# 代码架构

main.py：训练主程序

inference.py：测试主程序

train.py：用于训练一个 epoch 的模型，计算各种损失，并记录到Tensorboard中

test.py：用于测试一个 epoch 的模型，计算各种损失，并记录到Tensorboard中

dataset.py：功能涵盖了数据的读取、处理、归一化和为不同机械手器官（Yumi 机械手、Inspire Hand 机械手）准备不同数据集的过程。整个 **dataset.py** 文件的目的是为研究手语机器人模仿人体动作提供方便的数据处理工具。数据预处理，处理数据库的H5文件。

utils文件夹：用于存放自定义函数

models文件夹：

kinematics.py：实现 URDF 格式机器人前向运动学（Forward Kinematics）的 PyTorch 模块

loss.py：损失函数计算

model.py：使用 PyTorch Geometric 构建的神经网络模型，该模型基于图卷积网络（Graph Convolutional Network，GCN）的，用于处理图形数据，其中节点表示机器人手臂和手的相关信息，边表示它们之间的关联。整体模型的目标是通过学习有效的表示来推断机器人手臂和手的运动。

congfigs文件夹：

inference文件夹：

arm.yaml，hand.yaml，yumi.yaml

train文件夹：

arm.yaml，hand.yaml，yumi.yaml

YAML 配置文件用于指定数据集、模型、超参数、损失函数和其他相关设置，以便在深度学习模型的训练过程中使用

# 问题解决

一个共同的模型结构：使用 PyTorch Geometric 构建的图编码解码器的人机动作映射神经网络模型，该模型基于图卷积网络（Graph Convolutional Network，GCN）。

区别：该模型组成部分：基本结构+机器人各个部分模型网络

基本结构：SpatialBasicBlock: 用于实现图卷积（Graph Convolution）中的消息传递,用于在图结构上聚合邻接节点的信息。+Encoder+Decoder

机器人各个部分模型网络：ArmNet+HandNet+YumiNet

因为机器人只有手部不同，所以具体代码要修改HandNet部分

数据来源：浙大通过动作捕捉获得，我们通过华为手语翻译工具获得

训练（train）

浙大：源域(source\_path)：动作捕捉数据库的训练集

目标域（target\_path）: yumi-with-inspireHands

我们：源域(source\_path)：华为手语翻译工具数据库的训练集

目标域（target\_path）: yumi-with-schunkHands

测试（test）

浙大：源域(source\_path)：动作捕捉数据库的测试集

目标域（target\_path）: yumi-with-inspireHands

我们：源域(source\_path)：华为手语翻译工具数据库的测试集

目标域（target\_path）: yumi-with-schunkHands

因为数据来源和机械手不同，所以数据预处理代码不同：

浙大：dataset.py 我们：dataset\_HW.py

具体在模型训练和测试过程中

超参数（hyper-parameter）设置与模型训练效果

包括迭代次数（EPOCHS），每个小批量的大小BATCH\_SIZE），学习率（LEARNING\_RATE）

不同超参数会导致不一样的训练效果，定义多个损失函数，并创建了优化器，使用 Adam 优化算法，用TensorBoard显示训练效果