

基于单目视觉的低延迟4DoF机械臂实时模仿控制系统规划

I. 引言

A. 项目概述

本报告旨在规划一个项目，其核心目标是使一台uArm Swift Pro 4自由度(DoF)机械臂能够实时模仿人类用户的手臂运动。该系统的输入源于一个单一的、正对用户的标准RGB摄像头，该摄像头能够捕捉到用户完整的上半身及手臂姿态。此项目的关键技术挑战在于如何从缺乏直接深度信息的单目二维图像中，准确且低延迟地估计用户手部的三维空间位置，并基于此生成平滑、安全的机械臂末端执行器(effector)运动轨迹。用户已提出初步设想，涉及使用神经辐射场(NeRF)进行深度补全以及利用扩散模型(Diffusion Models)生成轨迹，本报告将对这些技术及其他相关前沿方法进行深入评估，以期规划出最优的技术路径。

B. 低延迟的关键性

在整个项目的设计与技术选型过程中，低延迟与实时性被确立为首要且不可妥协的约束条件。这里的“实时性”通常指系统需达到交互式的帧率，例如高于15-30 Hz，意味着整个感知-规划-控制循环的时间必须控制在33-66毫秒以内。对于旨在实现直观人机交互与模仿的应用而言，低延迟至关重要。任何显著的延迟都会破坏用户与机器人之间的流畅互动感，阻碍机械臂对用户动作的精确、即时跟随。高速响应系统，如机器人乒乓球¹，已充分证明了在反应性任务中最小化延迟的极端重要性。因此，本规划将始终以延迟作为核心评估指标，审视所有技术方案。

C. 报告结构

本报告将遵循以下结构展开：首先，深入探讨单目三维手部与手臂定位技术，重点评估其延迟性能；其次，分析用户提出的NeRF与扩散模型方案在实时应用中的可行性；接着，探索其他低延迟的定位与轨迹规划备选方案；然后，研究强化学习(RL)在优化控制回路中的应用潜力；随后，规划整体系统架构与数据流，确保各模块间的高效通信；最后，制定详细的实验方案与评估指标，将端到端延迟作为核心衡量标准。所有分析都将围绕实时性能这一核心约束进行。

II. 低延迟单目三维手部与手臂定位

A. 单目三维估计的挑战

从单一二维图像推断三维结构本质上是一个病态问题(ill-posed problem)，因为图像本身缺乏直接的深度信息⁴。这意味着多种不同的三维姿态可能投影到同一张二维图像上，造成固有的歧义性⁵。实际应用中还面临诸多困难，包括目标(手部)的自遮挡或被其他物体遮挡、快速运动导致图像模糊、光照条件变化以及手部关节的高度灵活性和复杂性⁶。这些因素都给准确、鲁棒的实时三维定位带来了巨大挑战。

B. 前沿技术概览

当前, 解决单目三维手部/手臂姿态估计问题存在多种技术路线, 各有优劣, 尤其在延迟性能上差异显著。

1. 直接三维坐标回归 (Direct 3D Coordinate Regression)

- 描述: 此类方法训练神经网络(通常是卷积神经网络CNN)直接从输入图像像素映射到三维关节点坐标⁴。
- 优点: 概念上相对简单, 如果网络结构轻量化, 推理速度可能较快。
- 缺点: 在没有显式约束或先验知识的情况下, 可能难以解决深度歧义问题, 并保证输出姿态的结构一致性(如骨骼长度、关节限制)。性能高度依赖于训练数据的质量和多样性。有研究指出, 直接回归三维坐标是一个高度非线性的映射, 学习难度较大¹¹。
- 参考文献示例: Tekin等人 (2017)⁴、Martinez等人 (2017)⁴ (尽管后者被指精度不足)的工作属于此类。ReJSHand¹¹ 等较新的工作也可能包含直接回归的元素, 尽管其最终输出可能是网格或更复杂的表示。

2. 基于模型的拟合方法 (Model-Fitting Approaches / Parametric Models)

- 描述: 利用参数化的人手模型, 如MANO⁷ 或人体模型SMPL/SMPL-X⁸, 这些模型提供了手部或身体形状与姿态的低维表示。网络的目标是回归这些模型参数, 而非直接回归关节点坐标。
- 优点: 融入了关于手部/身体结构的强先验知识, 能确保生成物理上合理的姿态和形状¹⁰。通过推断完整的模型状态, 能更好地处理部分遮挡情况。同时, 需要预测的参数数量减少¹⁰。
- 缺点: 可能涉及模型拟合或优化步骤, 计算成本相对较高¹⁹, 尽管也存在直接回归模型参数的方法¹⁰。对于模型训练数据中未充分覆盖的异常手型或姿态, 效果可能欠佳。推理速度因具体方法(回归 vs. 优化)差异巨大¹²。一些基于优化的方法还可能面临局部最优解的问题¹⁰。
- 参考文献示例: FrankMocap⁸ 使用SMPL-X模型。ReJSHand¹¹ 提及了MANO/SMPL, 但其自身似乎更侧重于直接预测顶点/关节点。一些工作明确使用了MANO损失函数⁷ 或SMPL+H模型¹⁵。

3. 深度推断 / 2.5D方法 (Depth Inference / 2.5D Methods)

- 描述: 首先估计手部区域的深度图或逐像素深度值, 然后利用估计的深度信息将二维关节点(可能独立检测或联合估计) 提升(lift)到三维空间⁴。MediaPipe Hands采用的即是一种2.5D方法¹³。
- 优点: 直接处理了深度歧义问题。可以利用成熟的二维关节点检测器。
- 缺点: 最终精度严重依赖于二维检测器和深度估计两者的准确性。从单目RGB图像估计深度本身就是一个难题。可能形成多阶段流水线, 若非端到端优化, 可能增加延迟。

- 参考文献示例: 文献⁴中的系统虽然使用深度传感器输入, 但也估计了精化的深度图。MediaPipe Hands¹³是典型的2.5D应用。

4. 混合方法 (Hybrid Methods, e.g., 2D Keypoints + Kinematics/Constraints)

- 描述: 结合鲁棒的二维关节检测(例如使用OpenPose¹⁹或MMPose/RTMPose²⁶)与运动学约束(骨骼长度、关节限制)或几何推理来推断三维姿态⁴。有时会涉及优化过程¹⁹或多视图三角化(本项目不适用)³⁰。
- 优点: 可以利用成熟的二维检测技术。显式地强制物理合理性。如果提升(lifting)步骤足够快, 计算效率可能很高。
- 缺点: 性能严重依赖于二维检测器的精度¹⁰。存在从二维到三维的误差传播问题。基于优化的提升方法可能较慢¹⁹。需要精确的相机标定信息¹⁹。
- 参考文献示例: 文献¹⁹采用了OpenPose结合优化的方法。VideoPose3D³⁰利用视频中的时序信息将二维姿态提升至三维。RTMW3D²⁸进行三维全身姿态估计, 可能使用了提升或坐标分类方法。

C. 性能基准分析 (聚焦延迟与精度)

对报告中提及的具备实时或接近实时性能的特定模型进行分析:

- **MediaPipe Hands:** 延迟极低, Pixel 6上GPU约12ms²⁴, Snapdragon X Elite上检测器+关键点模型合计约1-2ms²¹。专为移动和边缘设备优化²¹。输出21个关键点的图像坐标和世界坐标²⁴。局限性: 输入分辨率固定 (256x256)²¹, 在多手、复杂手势或低光照场景下性能可能下降²¹。与触摸屏对比的精度测试显示存在微小但统计显著的差异²³。采用手掌检测+关键点跟踪的流水线²⁴。
- **RTMPose / RTMW / RTMW3D:** FPS很高。RTMPose-m在CPU上超过90 FPS, GPU上超过430 FPS²⁷。RTMPose-s在骁龙865上超过70 FPS²⁷。RTMPose-l(全身)在CPU上超过130 FPS²⁷。RTMW-l/x在COCO-WholeBody上达到SOTA水平²⁸。RTMW特别关注提升手部关键点精度²⁸。RTMW3D探索使用坐标分类进行三维全身估计²⁸, 在H3WB数据集上达到57mm MPJPE³³。基于MMPose框架²⁶。
- **ReJSHand:** 报告在NVIDIA 2080Ti上达到72 FPS¹¹。精度良好 (PA-MPJPE 6.3mm)¹²。使用FastViT作为骨干网络¹²。明确为实时性能设计, 采用轻量级架构¹²。
- **FrankMocap:** 报告在RTX 2080上使用快速集成策略进行全身(手+身体)动作捕捉时约9.5 FPS¹⁸。采用模块化设计: 独立运行身体/手/面部回归器, 然后集成⁹。使用SMPL-X模型⁸。速度并非最快, 但如果需要全身背景信息, 它能提供。
- **Panteleris等人 (2017):** 报告在GTX 1070上达到18 FPS¹⁹。使用OpenPose(较慢步骤约30ms)+优化¹⁹。输出完整三维姿态(不像某些方法会归一化尺度³⁶)。
- **Hesse等人 (2019):** 报告使用Kinect V2深度输入时, 单人8个上半身关节估计达到40 FPS⁴。使用Slim Stacked Hourglass架构, 在单GPU上达到58 FPS⁴。注意: 该方法使用深度输入, 不直接可比, 但揭示了相关骨干架构的速度潜力。

为了更清晰地比较这些方法的性能, 下表汇总了关键信息:

表1: 实时三维手部/手臂姿态估计方法比较分析

方法名称	核心方法	输入	输出	报告延迟 /FPS	测试硬件	精度 (指标+值)	主要特点 /局限性
MediaPipe Hands	2.5D (手掌检测+关键点跟踪)	RGB	21个3D世界坐标点	~12ms (GPU, Pixel 6) ²⁴ , ~1-2ms (Snapdragon X Elite) ²¹	Pixel 6, Snapdragon X Elite	未报告标准指标, 但有对比测试 ²³	极低延迟, 移动端优化, 输出世界坐标; 固定分辨率, 复杂场景可能受限 ²¹
RTMPose/RTMW/RTMW3D	2D+提升 / 3D坐标分类	RGB	2D/3D全身关键点	RTMPose-m: >90 FPS (CPU), >430 FPS (GPU) ²⁷ ; RTMW3D: 57mm MPJPE (H3WB) ³³	Intel i7, GTX 1660 Ti, Snapdragon 865	手部 AP/AR见 ²⁷ ; RTMW3D: 57mm MPJPE ³³	极高FPS, 提供全身(含手臂)信息, RTMW3D直接输出3D; 需验证3D版本延迟, 手部精度可能低于专用模型
ReJSHand	轻量级网络直接预测 (2D+3D关键点+网格)	RGB	3D关键点+网格	72 FPS ¹²	NVIDIA 2080Ti ¹³	PA-MPJPE 6.3mm ¹²	速度与精度平衡良好, 输出网格; 需要较强GPU, 实现可用性待确认
FrankMo	参数模型回归	RGB	SMPL-X	~9.5 FPS	RTX	基准测试性能	提供完整身体模型

cap	(SMPL-X)		参数	(全身) ¹⁸	2080 ¹⁸	SOTA ¹⁸	, 模块化设计; 相对较慢, 主要针对全身
Panteleris et al. (2017)	2D (OpenPose) + 优化拟合	RGB	21个3D关节点	18 FPS ¹⁹	Intel i7 + GTX 1070 ¹⁹	未明确报告标准指标	输出完整3D姿态; OpenPose步骤较慢 (~30ms) ¹⁹ , 优化步骤可能增加延迟

此表集中呈现了分散在各文献中的关键性能数据⁴, 便于基于核心约束(延迟)和其他因素(精度、硬件要求)对不同技术路径进行直接比较。它清晰地揭示了速度、精度、模型复杂性(手部专用 vs. 全身)以及硬件需求之间的权衡, 为后续的技术选型提供了必要的依据。

D. NeRF用于实时深度补全的可行性分析

用户初步设想使用NeRF通过双目数据训练模型, 然后在运行时利用单目输入补全深度信息, 进而实现三维定位。对此方案的可行性进行评估:

- **动态场景NeRF研究现状:** 确实存在针对动态场景的NeRF变种, 如D-NeRF、HyperNeRF、Nerfies、NSFF等, 它们通常通过引入时间参数或学习形变场来处理运动³⁷。然而, 这些方法增加了计算复杂度, 并且其主要目标通常是生成动态场景的新视角合成图像, 而非实时地为控制任务估计深度³⁷。
- **实时NeRF/NVS技术:** 诸如模型烘焙(baking)³⁹、Instant-NGP⁴⁰、高斯泼溅(Gaussian Splatting, GS)⁴⁰、K-Planes⁴⁰、MERF³⁹、Quark (LDM)⁴¹等技术能够实现实时(>30 FPS)的渲染或新视角合成³⁹。但这些方法通常需要多视角输入进行训练或重建, 并且优化目标是渲染质量。它们并未直接解决用户所需的从单一视角实时推断动态目标的深度信息以用于控制的问题。
- **单视角NeRF面临的挑战:** 尽管NeRF可以从稀疏视角⁴⁰甚至单一视角(通常带有强先验)重建静态物体, 但将其应用于动态人体运动的实时深度推断则面临巨大障碍:
 - **训练数据:** 需要大量的(可能是配对的RGB-D或多视角)数据来学习从单一视角到各种人体姿态深度的映射。
 - **推理延迟:** NeRF的核心渲染过程(光线步进、MLP查询)本身计算量大且速度慢³⁹。即使是优化后的方法, 其目标也是渲染, 不一定能满足从单一、可能分布外的视角快速生成深度图的需求。对于本项目要求的<50ms延迟, 现有NeRF架构用于此特定任务似乎极难实现。EventNeRF⁴²展示了稠密重建的潜力, 但它依赖于事件

相机而非标准RGB相机。

- **结论:** 基于当前技术水平和项目的严格延迟约束, 采用NeRF进行实时单目人体手/臂深度补全的方案面临极高挑战, 很可能速度过慢。其计算成本和实现复杂性远超潜在收益, 相比之下, 专用的姿态估计方法更为现实。NeRF技术, 即使是为实时渲染优化的版本, 其核心计算模式(查询体积表示)与姿态估计中常用的前馈CNN或轻量级跟踪器相比, 在速度上存在根本性劣势¹³。

E. 其他低延迟定位策略评估

基于表1的分析, 重点评估最有希望满足低延迟要求的候选方案:

- **MediaPipe Hands:** 由于其极低的延迟²¹和直接输出3D世界坐标的能力²⁴, 是一个强有力的竞争者。关键在于: 其精度和鲁棒性是否满足用户需求, 特别是在考虑其潜在局限性²¹的情况下? 它是否提供完整手臂姿态, 还是仅限于手部?(很可能仅手部²⁴)。
- **RTMPose/RTMW/RTMW3D:** 速度非常快, 尤其在GPU上²⁷。能提供包括手臂在内的全身信息²⁸。RTMW3D提供直接的3D估计能力²⁸。关键问题: 延迟是否足够低, 特别是3D版本? 手部关键点精度是否足够²⁷? 坐标分类方法提供的深度是否稳定?
- **ReJSHand:** 在速度(72 FPS¹²)和精度(6.3mm PA-MPJPE¹²)之间取得了良好平衡。输出网格, 意味着包含3D坐标。关键问题: 基准测试需要较好的GPU(2080Ti¹³)。其实现是否可用且易于集成?
- **混合方法(2D + 提升):** 如果能将一个极快的2D检测器(如RTMPose 2D)与一个快速的提升机制(如简单的运动学求解器或轻量级回归网络)相结合, 则具有可行性。VideoPose3D³⁰展示了潜力, 但利用了时序信息。关键问题: 提升步骤能否在保证足够精度的情况下实现足够低的延迟?

F. 3D定位技术推荐

选择技术方案时, 必须首要考虑延迟, 其次是精度、鲁棒性、实现复杂度和输出需求(仅手部 vs. 手臂)。

- **初步推荐:**
 - 如果项目仅需手部跟踪, 且能接受MediaPipe Hands的潜在局限性, 那么**MediaPipe Hands**因其无与伦比的速度而成为首选。
 - 如果需要完整手臂跟踪, 或者对鲁棒性/精度有更高要求, **RTMW3D**(或在GPU资源允许的情况下考虑ReJSHand)是更合适的选择, 但前提是必须通过实实验证其在目标硬件上的真实延迟。
 - 如果直接的3D方法被证明不足, ****混合方法(快速2D检测+快速提升)****可作为备选方案。
- **强调:** 任何推荐都基于现有文献报告, 最终选择必须经过在用户目标硬件上的实证测试来确认。报告的基准测试结果⁴受到硬件和具体实现细节的极大影响。真实世界的性能必须通过实验来验证。同时, 静态数据集上的精度指标(如MPJPE)并不能完全反

映动态跟踪的稳定性或对遮挡/模糊的鲁棒性¹⁰，这些对于实时模仿至关重要。

G. 本章核心要点总结

本章节的分析揭示了几个关键点。首先，手部专用模型(如MediaPipe, ReJSHand)和通用全身模型(如RTMW)之间存在专业化与通用性的权衡。手部模型通常能实现极低延迟，因为它们处理的问题空间更小、更受限，允许使用高度优化的轻量级网络¹³。而全身模型需要处理更多样化的身体部位和姿态³²，可能导致网络更大或在手部细节的精度/速度上有所妥协，除非像RTMW那样特别关注手部²⁸。考虑到本项目需要手臂跟踪，全身模型可能更合适，前提是其延迟达标。

其次，对于实时控制输入而言，**NeRF**的应用是不切实际的。当前的NeRF研究，即使是针对动态场景和实时渲染，也与本项目所需的快速、单视角深度估计以用于控制回路的目标不太契合。实时NeRF主要关注从多视角数据或预处理模型进行视图合成/渲染³⁹，而动态NeRF增加了复杂性³⁷。单视角动态人体深度估计并非标准的、经过优化的NeRF任务。NeRF的基本计算单元(MLP查询、体积渲染)在速度上根本无法与专用姿态估计器中的前馈CNN或轻量级跟踪器相比¹³。因此，延迟要求很可能排除了NeRF在此特定功能上的应用。

最后，基准测试存在局限性。报告的FPS/延迟数据⁴高度依赖于测试硬件和具体实现。用户必须在自己的目标系统上进行实证测试。此外，诸如MPJPE之类的精度指标通常在静态数据集上计算¹⁰，未必能反映快速运动或遮挡情况下的性能⁶，而后者对实时模仿至关重要。动态指标(如抖动¹⁰)和定性评估同样重要。

III. 机器人控制的实时轨迹规划

A. 问题定义

轨迹规划模块的任务是将上游姿态估计模块输出的用户手/臂三维关键点(例如，作为机械臂末端执行器目标的腕部位置)的时序数据 $(x_h, y_h, z_h)t$ 转换为uArm Swift Pro API可接受的目标末端执行器坐标序列 $(x_r, y_r, z_r)t$ 。此过程必须满足以下约束：

1. 实时性: 轨迹生成延迟必须极低，以匹配整个系统的实时要求。
2. 平滑性: 生成的轨迹应足够平滑，以避免机械臂产生急动或振动。
3. 约束性: 轨迹必须在机械臂的4DoF运动学限制内(工作空间范围、关节角度限制、奇异点规避)，并可能需要考虑速度和加速度限制。

B. 模仿学习 (IL) 评估

模仿学习通过学习专家演示来获取策略，在机器人领域有广泛应用⁴³。

1. 行为克隆 (Behavioral Cloning, BC)

- 描述: 最简单的IL形式，直接学习从观察(估计的手部姿态/轨迹)到动作(机器人坐标/

指令)的映射。

- 优点: 实现简单, 如果策略网络规模小, 推理速度可能非常快。
- 缺点: 容易受到协变量偏移(covariate shift)的影响, 即训练时的专家数据分布与实际遇到的状态分布不匹配, 导致在分布外输入上可能出现灾难性失败。需要具有代表性的专家演示数据。除非特别设计, 否则不一定能保证生成平滑轨迹。

2. 高级IL方法

- 描述: 旨在改进简单BC的鲁棒性和泛化能力的方法, 如DAgger、GAIL, 以及利用大型数据集或结构化表示的更新方法。
- **ATM (Any-point Trajectory Modeling):** 通过预训练一个轨迹模型来预测视频帧内任意点的未来轨迹⁴⁵。这些预测轨迹随后用于指导视觉运动策略的学习。据称相比基线有显著改进, 且样本效率高⁴⁵。采用基于粒子的建模, 可能比视频预测更高效⁴⁵。延迟考量: 在每个时间步预测多个点的轨迹所需的计算成本是否满足实时要求, 需要评估⁴⁵。
- **DTP (Diffusion Trajectory-guided Policy):** 使用扩散模型生成二维轨迹以指导策略学习⁴⁴。旨在减少长时程任务中的误差累积。延迟考量: 依赖扩散模型, 引发延迟问题, 除非使用加速技术(可能仅用于初始引导, 而非步进控制)。
- **ViT + LSTM:** 在手术机器人IL中使用, 处理视频帧和序列数据⁴³。基于当前帧和先前姿态预测未来姿态。延迟考量: ViT和LSTM的推理计算量可能较大。

3. IL的实时性考量

- 策略推理速度是关键。简单的MLP策略很快, 但更复杂的架构(Transformer, LSTM, 轨迹预测器)会增加延迟⁵⁰。
- IL方法通常关注任务完成成功率⁴³, 不一定针对本项目所需的精细、低延迟轨迹跟随进行优化。其适用性需要仔细评估。对于本项目的连续跟踪任务, 相比直接生成或映射方法, 许多高级IL技术(如预测未来视频/轨迹⁴⁴)可能过于复杂或引入不必要的延迟。

C. 扩散模型用于轨迹生成的评估

1. 用户提议的方法: 使用扩散模型, 以估计的手部姿态为条件, 生成机器人轨迹, 并结合物理约束。

2. 推理延迟的关键分析:

- **标准扩散模型 (DDPM/DDIM):** 需要进行多次(数十到数千次)迭代去噪步骤才能采样生成结果⁵¹。这导致了高延迟(例如, 处理点云输入的DP3报告延迟约178ms⁵²), 使其不适用于实时控制回路⁵¹。这个固有的多步采样过程是标准扩散模型的一个主要瓶颈, 使其与实时控制需求不兼容。
- **加速扩散模型 (Consistency Models):** 一致性训练/蒸馏(Consistency Training/Distillation)等技术通过强制概率流ODE轨迹上的自治性, 实现了从噪声到清

晰数据在单步内的映射⁵¹。这极大地降低了推理延迟，使其进入毫秒级范围，从而使得在实时机器人应用中使用生成式轨迹模型成为可能。

- **FRMD (Fast Robot Motion Diffusion):** 明确地将一致性模型与运动基元 (Movement Primitives, MPs), 特别是概率动态运动基元 (ProDMPs) 相结合⁵¹。实现了单步推理⁵¹。报告的平均延迟约为17.2ms⁵¹。声称比DP和教师模型快7-10倍⁵¹。通过预测MP参数而非原始路径点, 旨在生成更平滑的轨迹⁵¹。这种结合不仅解决了延迟问题, 还利用MPs的结构特性改善了轨迹质量。
- **ManiCM:** 同样使用一致性蒸馏技术, 实现从点云输入的单步推理⁵²。报告比DP3快10倍⁵²。侧重于直接预测动作⁵²。
- 其他快速采样技术: 基于分数的模型⁵⁷也有相应的快速采样技术⁵⁷。
- **结论:** 标准扩散模型因其迭代采样过程带来的高延迟, 不适用于本项目的实时要求。然而, 基于一致性模型的技术 (如FRMD, ManiCM) 通过实现单步推理, 显著降低了延迟, 使得扩散模型在理论上可以满足实时轨迹生成的需求。

3. 结合约束: 如何在低延迟框架内确保生成的轨迹遵守4DoF机器人的物理约束?

- **隐式学习:** 期望模型从演示数据中自动学习约束。这种方法通常不可靠。
- **训练时显式约束:** 将约束纳入扩散模型的训练目标或条件信息中。这种方法与一致性模型结合的可行性有待研究。
- **后处理校正:** 生成轨迹后, 检查并修正 (裁剪、投影) 以满足约束。这会增加额外的延迟。FRMD使用MPs可能简化约束处理, 因为MPs通常具有内建的平滑性。

表2: 扩散模型轨迹生成技术延迟比较

方法类别	具体方法	推理步骤	报告延迟/ 速度提升	测试硬件	约束处理 方式	主要特点
标准扩散 (DDPM/D DIM)	Diffusion Policy (DP)	多步	较高 (e.g., ~178ms for DP3 ⁵²)	(未指定)	隐式/后处 理	基线方法, 延迟高
一致性模型	FRMD	单步	~17.2ms ⁵¹ ; 比DP快7倍 , 比MPD快 10倍 ⁵¹	(未指定)	MP参数预 测+?	结合MPs, 单步推理, 关注平滑性 ⁵¹
一致性模型	ManiCM	单步	比DP3快 10倍 ⁵²	(未指定)	?	单步推理, 直接预测 动作 ⁵²

此表直接对比了标准扩散模型与基于一致性模型的加速方法在关键延迟指标上的表现⁵¹。它突显了一致性模型作为使能技术, 通过实现单步推理将延迟降低了数量级(7-10倍)⁵¹, 从而使扩散模型在实时机器人应用中变得可行。同时, 它也比较了不同的快速扩散方法(如FRMD与ManiCM)的报告速度和特性(如FRMD使用MPs以提高平滑度⁵¹)。这些数据为评估用户最初提出的扩散模型方案提供了依据: 只有采用加速技术, 该方案才具有实际可行性。

D. 其他实时规划器探索

- **1. 直接映射与平滑 (Direct Mapping with Smoothing):** 最简单的方法。将估计的手部位置 (xh,yh,zh) 直接(可能经过缩放/偏移)映射到机器人目标 (xr,yr,zr)。应用滤波器(如卡尔曼滤波器⁶, 移动平均)平滑目标轨迹。最后检查约束。
 - 优点: 延迟极低。
 - 缺点: 可能违反运动学约束; 平滑处理会引入滞后; 生成的轨迹可能对机器人而言并非动态可行。
- **2. 运动学求解器 / 逆运动学 (IK):** 给定目标 (xh,yh,zh), 计算4DoF机械臂所需的关节角度。检查可达性和关节限制。
 - 优点: 保证运动学可行性。是标准的机器人技术。
 - 缺点: 对于冗余或接近奇异点的构型, IK求解可能缓慢或不稳定。本身不保证轨迹平滑性或动态可行性。需要在控制频率下运行。
- **3. 基于优化的方法 (若足够快):** 模型预测控制(MPC)或轨迹优化等技术可以生成平滑且满足约束的轨迹。MPCNet⁵⁸ 展示了由MPC指导的策略学习。
 - 优点: 能显式处理约束和优化目标(平滑度、精度)。
 - 缺点: 通常计算成本高, 除非经过高度优化或简化, 否则往往不适用于高频实时控制⁵⁰。推理时间可以在毫秒级, 但严重依赖于问题的复杂性⁵⁰。

E. 轨迹规划推荐

选择时优先考虑低延迟和平滑性。

- **初步推荐:** 加速扩散模型(特别是FRMD⁵¹) 似乎是最有前途的现代方法。它提供了生成模型的灵活性, 同时通过单步推理实现了低延迟(约17ms), 并可能通过运动基元获得更好的平滑性。这直接回应了用户的兴趣, 同时在技术上是可行的。
- **备选方案:** 如果扩散模型实现或调优过于复杂, 可以采用直接映射结合鲁棒平滑和约束检查, 或一个高度优化的快速IK求解器, 但需要接受在轨迹质量或约束处理上可能存在的局限性。相比之下, 模仿学习方法对于本项目所需的精细、连续跟踪任务, 似乎不如直接生成或映射方法那么匹配。

F. 本章核心要点总结

本节分析强调了几个关键方面。首先, 标准扩散模型是延迟陷阱。用户最初的想法需要重大修改; 标准扩散模型的多步迭代采样过程⁵¹ 导致的高延迟(常达数百毫秒⁵²)与实时控制所

需的低延迟(通常<50ms)要求不兼容。因此,若不采用加速技术,该方案对本应用而言存在根本缺陷。

其次,一致性模型是关键的使用技术。单步扩散(一致性模型)的最新进展对于使生成式轨迹模型在实时机器人领域变得可行至关重要。通过将迭代采样压缩为单步⁵¹,一致性模型将推理延迟降低了数量级(例如10倍⁵¹),使其进入控制所需的毫秒范围⁵¹,从而克服了在机器人中使用扩散模型的主要瓶颈。

第三,运动基元(MPs)有助于提升平滑性和结构性。将快速扩散与MPs结合(如FRMD⁵¹)不仅解决了延迟问题,还针对性地处理了扩散模型常见的输出抖动/不平滑问题⁵³。扩散模型直接操作原始路径点时可能难以保证时间一致性⁵³,而MPs(如DMPs/ProMPs)提供了一种低维、内禀平滑的轨迹表示⁵¹。通过让扩散模型预测MP参数而非路径点,可以利用这种结构性先验,潜在地生成更平滑、动态更合理的运动⁵¹。这种协同组合同时解决了扩散模型的两大局限。

IV. 低延迟强化学习(RL)集成

A. RL在系统中的角色

明确RL在此系统中的定位:并非用于从像素到动作的端到端控制(这对于实时系统过于复杂和缓慢),而是作为轨迹规划模块(IL、扩散模型或其他)输出的轨迹的精炼器(refiner)。其目的是对规划器生成的初始轨迹进行微调和优化。

潜在的好处包括:提高最终的跟踪精度,增强运动的平滑性,适应姿态估计或系统模型中的微小误差,更鲁棒地强制执行约束,以及可能提高系统的响应速度。

B. 合适RL算法的选择(关注推理速度)

1. 候选算法:

- **PPO (Proximal Policy Optimization):** 属于在线策略(On-policy)算法,以其训练稳定性和在多种任务上的良好性能而闻名⁵⁹。在机器人领域有广泛应用⁶¹。由于其在线策略特性,PPO在训练时计算量可能较大,但学习到的策略推理(执行)通常很快(神经网络的前向传播)⁶³。LFPPPO变种声称能改善探索效率⁶⁰。已有针对PPO训练的硬件加速研究⁶³。
- **SAC (Soft Actor-Critic):** 属于离线策略(Off-policy)算法,通常比PPO具有更高的样本效率⁵⁹。同样被广泛使用,并在连续控制任务中表现出色⁶²。其推理过程(执行Actor神经网络的前向传播)通常也很快。已有研究探索使用低精度训练SAC以减少计算和内存需求⁶⁴。

2. 延迟考量:

- **策略推理时间:** 对于PPO和SAC,推理时间主要取决于策略网络的大小和架构。中小型

MLP(多层感知器)通常可以在毫秒级完成推理⁵⁰。只要网络规模保持合理,这部分不应成为主要的延迟瓶颈。

- 训练时间/样本效率: 虽然不是实时约束,但SAC的离线策略特性可能使其能更有效地利用收集到的交互数据进行训练,相比PPO可能需要更少的实际交互时间。

3. 推荐: PPO或SAC均可能适用。考虑到在实际机器人应用中样本效率通常很重要, **SAC**可能略有优势。最终选择可能取决于团队对特定算法的熟悉程度和调优经验。关键在于保持策略网络相对较小,以确保最低的推理延迟。对于PPO/SAC这类标准RL算法,只要策略网络不过于庞大,其前向传播通常非常快(毫秒级),远低于复杂视觉模型或迭代式规划器的计算成本⁵⁰。因此,选择的重点应更多地放在训练特性(稳定性、样本效率)而非推理速度上。

C. 奖励函数设计策略

设计奖励函数 R 以引导RL智能体学习期望的行为。需要仔细平衡以下组成部分:

- 跟踪精度: 惩罚机器人末端执行器实际位置与规划器给出的目标位置之间的偏差。例如: $R_{acc} = - // posrobot - postarget // 2$ 。
- 平滑性: 惩罚过高的关节速度、加速度或加加速度(jerk),以鼓励平滑运动。例如: $R_{smooth} = - // accelrobot // 2$ 或 $R_{smooth} = - // jerkrobot // 2$ 。
- 响应性/低延迟: 可以设计奖励项来鼓励快速响应。例如,奖励快速达到目标速度,或惩罚与规划器即时目标点的偏差,或者直接惩罚延迟。例如: $R_{resp} = -delay$ 或 $R_{resp} = +velocityachieved$ 。为了使RL智能体的目标与系统的低延迟要求保持一致,奖励函数必须明确地激励响应速度并惩罚滞后。如果奖励仅关注精度和平滑度,智能体可能会学到一个准确但滞后的策略。
- 约束遵守: 对违反关节限制、速度限制或工作空间边界的行为施加惩罚。例如: $R_{constraint} = -penalty \text{ if violated}$ 。
- 动作正则化: 惩罚相邻时间步之间动作指令的大幅变化,有助于平滑控制信号。例如: $R_{action} = - // actiont - actiont-1 // 2$ 。

需要强调的是,这些奖励项的权重需要通过实验仔细调整,以达到最佳的整体性能。

D. 本章核心要点总结

本节分析得出以下结论。首先, **RL**应作为精炼器而非主控制器。考虑到延迟约束和问题复杂度,采用RL来微调规划好的轨迹比从视觉进行端到端学习更为可行。端到端的图像输入RL不仅样本效率低下,且通常需要复杂的网络结构,很可能导致高推理延迟。将感知、规划和控制分离是标准的机器人系统设计方法¹。使用RL调整规划轨迹能结合规划(提供良好初始解)和RL(局部优化、适应性)的优势,使问题更易处理且可能更快。

其次,策略推理速度通常不是瓶颈。对于如PPO/SAC的标准RL算法,只要策略网络规模适中(例如MLP⁶³),其前向传播通常非常快(毫秒级)⁵⁰。主要的挑战在于训练效率和稳定

性。因此，在PPO和SAC之间的选择应更多地基于训练特性，而非推理速度。

最后，奖励设计对于实现低延迟行为至关重要。奖励函数必须明确地激励快速响应并惩罚滞后⁵⁹，才能将RL智能体的优化目标与系统的低延迟要求对齐。仅仅奖励精度和平滑度可能导致智能体学到滞后的策略。需要加入与速度、延迟或快速收敛到目标相关的项⁶⁰，以直接引导智能体学习低延迟行为。

V. 系统架构与数据流

A. 建议的系统框图

为了清晰地展示信息流动和处理步骤，建议采用以下模块化系统架构：

Code snippet

graph LR

```
A[摄像头] --> B(图像采集与预处理);
B -- 处理后图像 --> C{3D 手部/手臂姿态估计器 (例如 RTMW3D)};
C -- 估计的3D手部轨迹 --> D;
D -- 规划的机器人轨迹 --> E{RL 策略 (精炼)};
E -- 最终机器人执行器命令 (x,y,z) --> F;
F --> G((机械臂));
```

```
style C fill:#f9f,stroke:#333,stroke-width:2px
style D fill:#ccf,stroke:#333,stroke-width:2px
style E fill:#9cf,stroke:#333,stroke-width:2px
```

- 流程说明: 摄像头捕捉图像，经过预处理(如缩放、归一化)后送入姿态估计器。估计器输出用户手/臂的三维关键点序列。轨迹规划器根据这些输入生成期望的机器人末端执行器轨迹。RL策略接收规划轨迹和可能的当前状态信息，输出最终的、经过精炼的(x,y,z)坐标指令。该指令通过uArm API发送给机械臂执行。
- 反馈: 可选地，如果uArm API能提供实时的机器人状态(如关节角度或末端执行器位置)，这些信息可以反馈给RL策略或规划器，用于闭环控制或状态估计。

这种模块化设计¹具有重要优势。它允许独立开发、测试和优化每个组件。例如，如果发现姿态估计器太慢或不够准确，可以在不重新设计整个系统的情况下，替换为满足相同接口定义的其他模型。这种灵活性对于迭代优化和解决特定性能瓶颈至关重要。

B. 模块接口定义

为确保模块间的顺畅集成, 需要明确定义接口:

- **图像采集:** 输出: RGB图像帧(例如, 分辨率640x480, 帧率30fps)。指定数据格式(如OpenCV Mat对象)。
- **预处理:** 输入: 原始图像帧。输出: 处理后的图像(例如, 缩放到姿态估计器所需的输入尺寸²¹, 进行归一化)。函数签名示例: `preprocess(raw_image) -> processed_image`。
- **姿态估计器:** 输入: 处理后的图像。输出: 三维关键点坐标列表/数组(例如, 手腕、肘部、肩部), 坐标系需明确(相机坐标系或世界坐标系)。函数签名示例: `estimate_pose(processed_image) -> List[Point3D]`。
- **轨迹规划器:** 输入: 最近一段时间内的估计手部关键点序列(例如, 作为目标的执行器位置)。输出: 规划出的未来一小段时间内的机器人末端执行器三维坐标序列。函数签名示例: `plan_trajectory(hand_trajectory_history) -> List[Point3D]`。
- **RL策略:** 输入: 当前状态(包括估计的手部姿态、规划轨迹段, 可能还有当前机器人状态)。输出: 精炼后的机器人末端执行器指令(可以是绝对位置, 也可以是相对于规划轨迹的调整量)。函数签名示例: `refine_action(state) -> Point3D_command`。
- **uArm API接口:** 输入: 目标 (x,y,z) 坐标。函数签名示例: `set_position(x, y, z)`。需要了解此API是阻塞式还是非阻塞式, 以及其通信延迟特性。

C. 低延迟通信策略

模块间的通信效率对整体系统延迟有显著影响, 必须仔细设计:

- **最小化数据拷贝:** 如果模块在不同进程中运行, 应使用共享内存或高效的序列化库(如Protobufs, 已在类似高速系统中使用¹⁾)来传递数据, 避免不必要的内存复制开销。
- **异步处理/流水线:** 探索姿态估计、规划和控制步骤是否可以并行或流水线执行以隐藏延迟。例如, 在执行第 t 帧的命令时, 开始处理第 t+1 帧的姿态估计和规划。这需要精密的同步机制。
- **优化数据流:** 确保从图像采集到机器人指令生成的关键路径具有最小的通信开销。
- **直接内存访问 (DMA):** 如果硬件条件允许(例如, 摄像头数据直接传输到GPU内存), 利用DMA可以绕过CPU瓶颈, 进一步降低延迟。
- **与uArm API的通信:** 了解与uArm机械臂通信的具体协议(很可能是USB串行通信)及其固有延迟。高效地发送指令。对于实时跟随任务, 批量发送指令可能不适用, 需要高频率地发送单个目标点。

模块间的数据传输时间(例如, 图像、姿态、轨迹在软件模块或进程间传递)如果管理不当, 会显著增加总延迟。大型数据(如高分辨率图像)传输更慢。忽略通信延迟会削弱在单个模块内所做的优化工作。

此外, 实现超低延迟通常需要硬件与软件的协同设计。需要考虑运行平台(CPU、GPU、嵌入式SoC等⁴⁾), 并针对性地优化软件(例如, 使用GPU加速库²¹⁾)。数据在CPU和GPU之间的

传输也可能成为瓶颈。优化整个流水线, 包括数据如何流入和流出加速器, 对于最小化端到端延迟至关重要¹。

VI. 实验设计与评估协议

A. 数据收集计划 (如有必要)

- 若采用**NeRF** (不推荐): 需要同步的多视角视频或RGB-D数据, 覆盖多样的手/臂运动。
- 若采用**IL/扩散模型规划器**: 需要专家演示数据。记录用户手/臂的三维姿态序列(使用选定的估计器)以及对应的期望机器人执行器轨迹(可通过遥操作或已知良好控制器生成)。确保数据覆盖模仿任务相关的广泛运动范围。
- 若采用**RL**: 数据在与(模拟或真实)环境的在线交互过程中收集。

B. 仿真环境 (强烈推荐)

- 目的: 进行安全的初步测试、算法调优、RL智能体训练, 以及在不损坏硬件的情况下评估不同组件。对于高速系统尤其重要¹。在硬件上直接测试高速或基于学习的控制既有风险又效率低下。仿真允许进行数百万次的训练迭代(对RL而言)或参数扫描, 这在真实硬件上是不可行的⁶⁶。
- 设置: 使用物理模拟器(如PyBullet, MuJoCo⁶², Isaac Gym)。建立uArm Swift Pro的精确模型(运动学、尽可能包含动力学、工作空间限制)。模拟摄像头视图和姿态估计过程(可以加入噪声/延迟模型)。一个好的模拟器可以在真实部署前测试延迟建模和噪声鲁棒性等问题²。

C. 真实世界测试流程

- 1. 组件验证: 在目标硬件上独立测试每个模块。测量姿态估计器、规划器和RL策略推理的延迟和精度。
- 2. 集成系统测试 (开环): 运行从摄像头到生成机器人指令的完整流水线, 但暂时不将指令发送给机器人。测量从图像采集到指令生成的端到端延迟。记录规划出的轨迹。
- 3. 集成系统测试 (闭环 - 谨慎进行):
 - 从缓慢、平滑的人类动作开始。
 - 验证机器人在工作空间内是否正确、安全地跟随。
 - 逐步增加人类动作的速度和复杂度。
 - 录制人类和机器人的同步视频, 记录所有中间数据(姿态、规划轨迹、指令)及时间戳。
- 4. 定量评估: 执行特定的、可重复的测试动作(如画图形、点到点移动), 并使用下述关键性能指标进行测量。评估不能仅限于静态姿态或简单动作, 必须包含动态场景, 因为真实的人体运动涉及变速、变加速和可能的自遮挡⁶, 这些会影响系统性能。

D. 关键性能指标 (KPIs)

- **1. 延迟指标 (首要关注):**
 - **端到端系统延迟 (ms):** 从摄像头图像时间戳到uArm API指令执行确认(如果可能)的总时间。这是衡量实时性能和用户体验的最终指标。需要精确记录各阶段时间戳。分析平均值、中位数、最大值和分布。研究延迟对稳定性的影响²。虽然组件延迟有助于调试,但决定实时性能的是从感知到动作的总时间。
 - **组件延迟 (ms):** 分别测量预处理、姿态估计、轨迹规划、RL推理、API通信的执行时间。用于识别性能瓶颈。
- **2. 精度指标:**
 - **姿态估计精度:** 如果有地面真值(例如,使用动作捕捉系统进行验证),计算手/臂关节的平均关节位置误差(MPJPE)。
 - **轨迹跟踪误差 (mm):** 目标人手位置(映射到机器人工作空间)与机器人实际末端执行器位置(需要获取机器人状态)之间的欧氏距离。计算轨迹上的均方根误差(RMSE)。
- **3. 平滑性指标:**
 - 分析记录的机器人关节速度、加速度和加加速度(jerk)。高值表示运动不平滑或有抖动⁵³。可计算光谱弧长(SPARC)或对数无量纲加加速度(LDLJ)等指标。
- **4. 任务特定指标:**
 - 特定手势或动作模仿的成功率(如果适用)。
 - 模仿质量的定性评估(流畅度、响应性)。
- **5. 计算负载:**
 - CPU利用率 (%)。
 - GPU利用率 (%), GPU显存使用量 (MB)。
 - 系统整体资源消耗。

E. 本章核心要点总结

实验设计和评估是确保系统满足要求的关键环节。端到端延迟是最终衡量标准,因为它直接反映了系统的实时交互能力,累积了从感知到执行的所有环节的延迟。仿真环境对于安全高效的开发至关重要,它允许在部署到真实硬件之前进行广泛的测试、训练和参数调整,避免潜在的硬件损坏并加速开发周期。最后,基准测试必须包含动态场景,因为系统的性能(如姿态估计的准确性、控制的稳定性)在快速、复杂的运动中可能与静态或简单运动时表现不同,而这些动态场景恰恰是真实人机交互的常态。

VII. 结论与后续步骤

A. 建议总结

基于对低延迟要求的严格考量以及对现有技术的分析,本报告推荐采用以下技术路线:

- **3D姿态估计:** 优先考虑 **RTMW3D**²⁸, 因其能提供包括手臂在内的全身3D信息,且基于的RTMPose框架展示了极高的实时性能²⁷。若仅需手部跟踪且对延迟要求极为苛刻, **MediaPipe Hands**²¹ 是备选。**ReJSHand**¹² 在GPU性能允许时也可考虑。最终选择

需实测验证。

- **轨迹规划:** 推荐采用基于一致性模型的加速扩散模型, 特别是 **FRMD**⁵¹, 它通过单步推理实现了低延迟(约17ms), 并利用运动基元提升轨迹平滑性, 直接解决了标准扩散模型的速度瓶颈。
- **RL精炼:** 建议使用 **SAC**⁵⁹ 或 **PPO**⁵⁹, 侧重于使用小型策略网络以保证低推理延迟, 并通过精心设计的奖励函数来优化跟踪精度、平滑度和响应速度。

此推荐旨在平衡低延迟、性能(精度、平滑度)和实现可行性。

B. 核心挑战应对

本规划通过以下方式应对核心的低延迟挑战:

- **技术选型:** 选择了已知具有高FPS/低延迟特性的姿态估计器(RTMW3D/MediaPipe), 以及能够实现单步推理的轨迹规划方法(FRMD), 并确认了RL策略推理通常不是瓶颈。
- **系统设计:** 强调模块化设计以便优化瓶颈, 并提出低延迟通信策略(共享内存、流水线)以减少模块间开销。
- **评估重点:** 将端到端延迟和组件延迟作为首要评估指标, 并在动态场景下进行测试。

C. 实施路线图

建议的初步实施步骤如下:

1. **环境搭建:** 配置开发环境, 连接并测试摄像头和uArm Swift Pro机械臂。
2. **姿态估计器实现与基准测试:** 在目标硬件上实现选定的3D姿态估计器(如RTMW3D), 测量其实际延迟和精度。
3. **轨迹规划器实现与基准测试:** 实现选定的轨迹规划器(如FRMD)。这可能需要设置训练流程(可能需要收集专家数据), 并验证单步推理延迟。
4. **仿真环境开发:** 构建包含uArm模型和模拟传感器的仿真环境。
5. **仿真集成:** 在仿真环境中集成姿态估计器和规划器, 进行初步调试。
6. **RL训练:** 在仿真环境中实现并训练RL精炼策略(如SAC)。
7. **实物集成与测试:** 遵循评估协议, 谨慎地在真实uArm机械臂上进行集成测试和性能评估。

D. 未来工作与潜在增强

项目成功实施后, 可考虑以下扩展方向:

- **自适应控制:** 根据用户运动速度或任务复杂度动态调整控制参数或规划策略。
- **多用户交互:** 扩展系统以支持多个用户或更复杂的交互场景。
- **力反馈集成:** 如果硬件支持, 可以加入力反馈, 增强交互的沉浸感和物理真实感。
- **更复杂机器人/任务:** 将此框架应用于具有更高自由度或执行更复杂操作任务的机器

人。

Works cited

1. Robotic Table Tennis: A Case Study into a High Speed Learning System - arXiv, accessed April 19, 2025, <https://arxiv.org/html/2309.03315v2>
2. Robotic Table Tennis: A Case Study - Google Sites, accessed April 19, 2025, <https://sites.google.com/view/robotictabletennis>
3. Robotic Table Tennis: A Case Study into a High Speed Learning System - arXiv, accessed April 19, 2025, <https://arxiv.org/abs/2309.03315>
4. www.scitepress.org, accessed April 19, 2025, <https://www.scitepress.org/Papers/2019/73947/73947.pdf>
5. Cross-View Tracking for Multi-Human 3D Pose Estimation at Over 100 FPS - CVF Open Access, accessed April 19, 2025, https://openaccess.thecvf.com/content_CVPR_2020/papers/Chen_Cross-View_Tracking_for_Multi-Human_3D_Pose_Estimation_at_Over_100_CVPR_2020_paper.pdf
6. 3D Pose Estimation of Two Interacting Hands from a Monocular Event Camera - arXiv, accessed April 19, 2025, <https://arxiv.org/html/2312.14157v1>
7. 3D Pose Estimation of Two Interacting Hands from a Monocular Event Camera - OpenReview, accessed April 19, 2025, <https://openreview.net/pdf?id=DExV8hqPOU>
8. FrankMocap: A Monocular 3D Whole-Body Pose Estimation System via Regression and Integration - ResearchGate, accessed April 19, 2025, https://www.researchgate.net/publication/356510588_FrankMocap_A_Monocular_3D_Whole-Body_Pose_Estimation_System_via_Regression_and_Integration
9. FrankMocap: A Monocular 3D Whole-Body Pose Estimation System via Regression and Integration - CVF Open Access, accessed April 19, 2025, https://openaccess.thecvf.com/content/ICCV2021W/ACVR/papers/Rong_FrankMocap_A_Monocular_3D_Whole-Body_Pose_Estimation_System_via_Regression_ICCVW_2021_paper.pdf
10. Direction-Aware Hybrid Representation Learning for 3D Hand Pose and Shape Estimation, accessed April 19, 2025, <https://arxiv.org/html/2504.01298v1>
11. ReJSHand: Efficient Real-Time Hand Pose Estimation and Mesh Reconstruction Using Refined Joint and Skeleton Features | Request PDF - ResearchGate, accessed April 19, 2025, https://www.researchgate.net/publication/389714661_ReJSHand_Efficient_Real-Time_Hand_Pose_Estimation_and_Mesh_Reconstruction_Using_Refined_Joint_and_Skeleton_Features
12. ReJSHand: Efficient Real-Time Hand Pose Estimation and Mesh Reconstruction Using Refined Joint and Skeleton Features - arXiv, accessed April 19, 2025, <https://arxiv.org/html/2503.05995v1>
13. Efficient Real-Time Hand Pose Estimation and Mesh Reconstruction Using Refined Joint and Skeleton Features - arXiv, accessed April 19, 2025, <https://www.arxiv.org/pdf/2503.05995>

14. [2503.05995] ReJSHand: Efficient Real-Time Hand Pose Estimation and Mesh Reconstruction Using Refined Joint and Skeleton Features - arXiv, accessed April 19, 2025, <https://arxiv.org/abs/2503.05995>
15. Modeling and Capturing Hands and Bodies Together - arXiv, accessed April 19, 2025, <https://arxiv.org/pdf/2201.02610>
16. Towards Robust and Expressive Whole-body Human Pose and Shape Estimation | OpenReview, accessed April 19, 2025, <https://openreview.net/forum?id=XfKnoW4Zef-eId=5bbex49VKz>
17. eth-siplab/EgoPoser: Official Code for ECCV 2024 paper "EgoPoser: Robust Real-Time Egocentric Pose Estimation from Sparse and Intermittent Observations Everywhere" - GitHub, accessed April 19, 2025, <https://github.com/eth-siplab/EgoPoser>
18. [2008.08324] FrankMocap: Fast Monocular 3D Hand and Body ..., accessed April 19, 2025, <https://ar5iv.labs.arxiv.org/html/2008.08324>
19. Using a single RGB frame for real time 3D hand pose estimation in the wild - ar5iv - arXiv, accessed April 19, 2025, <https://ar5iv.labs.arxiv.org/html/1712.03866>
20. FrankMocap: Fast Monocular 3D Hand and Body Motion Capture by Regression and Integration - Yu Rong, accessed April 19, 2025, https://penincillin.github.io/frankmocap_iccvw2021
21. MediaPipe Hand Detection · Models · Dataloop, accessed April 19, 2025, https://dataloop.ai/library/model/qualcomm_mediapipe-hand-detection/
22. Real-Time Gesture-Based Hand Landmark Detection for Optimized Mobile Photo Capture and Synchronization - MDPI, accessed April 19, 2025, <https://www.mdpi.com/2079-9292/14/4/704>
23. Quantifying Similarities Between MediaPipe and a Known Standard to Address Issues in Tracking 2D Upper Limb Trajectories: Proof of Concept Study, accessed April 19, 2025, <https://pmc.ncbi.nlm.nih.gov/articles/PMC11683656/>
24. Hand landmarks detection guide | Google AI Edge | Google AI for ..., accessed April 19, 2025, https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker
25. What is OpenPose? A Guide for Beginners. - Roboflow Blog, accessed April 19, 2025, <https://blog.roboflow.com/what-is-openpose/>
26. 3D Pose Estimation | Papers With Code, accessed April 19, 2025, <https://paperswithcode.com/task/3d-pose-estimation/latest>
27. (PDF) RTMPose: Real-Time Multi-Person Pose Estimation based on ..., accessed April 19, 2025, https://www.researchgate.net/publication/369233598_RTMPose_Real-Time_Multi-Person_Pose_Estimation_based_on_MMPose
28. Releases · open-mmlab/mmpose - GitHub, accessed April 19, 2025, <https://github.com/open-mmlab/mmpose/releases>
29. RTMPose: Real-Time Multi-Person Pose Estimation based on ..., accessed April 19, 2025, <https://arxiv.org/pdf/2303.07399>
30. Deep Learning for 3D Human Pose Estimation and Mesh Recovery: A Survey - arXiv, accessed April 19, 2025, <https://arxiv.org/html/2402.18844v1>
31. 3D Human Pose Estimation Experiments and Analysis - KDnuggets, accessed April

- 19, 2025,
<https://www.kdnuggets.com/2020/08/3d-human-pose-estimation-experiments-analysis.html>
32. RTMW: Real-Time Multi-Person 2D and 3D Whole-body Pose Estimation - arXiv, accessed April 19, 2025, <https://arxiv.org/html/2407.08634v1>
 33. RTMW: Real-Time Multi-Person 2D and 3D Whole-body Pose Estimation, accessed April 19, 2025,
<https://paperswithcode.com/paper/rtmw-real-time-multi-person-2d-and-3d-whole>
 34. [2407.08634] RTMW: Real-Time Multi-Person 2D and 3D Whole-body Pose Estimation, accessed April 19, 2025, <https://arxiv.org/abs/2407.08634>
 35. We've released the highly efficient 2D/3D whole-body pose estimation models RTMW/RTMW3D. : r/computervision - Reddit, accessed April 19, 2025,
https://www.reddit.com/r/computervision/comments/1e3p1lj/weve_released_the_highly_efficient_2d3d_wholebody/
 36. arXiv:1712.03866v1 [cs.CV] 11 Dec 2017, accessed April 19, 2025,
<http://arxiv.org/pdf/1712.03866>
 37. Dynamic NeRF: A Review - arXiv, accessed April 19, 2025,
<https://arxiv.org/html/2405.08609>
 38. Motion-Oriented Compositional Neural Radiance Fields for Monocular Dynamic Human Modeling - arXiv, accessed April 19, 2025,
<https://arxiv.org/html/2407.11962v2>
 39. City-on-Web: Real-time Neural Rendering of Large-scale Scenes on the Web - arXiv, accessed April 19, 2025, <https://arxiv.org/html/2312.16457v1>
 40. Evaluating Human Perception of Novel View Synthesis: Subjective Quality Assessment of Gaussian Splatting and NeRF in Dynamic Scenes - arXiv, accessed April 19, 2025, <https://arxiv.org/html/2501.08072v1>
 41. Quark: Real-time, High-resolution, and General Neural View Synthesis - arXiv, accessed April 19, 2025, <https://arxiv.org/html/2411.16680v1>
 42. CVPR Poster EventNeRF: Neural Radiance Fields From a Single Colour Event Camera, accessed April 19, 2025,
<https://cvpr2023.thecvf.com/virtual/2023/poster/22210>
 43. Sequence-based imitation learning for surgical robot operations - OAE Publishing Inc., accessed April 19, 2025, <https://www.oaepublish.com/articles/ais.2024.32>
 44. Diffusion Trajectory-guided Policy for Long-horizon Robot Manipulation - arXiv, accessed April 19, 2025, <https://arxiv.org/html/2502.10040v1>
 45. Any-point Trajectory Modeling for Policy Learning - arXiv, accessed April 19, 2025, <https://arxiv.org/html/2401.00025v2>
 46. Any-point Trajectory Modeling for Policy Learning - arXiv, accessed April 19, 2025, <https://arxiv.org/html/2401.00025v1>
 47. [2401.00025] Any-point Trajectory Modeling for Policy Learning - arXiv, accessed April 19, 2025, <https://arxiv.org/abs/2401.00025>
 48. Any-point Trajectory Modeling for Policy Learning - Xingyu Lin, accessed April 19, 2025, <https://xingyu-lin.github.io/atm/>
 49. Prof. Yang Gao's Research Team Proposes An Innovative Any-point Trajectory

- Modeling (ATM) for Policy Learning, accessed April 19, 2025,
<https://iis.tsinghua.edu.cn/en/show-10563-1.html>
50. Embedding Koopman Optimal Control in Robot Policy Learning - GitHub Pages, accessed April 19, 2025,
https://koopman-learning.github.io/web/files/hang_embedding.pdf
 51. FRMD: Fast Robot Motion Diffusion with Consistency-Distilled Movement Primitives for Smooth Action Generation - arXiv, accessed April 19, 2025,
<https://arxiv.org/html/2503.02048v1>
 52. ManiCM: Real-time 3D Diffusion Policy via Consistency Model for Robotic Manipulation - arXiv, accessed April 19, 2025, <https://arxiv.org/pdf/2406.01586?>
 53. FRMD: Fast Robot Motion Diffusion with Consistency-Distilled Movement Primitives for Smooth Action Generation - arXiv, accessed April 19, 2025,
<https://arxiv.org/pdf/2503.02048?>
 54. [2503.02048] FRMD: Fast Robot Motion Diffusion with Consistency-Distilled Movement Primitives for Smooth Action Generation - arXiv, accessed April 19, 2025, <https://arxiv.org/abs/2503.02048>
 55. [Literature Review] FRMD: Fast Robot Motion Diffusion with Consistency-Distilled Movement Primitives for Smooth Action Generation - Moonlight, accessed April 19, 2025,
<https://www.themoonlight.io/review/frmd-fast-robot-motion-diffusion-with-consistency-distilled-movement-primitives-for-smooth-action-generation>
 56. arxiv.org, accessed April 19, 2025, <https://arxiv.org/pdf/2503.02048>
 57. Diffusion Models for Robotic Manipulation: A Survey - arXiv, accessed April 19, 2025, <https://arxiv.org/html/2504.08438v1>
 58. Imitation Learning from MPC for Quadrupedal Multi-Gait Control (ICRA 2021 Presentation), accessed April 19, 2025,
<https://www.youtube.com/watch?v=S4WYs6mBAwI>
 59. A Comprehensive Review of Deep Learning Techniques in Mobile Robot Path Planning: Categorization and Analysis - MDPI, accessed April 19, 2025,
<https://www.mdpi.com/2076-3417/15/4/2179>
 60. Optimizing Autonomous Vehicle Performance Using Improved Proximal Policy Optimization, accessed April 19, 2025,
<https://www.mdpi.com/1424-8220/25/6/1941>
 61. (PDF) Digital Twin-Empowered Robotic Arm Control: An Integrated PPO and Fuzzy PID Approach - ResearchGate, accessed April 19, 2025,
https://www.researchgate.net/publication/387922790_Digital_Twin-Empowered_Robotic_Arm_Control_An_Integrated_PPO_and_Fuzzy_PID_Approach
 62. Sim-to-Real: A Performance Comparison of PPO, TD3, and SAC Reinforcement Learning Algorithms for Quadruped Walking Gait Generation - Scientific Research Publishing, accessed April 19, 2025,
<https://www.scirp.org/journal/paperinformation?paperid=131938>
 63. HEPPO: HARDWARE-EFFICIENT PROXIMAL POLICY OPTIMIZATION - MacSphere, accessed April 19, 2025,
https://macsphere.mcmaster.ca/bitstream/11375/30655/2/taha_hazem_a_202412_masc.pdf

64. Low-Precision Reinforcement Learning: Running Soft Actor-Critic in Half Precision
- CS@Cornell, accessed April 19, 2025,
https://www.cs.cornell.edu/gomes/pdf/2021_bjorck_icml_sac.pdf
65. Low-Precision Reinforcement Learning - CS@Cornell, accessed April 19, 2025,
https://www.cs.cornell.edu/gomes/pdf/2021_bjorck_arxiv_rl.pdf
66. Sample-efficient Reinforcement Learning in Robotic Table Tennis* - arXiv,
accessed April 19, 2025, <https://arxiv.org/html/2011.03275v4>