月26日

相关文件的修改

根据urdf模型文件改好了urdf2graph图形化文件，能够正常出模型图

前面实习生代码：

```
handGraph = inspireHand2graph(urdf_file='C:/Users/CNZIMA1/Documents/sign-yumi/dataHW/target/yumi-all/yumi_with_hands.urdf', cfg=hand_cfg)
print('hand: ', handGraph)
```

修改后：

```
handGraph = schunkHand2graph(urdf_file='C:\\Projects\\MA_Ziqi\\sign-yumi-schunk\\dataHW\\target\\yumi-with-schunkHands\\yumi_with_schunk_hands.urdf', cfg=hand_cfg)
print('hand: ', handGraph)
```


相关文件的修改

根据urdf模型文件改好了urdf2graph图形化文件，能够正常出模型图

URDF模型文件：

```
C: > Projects > MA_Ziqi > sign-yumi-schunk > dataHW > target > yumi-with-schunkHands > yumi_with_schunk_hands.urdf
464
465 <!-- HAND JOINTS -->
466 <joint name="LPart_Hand" type="revolute">
467   <parent link="Le1" />
468   <child link="Le2" />
469   <limit effort="1000.0" lower="0.0" upper="0.98786" velocity="1" />
470   <origin xyz="-0.01840 0.006 0.0375" rpy="0 0 0" />
471   <axis xyz="0 0 -1" />
472   <mimic joint="LThumb_Opposition" multiplier="1" offset="0" />
473 </joint>
474
475 <!-- THUMB JOINTS -->
476 <joint name="LThumb_Opposition" type="revolute">
477   <parent link="Le1" />
478   <child link="Lthumb_1" />
479   <limit effort="1000.0" lower="0.0" upper="0.9879" velocity="1" />
480   <origin xyz="0.01690 0.02626 0" rpy="0 0.2618 1.571" />
481   <axis xyz="0 0 1" />
482 </joint>
483
```

相关文件的修改

根据urdf模型文件改好了urdf2graph图形化文件，能够正常出模型图

前面实习生代码

```
hand_cfg = {  
    'joints_name': [  
        'yumi_link_7_r_joint',  
        'Link1',  
        'Link11',  
        'Link1111',  
        'Link2',  
        'Link22',  
        'Link2222',  
        'Link3',  
        'Link33',  
        'Link3333',  
        'Link4',  
        'Link44',  
        'Link4444',  
        'Link5',  
        'Link51',  
        'Link52',  
        'Link53',  
        'Link5555',  
    ],  
}
```

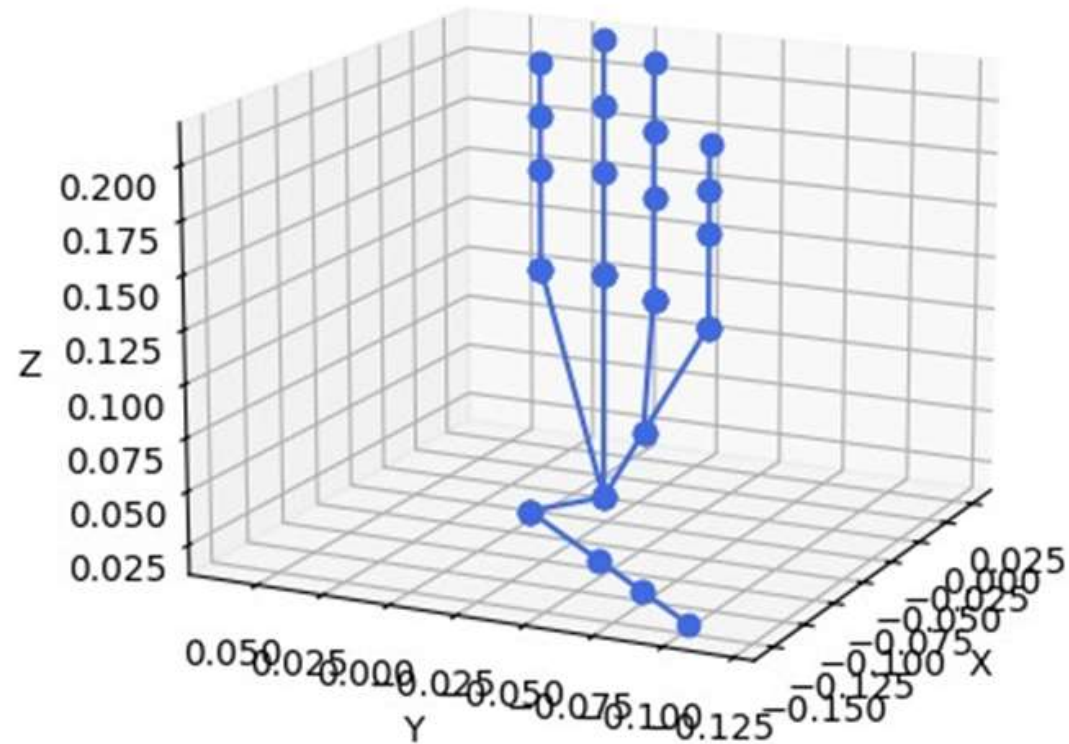
修改后:

```
hand_cfg = {  
    'joints_name': ['yumi_link_7_l_joint',  
        'LThumb_Opposition', 'LThumb_Flexion', 'LThumb_3', 'LThumb_4', 'LThumb_ee',  
        'LIndex_Spread', 'LIndex_Proximal', 'LIndex_3', 'LIndex_4', 'LIndex_ee',  
        'LMiddle_Spread_Dummy', 'LMiddle_Proximal', 'LMiddle_3', 'LMiddle_4', 'LMiddle_ee',  
        'LPart_Hand',  
        'LRing_Spread', 'LRing_Proximal', 'LRing_3', 'LRing_4', 'LRing_ee',  
        'LPinky_Spread', 'LPinky_Proximal', 'LPinky_3', 'LPinky_4', 'LPinky_ee',  
    ],  
}
```

相关文件的修改

根据urdf模型文件改好了urdf2graph图形化文件，能够正常出模型图

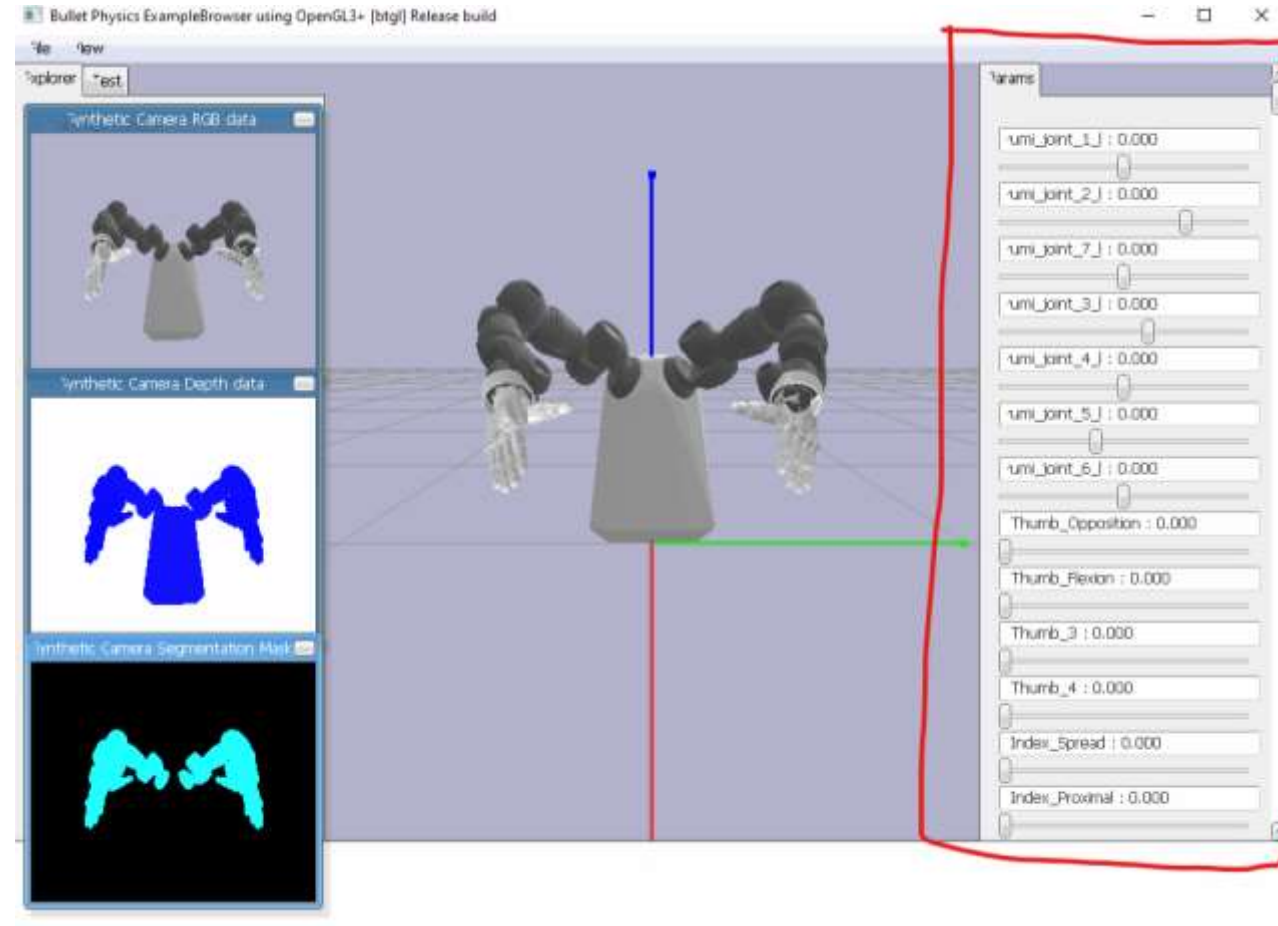
输出结果：



当前问题分析

手部模型大拇指错位问题导致Finger Loss居高不下

经过对比urdf模型文件和yumi-gym仿真环境中的具体模型，发现当前dataset_HW文件中的手部模型设定无错误



当前问题分析

手部模型大拇指错位问题导致Finger Loss居高不下

通过输出的模型图分析大拇指是否轴向定位（axis）错误，经过调试发现不是此处出错。

下一步就是继续分析其他数据处理部分，解决手部模型大拇指错位问题

```
LPart_Hand [ 0.  0. -1.]
LThumb_Opposition [0. 0. 1.]
LThumb_Helper [1. 0. 0.]
LThumb_Flexion [ 0.  0. -1.]
LThumb_3 [ 0.  0. -1.]
LThumb_4 [ 0.  0. -1.]
LThumb_ee [1. 0. 0.]
LIndex_Spread [ 0.  0. -1.]
LIndex_Proximal [0. 0. 1.]
LIndex_3 [0. 0. 1.]
LIndex_4 [0. 0. 1.]
LIndex_ee [1. 0. 0.]
LMiddle_Spread_Dummy [1. 0. 0.]
LMiddle_Proximal [0. 0. 1.]
LMiddle_3 [0. 0. 1.]
LMiddle_4 [0. 0. 1.]
LMiddle_ee [1. 0. 0.]
LRing_Spread [ 0.  0. -1.]
LRing_Proximal [0. 0. 1.]
LRing_3 [0. 0. 1.]
LRing_4 [0. 0. 1.]
LRing_ee [1. 0. 0.]
LPinky_Spread [ 0.  0. -1.]
LPinky_Proximal [0. 0. 1.]
LPinky_3 [0. 0. 1.]
LPinky_4 [0. 0. 1.]
LPinky_ee [1. 0. 0.]
```


ABB