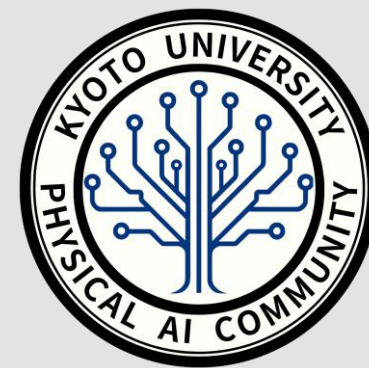


KUPACにおけるVLA検証基盤の構築

平塚謙良

京都大学大学院 情報学研究科

2026/09/19 第15回 iHR研究会



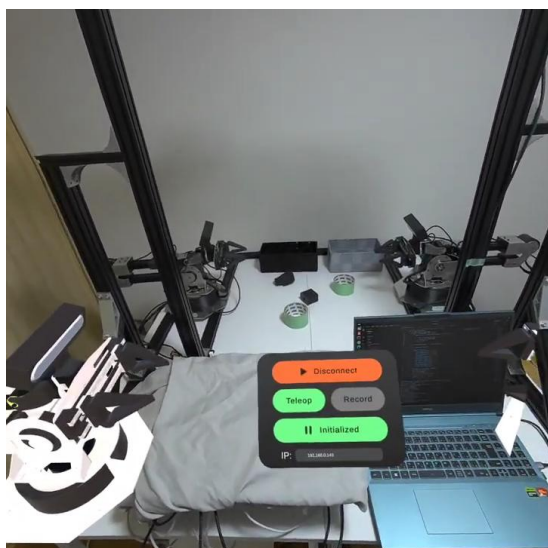
自己紹介



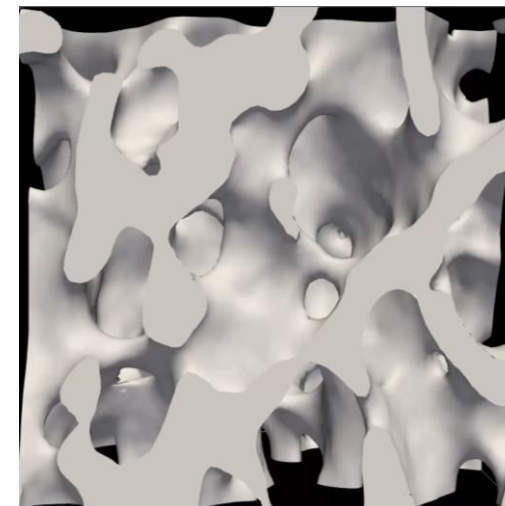
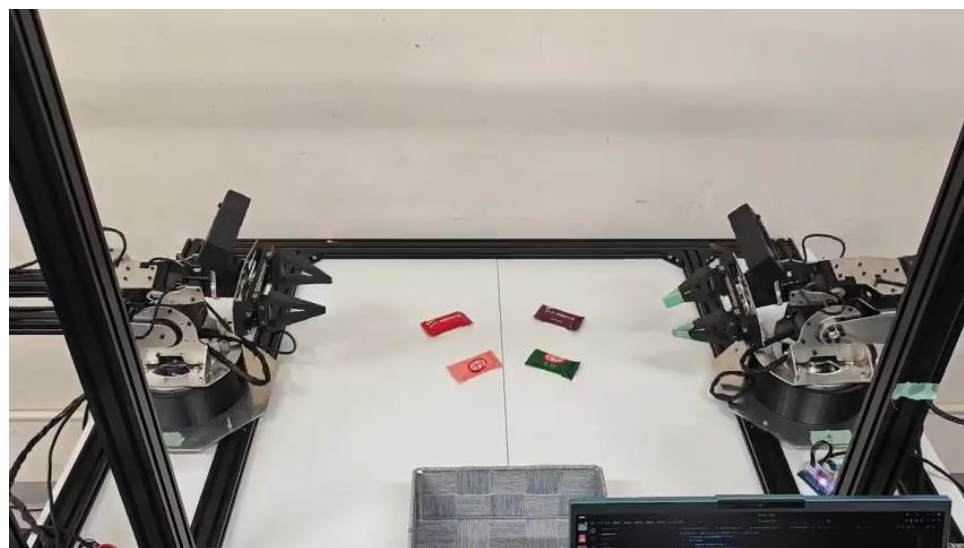
氏名：平塚 謙良

所属：京都大学大学院 情報学研究科 森本研 M1

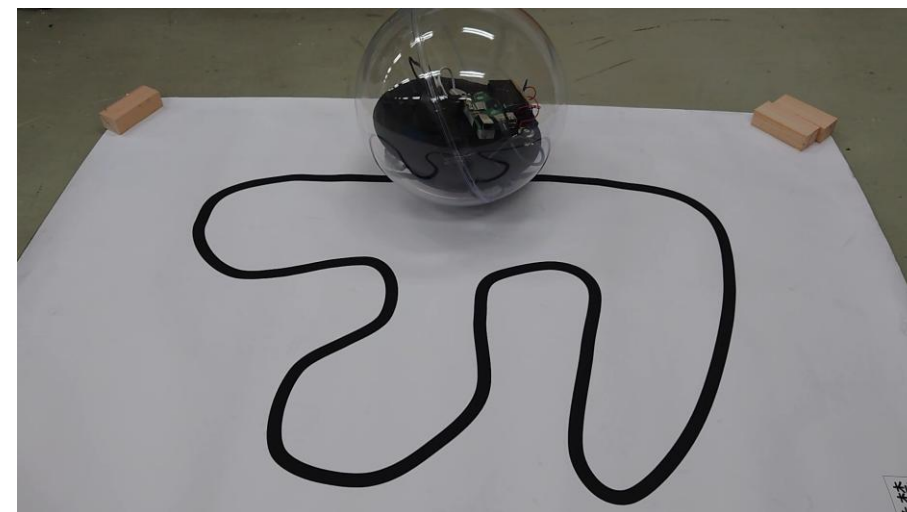
分野：ロボット学習, ロボット聴覚, 微分可能シミュレータ



自作ロボットを用いたデータ収集・模倣学習



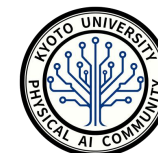
微分可能骨代謝シミュレータ



強化学習を使った球形ライントレーサ²

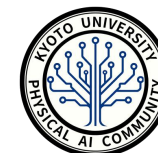
目次

- VLA検証基盤を作る
- VLAを動かしてみる
- 様々な課題への対応



目次

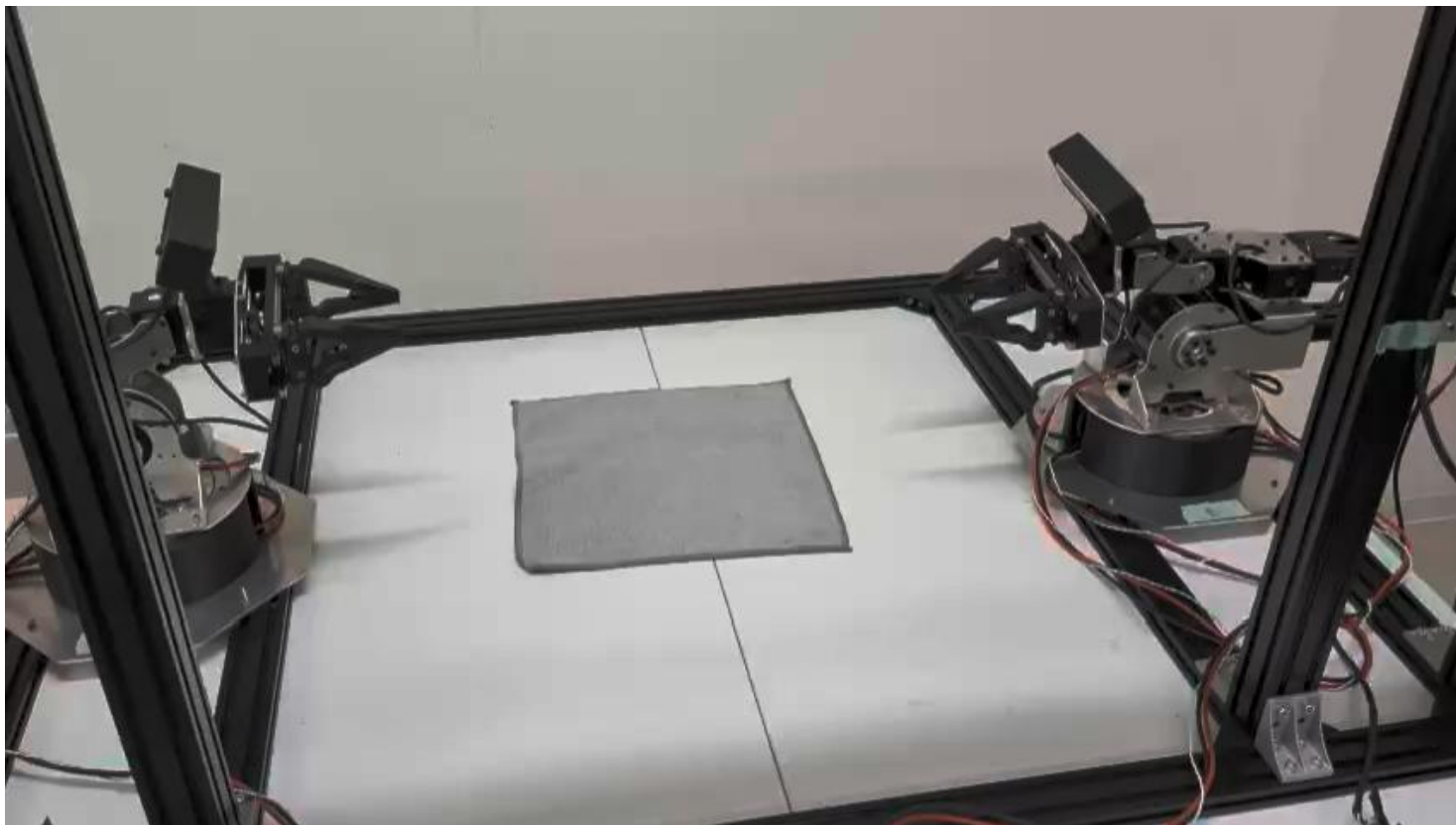
- VLA検証基盤を作る
- VLAを動かしてみる
- 様々な課題への対応



最終的にできたもの

X-VLAを用いた自作双腕ロボットILOHAによるハンカチの折り畳み

- 低予算で実機検証を回せる環境をどう構築したか
- 構築の過程で躓いた点と，そこから得た知見を中心に紹介



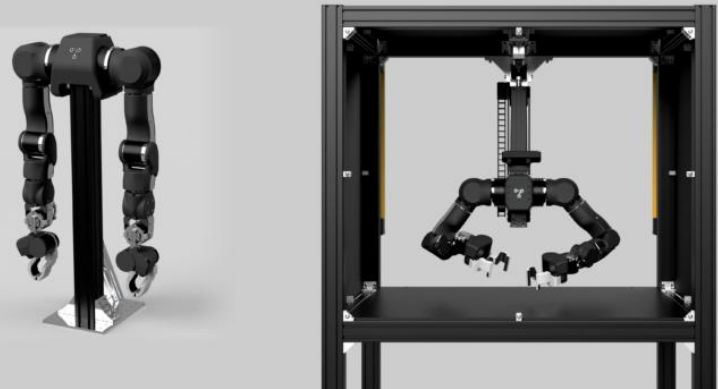
VLA検証を支える既存基盤

ロボット, データ収集, モデル学習・デプロイの各領域で基盤が整ってきた

- ALOHA/OpenArm : データ収集・評価環境まで可能な双腕テレオペ基盤
- YUBI : UMI形式の安価な双腕データを収集
- LeRobot : データ収集から学習・実行までを統一

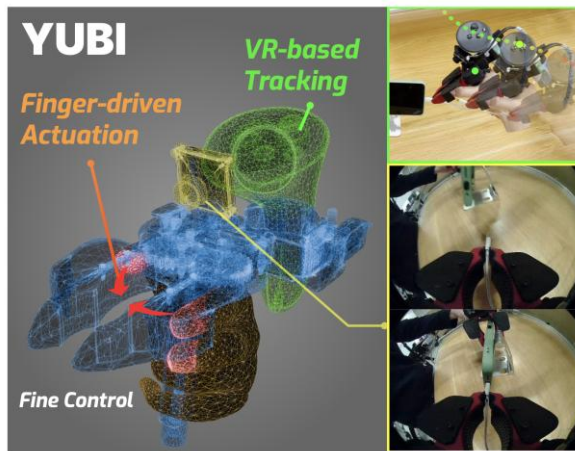
ロボット

ALOHA / OpenArm 2.0



データ収集

Leader-Follower / YUBI



モデル学習・デプロイ

LeRobot

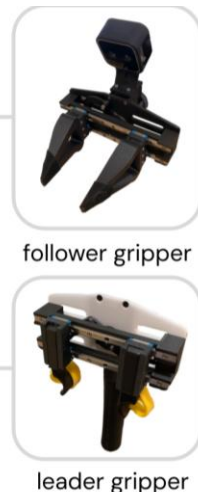
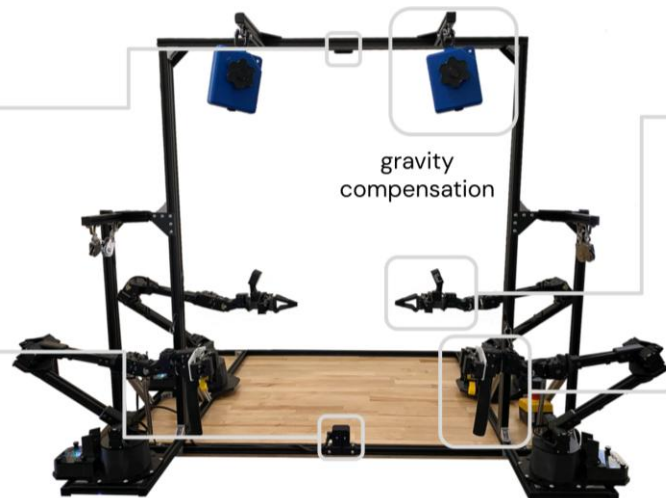
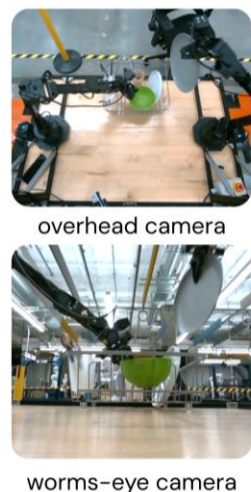


KUPACでも一連の検証ループを回せる環境を作りたい

KUPACでのVLA検証基盤構築

限られた予算の中で，VLAの実機検証まで行える基盤を自作する

- コミュニティ発足当初から開発を進めていたため，予算には限りがあった
- ALOHAはフォロワーアームに加え，操作用アームも必要でコストが課題
- 低コストで実機検証まで到達できる基盤の構築を目標に



ALOHA : 数百万円, 4本のアーム

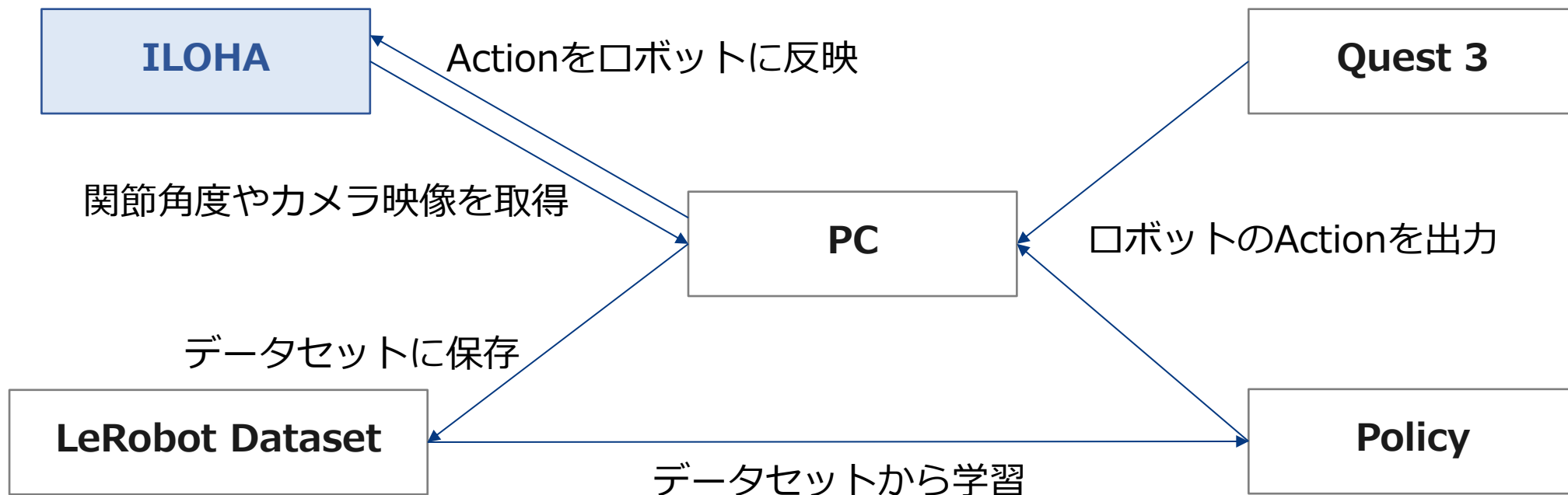


ILOHA : ~50万円, アーム2本+ヘッドセット

構築したVLA検証基盤

データ収集→学習→デプロイ までの一連の流れを実装

- Quest 3のコントローラ姿勢でILOHAを操作し，画像と関節角度をLeRobot Datasetに記録
- 学習したPolicyを，ILOHAへデプロイして実機検証



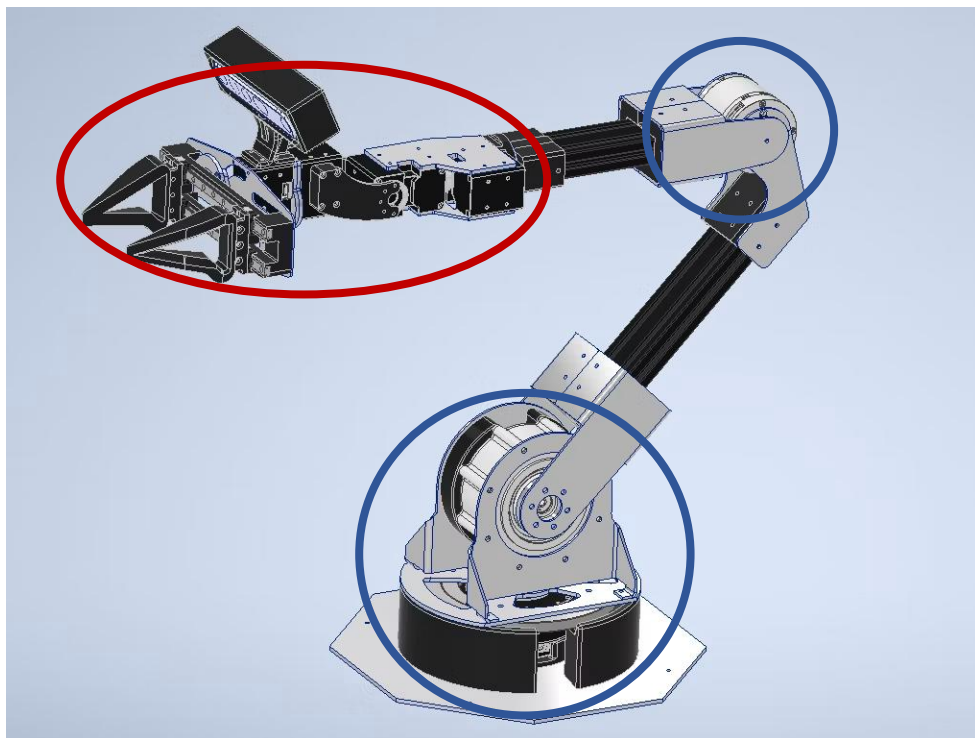
ILOHA : ハードウェア構成

「低コスト」と「標準機との互換性」を両立する構成

- ベース側：ALOHAのダブルモータ構成を，高トルクなQDDモータ（Robstride）に置換
- エンドエフェクタ側：観測画像に写りやすいため，標準機（WidowX）と同じDynamixel構成
- 6軸＋グリッパ×2の双腕構成

視覚的互換性

エンドエフェクタ側にはカメラが取り付けられているため，ALOHAと同じ見た目となるよう設計



低コスト化

ベース側をダブルモータから安価なQDDモータへ

■ 先端側：Dynamixel

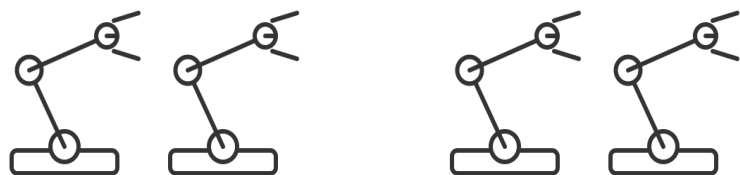
■ 根元側：Robstride（QDD）

ILOHA : リーダーアームの仮想化

物理的なリーダーアームを置かず，MR空間内の仮想的なアームを利用

- 一般的なALOHA型 : Leader $\times 2$ + Follower $\times 2$ $\rightarrow$ 費用と設置スペースが増加
 - VRテレオペ : Followerアームのみで済むため，省コスト，省スペース
- $\rightarrow$ Quest 3のコントローラ姿勢からIKで双腕の関節角度を計算し，ILOHAを操作

一般的なALOHA型

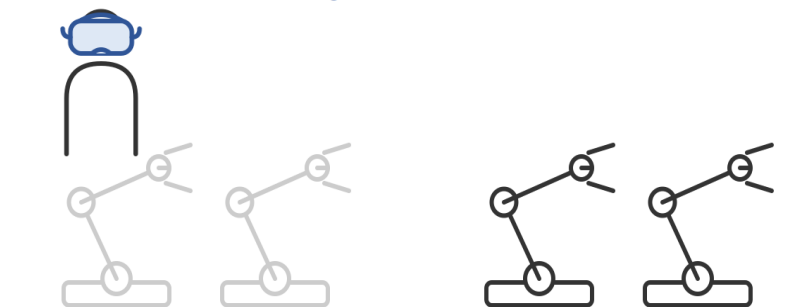


Leader $\times 2$

Follower $\times 2$

アーム4本+設置スペース
(リーダー分の費用と場所が必要)

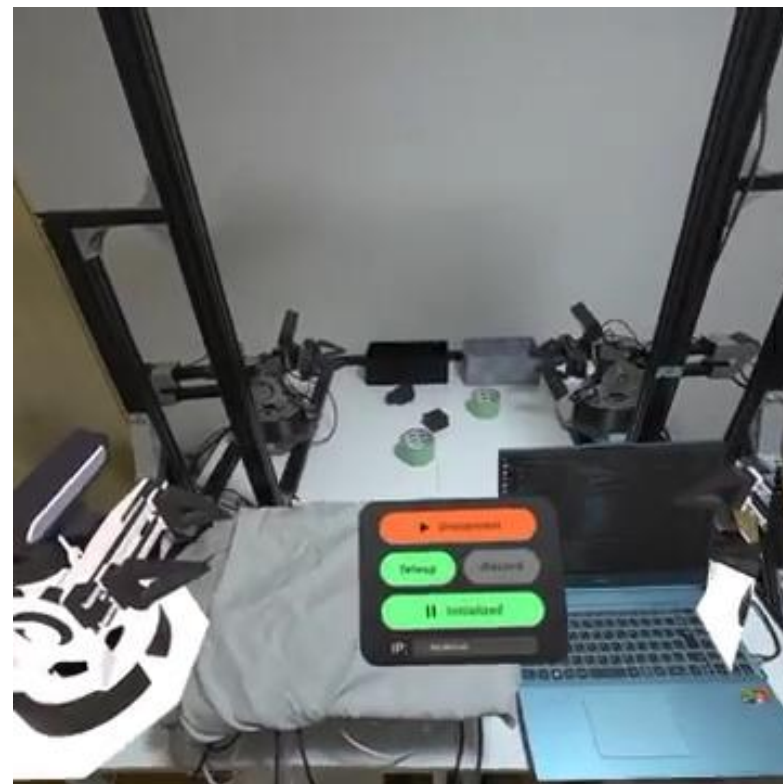
ILOHA



Virtual Leader
(Quest 3のコントローラ)

Follower $\times 2$

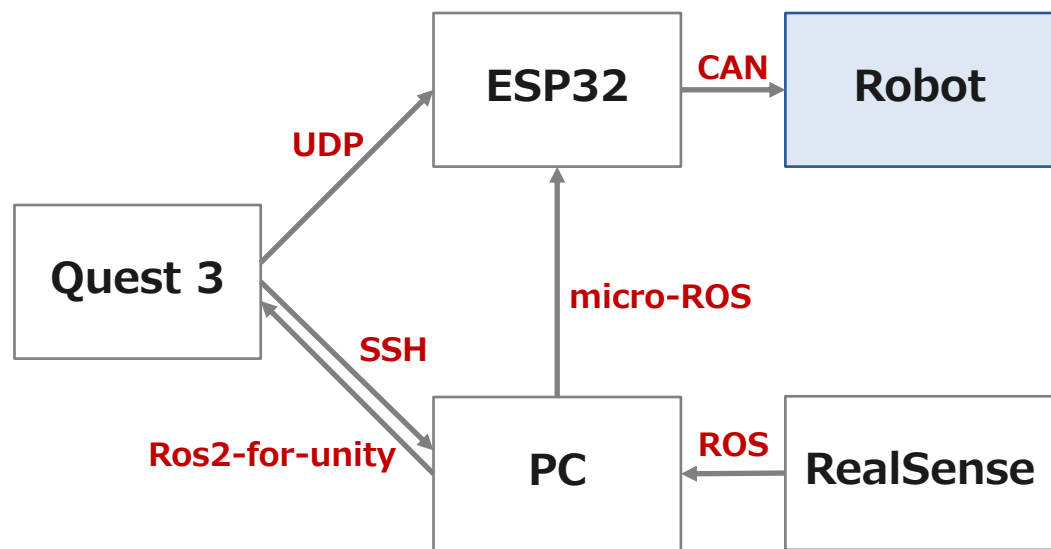
アーム2本+ヘッドセット
(物理リーダーを持たない)



初期システム構成

低遅延な追従を最優先にした結果，通信経路が複雑化

- Quest 3からESP32へUDPで関節角度を直接送信
- RealSenseの点群をROS経由でQuest 3にMR表示 → 帯域不足で粗く，操作性は向上せず
- 制御用UDPと映像用ROSが並立し，画像・状態の同期，起動，デバッグが複雑化



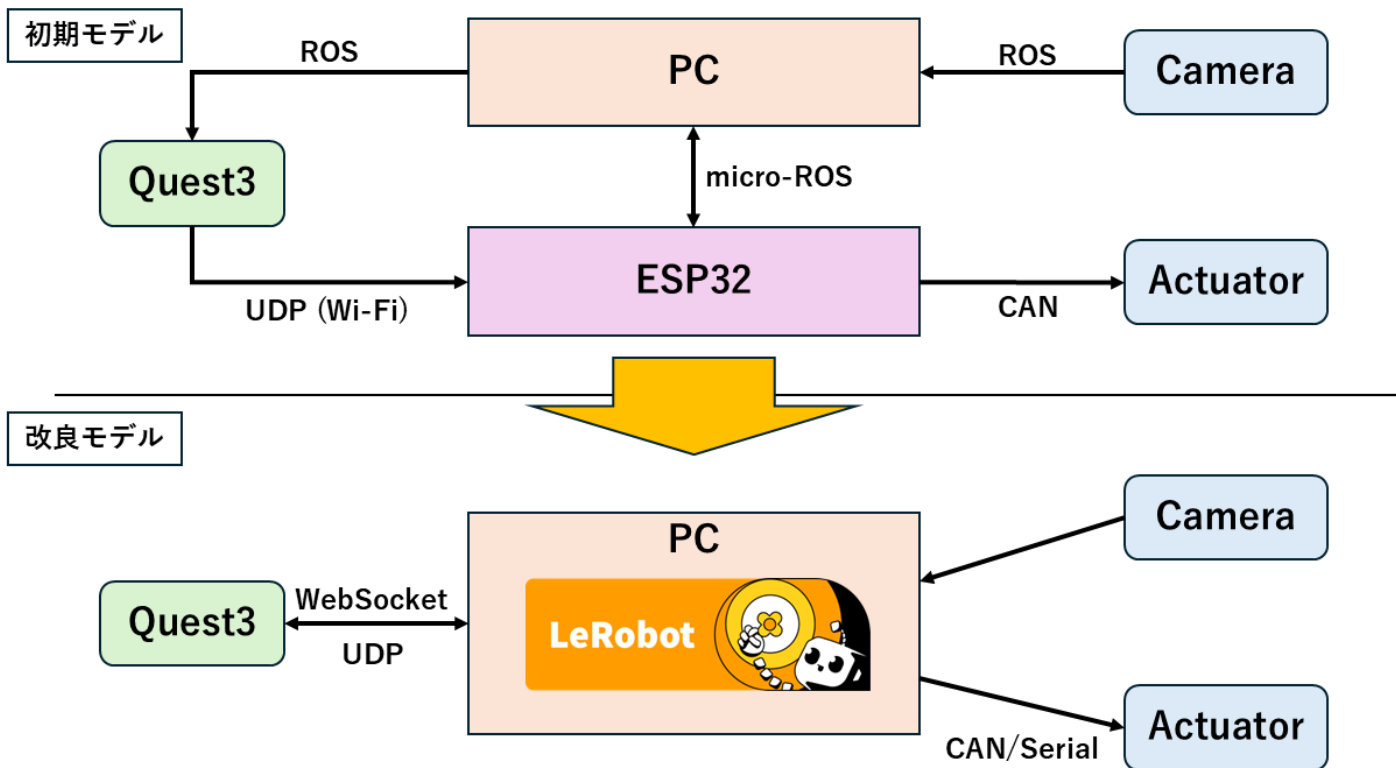
制御はUDP，映像はROSと経路が分かれ，同期と起動手順が複雑に



使いやすさを優先したシステム構成へ

低遅延だけでなく、収集・起動・デバッグのしやすさを重視した構成へ整理

- Quest 3は入力デバイス（IK計算まで）、PCが制御・データ保存を統括する役割分担に
- 確実性が必要な操作はWebSocket, 関節角度の高速配信はUDPに分離



ROS中心の設計で通信経路が2系統
→同期・起動・デバッグが煩雑

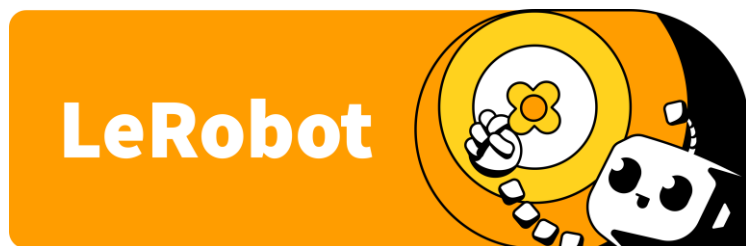
PC中心の設計で通信を簡略化
WebSocket（制御コマンド）
+ UDP（関節角度）

ヘッダー (1byte): 動作モードフラグ (動作中/停止中)
左手データ (28bytes): Float×7 (関節角度6軸+グリップパー)
右手データ (28bytes): Float×7

■ LeRobotによるデータ収集基盤

PC側のソフトウェアはできるだけLeRobotに乗っかる

- LeRobotのRobotクラスを継承し, ILOHA用のクラスを実装
 - 観測: 画像 (480×640×3) + 関節角度14次元, 行動: モータ指令値
- データ収集, Policyの学習, 実機デプロイまでが1つのレポジトリ内で完結



LeRobot互換で実装しておけば, 何もしなくても使えるPolicyや手法がどんどん増えていく

Teleoperation

Quest 3
→ Robotクラス

Record

非同期で画像・状態
を保存

Dataset

LeRobotDataset形式
で保存

Train

LeRobotに実装され
ている様々なPolicyで
学習可能

Deploy

Robotクラスで
実機推論

目次

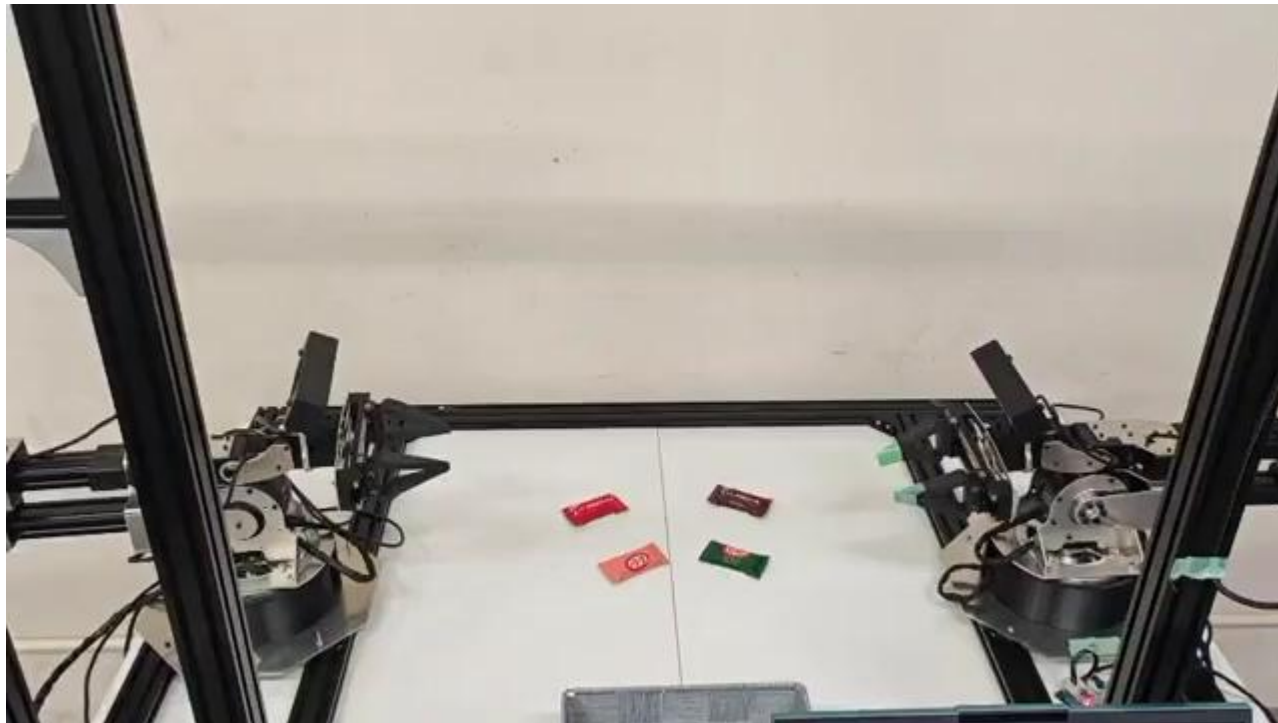
- VLA検証基盤を作る
- VLAを動かしてみる
- 様々な課題への対応



模倣学習による基盤の動作検証

データ収集から学習，実機推論までの流れが一通り動くことを確認

- 収集したデータでACTを学習し，お菓子を箱へ移すピック&プレースを実機で確認
- 構築したVLA検証基盤が動くことを確認



この時点で確認できたこと

- 収集したデータを直接学習へ投入できる
- 学習したPolicyを実機にデプロイできる
- ピック&プレースを高確率で成功

→VLA検証に向けたマイルストーン

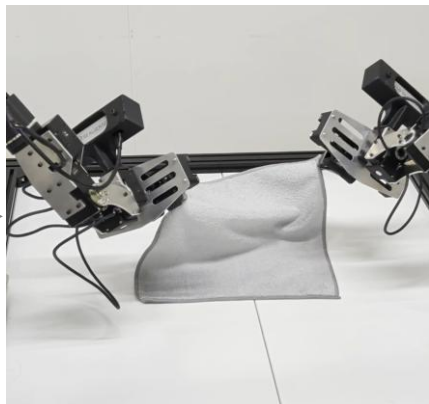
VLAでの柔軟物操作へ向けて

VLAを用いた状態変化の大きい柔軟物操作へ

- 30cm四方のハンカチを2回折り返し，長方形に仕上げるタスク
- 変形・接触・遮蔽が同時に生じるため，剛体のピック&プレイスより難しい



把持



持ち上げ



折る



再把持



折る

データ収集条件の多様化

多様な配置条件，環境条件でデータを収集

- ハンカチの初期位置・向きをランダム化
- 背景の物体配置を変更，3段階の照明条件 → 合計約307エピソードを収集

強照明

通常照明

弱照明

初期配置：
まっすぐ



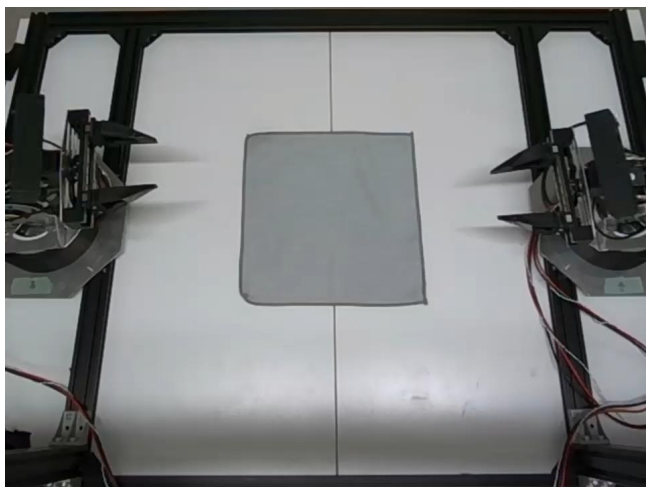
初期配置：
ななめ



デモ品質の評価

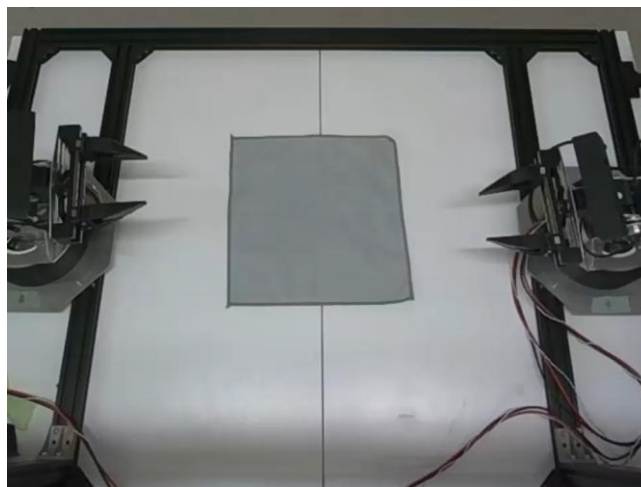
- 複数人で収集するとデータ品質に差が生じるため、キュレーションが必要
- 収集したエピソードを目視でABCの3段階に分類し、学習に用いるデータを選別

A : 通常の学習に使える高品質データ



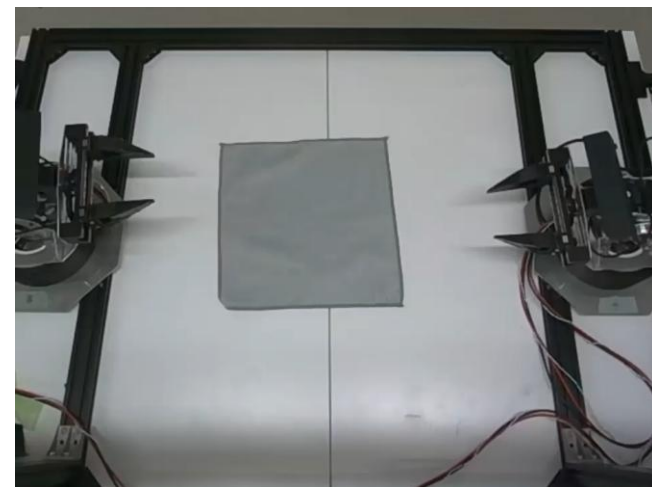
200

B : 品質がやや低いデータ



53

C : 学習に使うのが難しいデータ

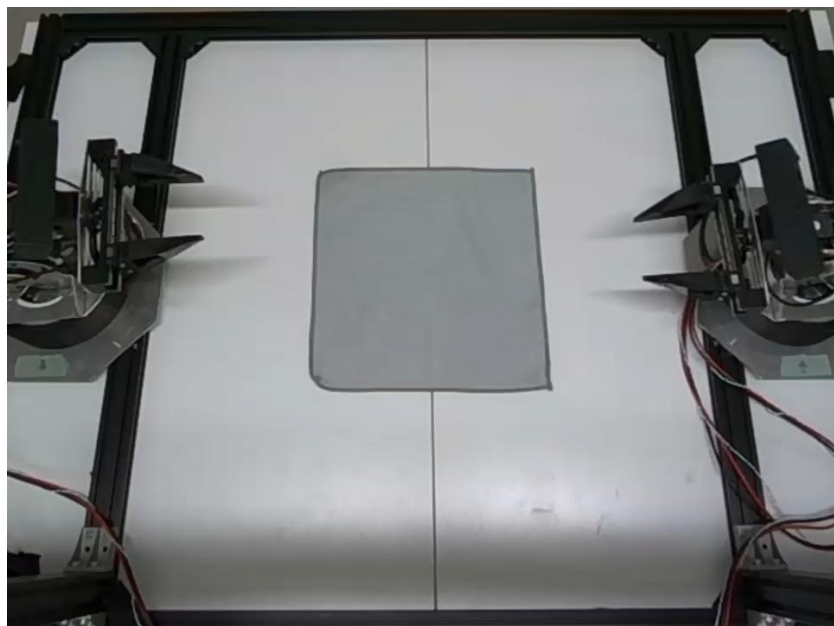


54+

既存データセットとの互換性

構成が近いロボットでも，既存データセットがそのまま使えるとは限らない

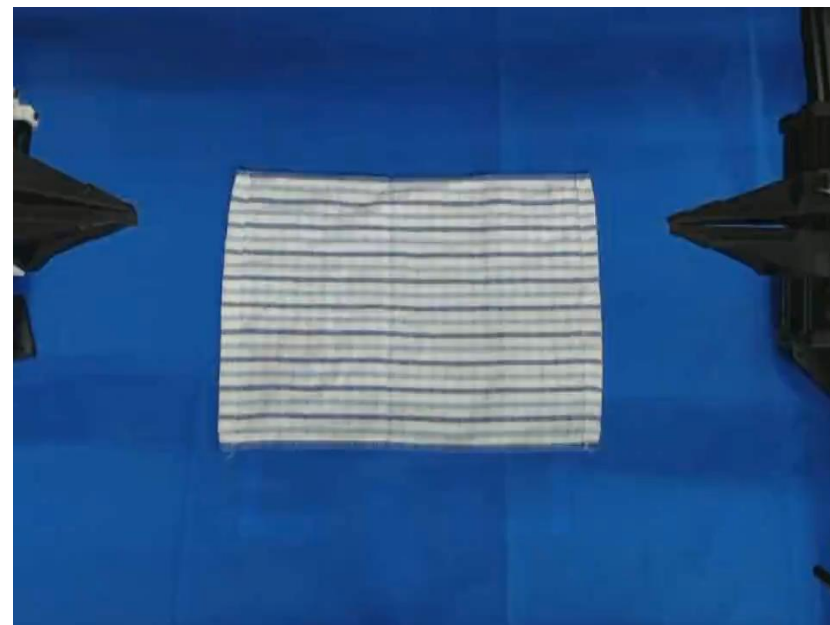
- ILOHAはALOHAに近い構成のため，AIST Bimanual Manipulation Datasetの流用を検討
 - 手首関節周辺の挙動差が大きく，事前学習への利用を断念
- 関節の構成が同じでも，IKの挙動などにより関節角度分布が大幅に変わりうる



ILOHA



手首関節の動きが
大幅に異なる



AIST Bimanual Manipulation Dataset

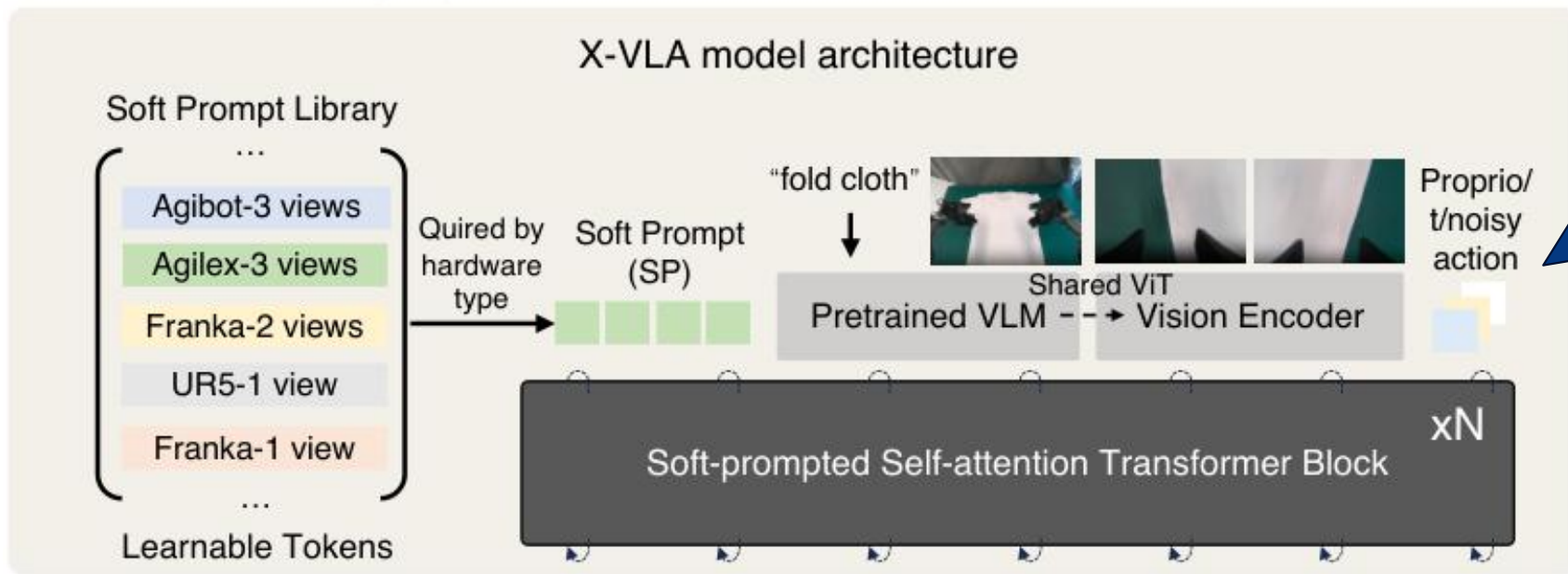
ベースPolicyの選定

ベースPolicyとしてX-VLAを採用

- X-VLA : Embodimentごとに異なるsoft promptを学習しており、 Cross-embodimentを想定
- Tシャツ畳みタスクで事前学習された重みを, ILOHAのデータでFine-tuning (soft promptも学習)

X-VLA

A simple yet scalable cross-embodiment foundation model



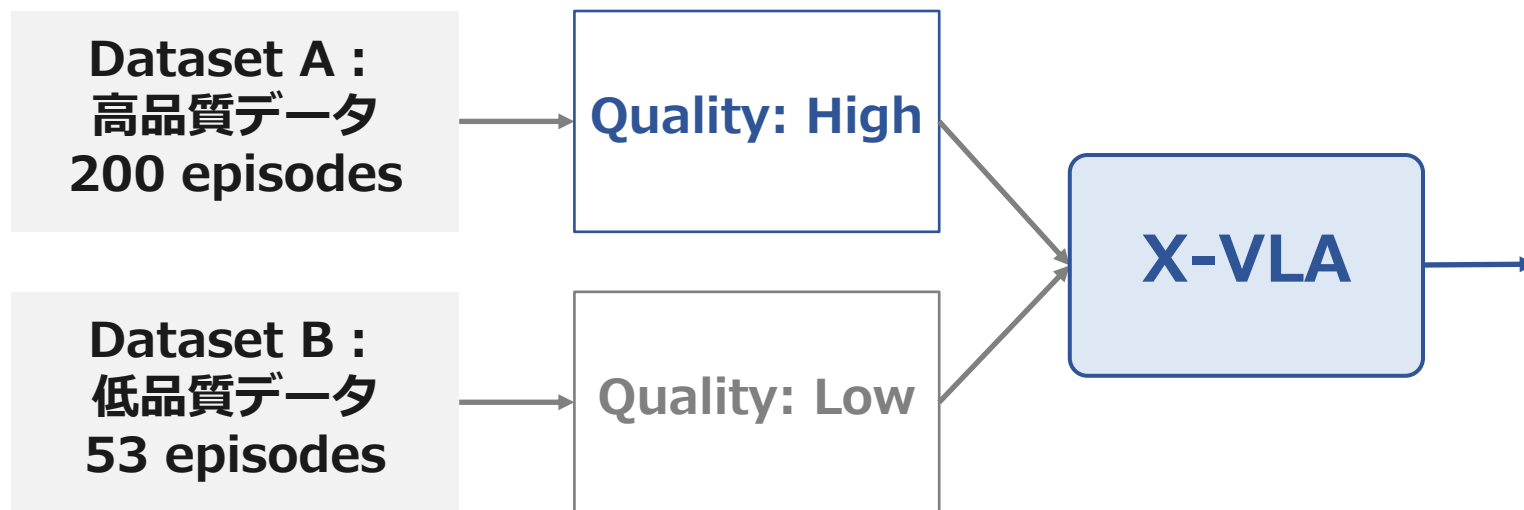
- xvla-soft-foldという大規模な柔軟物操作データセットで学習済みの重みが存在する
- VLAの中でも比較的軽量で扱いやすい (0.9B)

デモ品質を条件として学習

低品質データを捨てず，品質ラベルを言語指示に付与して学習

- $\pi_{0.7}$ のコンテキストの与え方に基づき，言語指示にデータ品質ラベルを付与
- A : 指示末尾に "Quality: High", B : "Quality: Low" を付与
→推論時はHighを指定

低品質データも「品質ラベル付きの学習データ」として活用



High/Lowによる明確な挙動差は未確認であり，効果の定量的検証は今後の課題

目次

- VLA検証基盤を作る
- VLAを動かしてみる
- 様々な課題への対応



低遅延なIKと特異姿勢

Quest 3上でのIKの不安定さ

- 6軸で冗長自由度がないため、IKを解く際に反復計算を避け、解析的に導出
→高速に処理できる一方で、特異姿勢（関節軸が一直線に並ぶ姿勢）の付近で関節角度が不安定に
- 目標姿勢と手首の関節の方向ベクトルの内積を監視し、閾値を超えたら目標姿勢を調整することで不安定な挙動を回避



Before

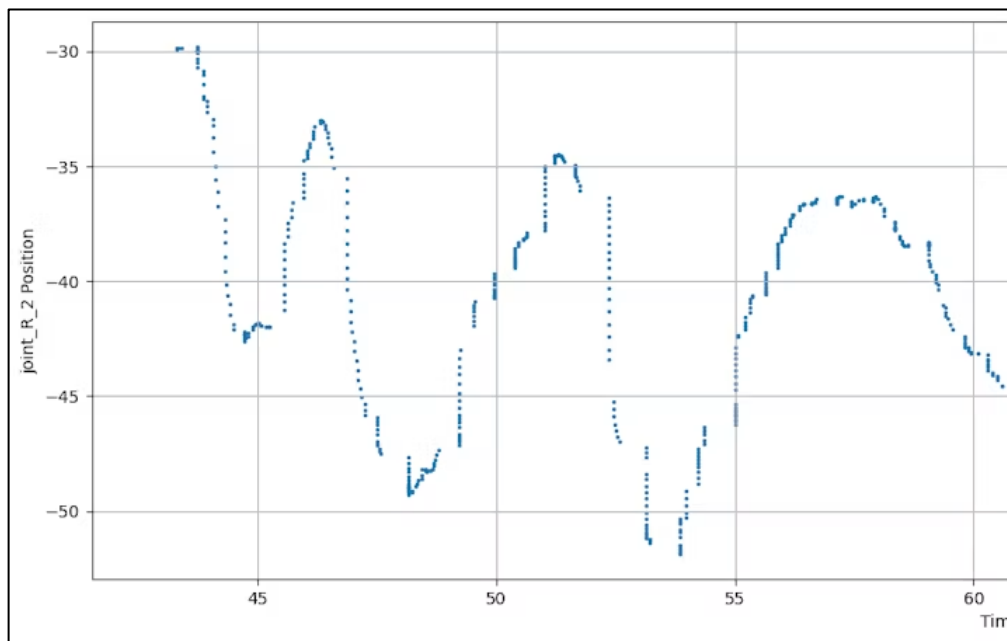


After

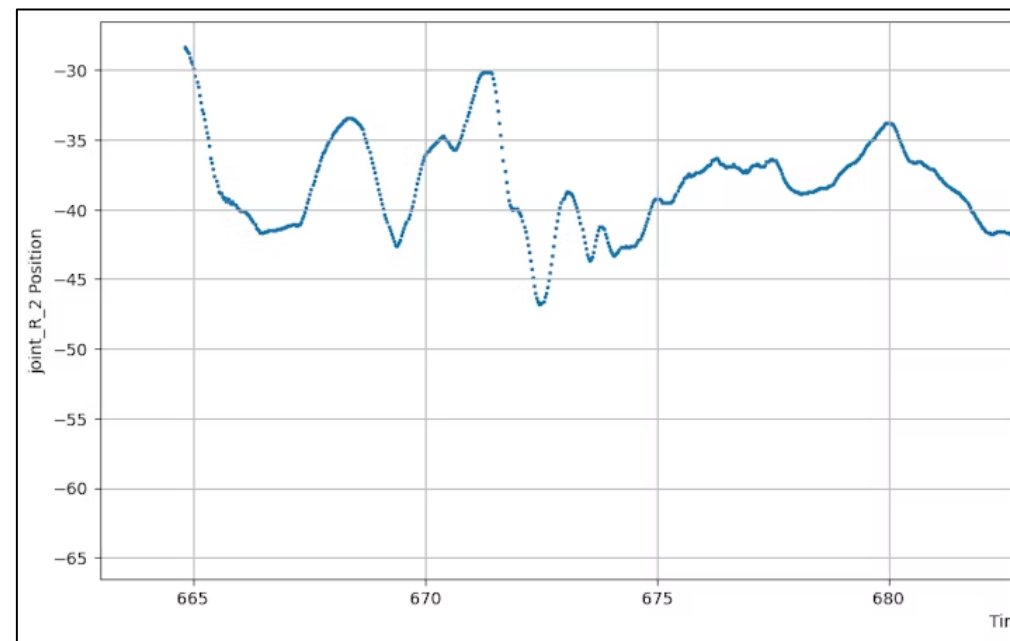
無線通信品質の問題

モバイルルータではパケットロスが頻発し，ロボットが暴走

- 無線通信品質が悪いと，受信した関節角度が階段状になり，急な動作や予期しない挙動に繋がった
- Wi-Fi 6対応ルータ（TP-Link Archer AX3000）へ変更することで，パケットロスがほぼ解消



モバイルルータ



Wi-Fi 6ルータ

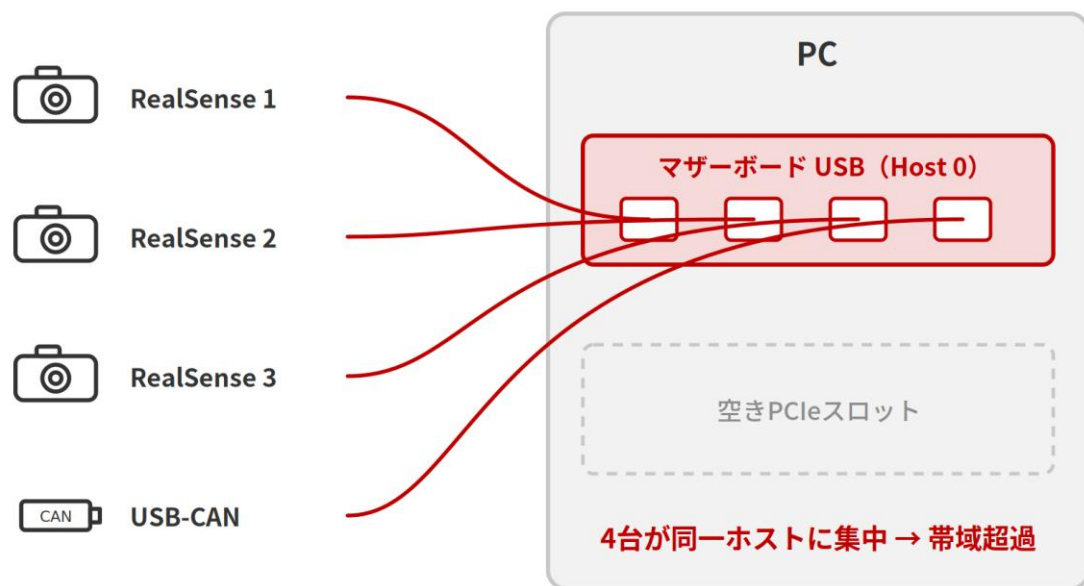
見落としがちだが，ロボット制御において無線通信の品質もかなり重要

USBコントローラの問題

RealsenseがUSBコントローラの帯域・電力を圧迫

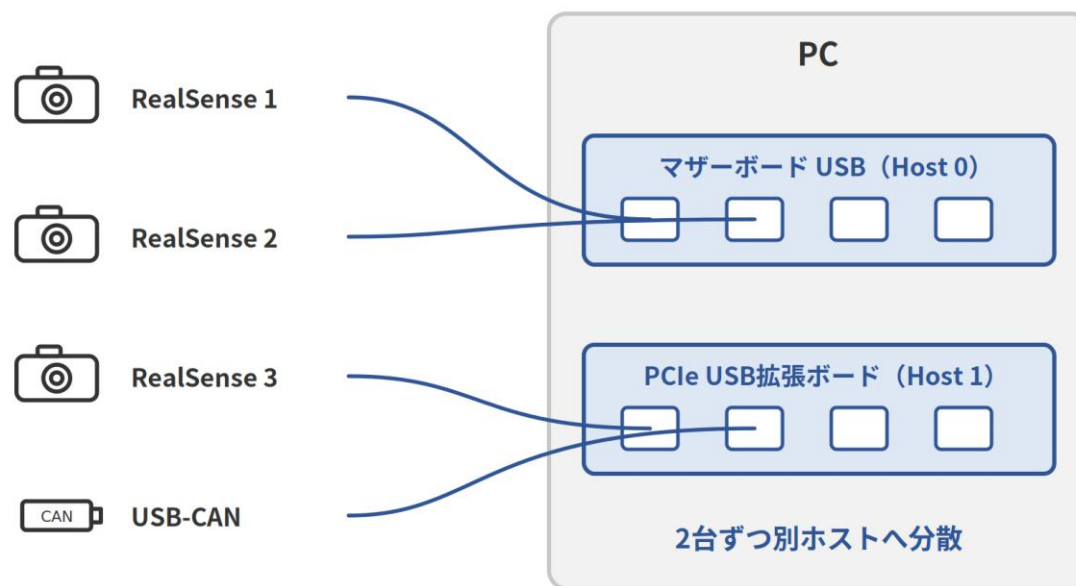
- RealSense 3台とUSB-CANが同一USB系統に集中し, フレーム落ちが発生
- PCIe USB拡張ボードを追加し, USBコントローラを分散

Before



単一のコントローラにRealsenseが集中し,
帯域・電力不足に

After : PCIe USB拡張ボードを追加

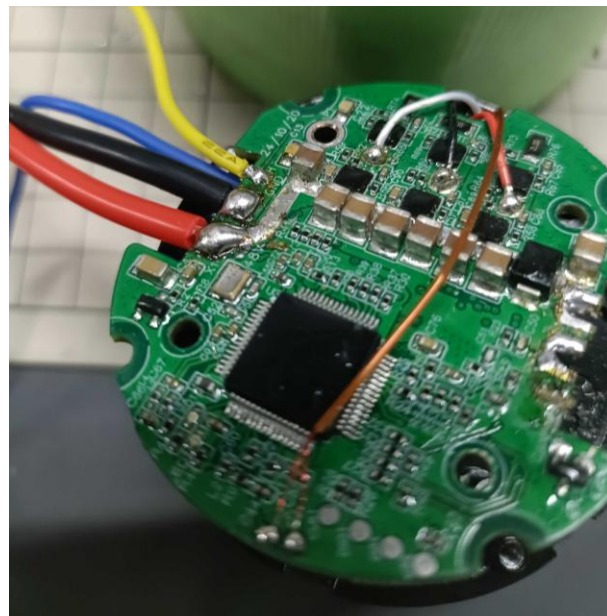
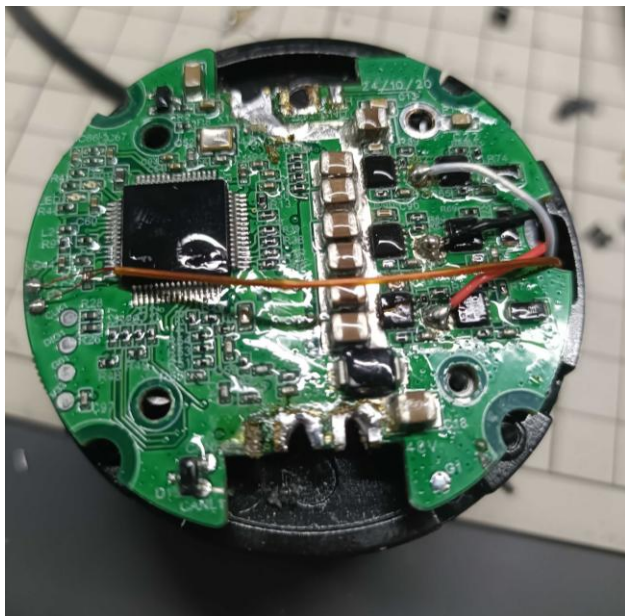


ホストコントローラを分散 → 帯域・電力を確保

Robstrideの接触不良問題

QDDモータでよく採用されているXT30 (2+2)コネクタの緩み

- XT30 (2+2)コネクタは爪がなく，挿抜回数も低いため緩みやすい
- →接触不良によりRobstrideが制御不能に
- XT30 (2+2)コネクタを基板から外し，配線を直接はんだ付けすることで，問題を根本解決



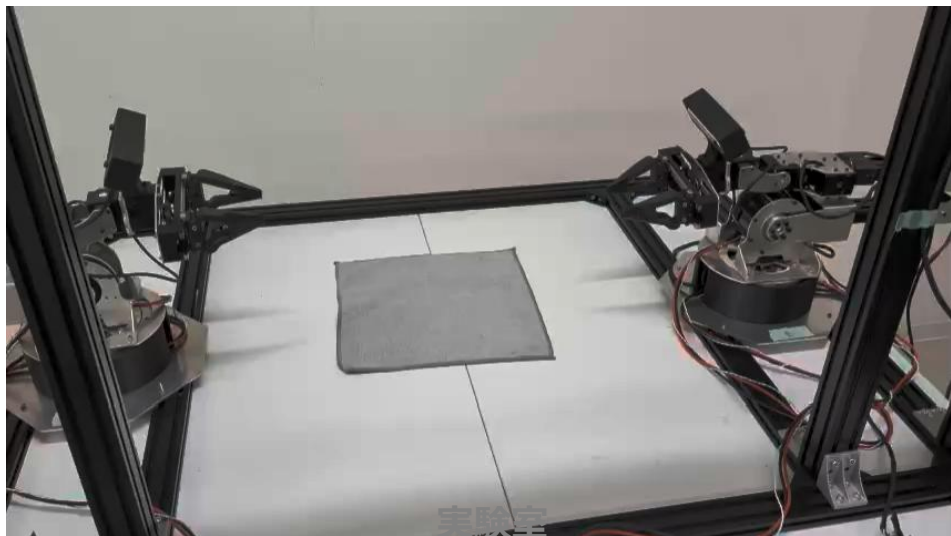
XT30 (2+2)コネクタを外して，配線を直接はんだ付け

環境変化への適応

環境が変わるとPolicy性能が大きく低下する

- SusHi Tech Tokyo 2026会場では照明・背景・カメラ角度が変化し，成功率が60～70%に低下
- 会場でPolicyを自律実行させつつ，成功データ50エピソード（約70分）収集
- 既存データと合わせ追加Fine-tuningすることで，成功率が90%近くまで回復

ラボ

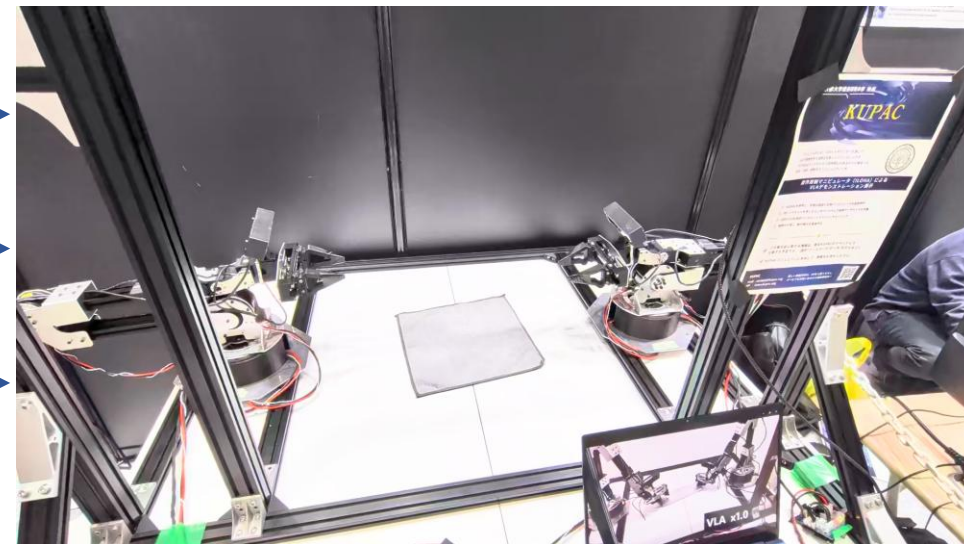


Lighting

Background

Camera Pose

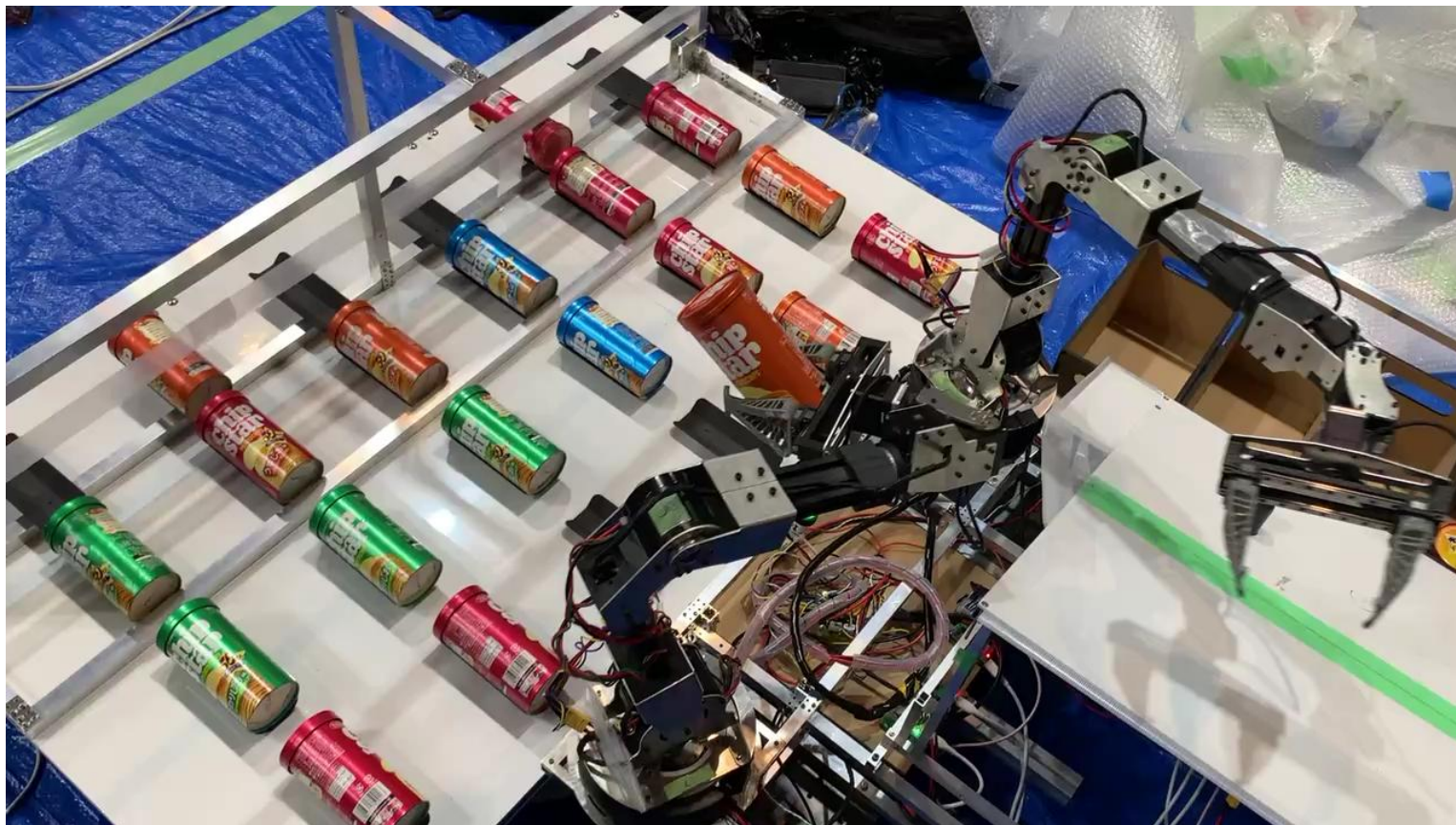
会場



ハードウェア改善後のロボットの動作

地味な改善の積み重ねで、ロボットを動かし続けられる状態に

- 安定的に動作するハードウェアがあって初めて、VLAの検証基盤が成立する



VLA検証基盤を作る

- ILOHA + Quest 3で双腕テレオペを低コスト化
- LeRobotに統合し, **収集 → 学習 → デプロイ**を一貫化

VLAを動かしてみる

- 柔軟物操作では, **デモ品質・Embodiment差・環境差**が性能に影響
- X-VLAをILOHAへFine-tuningし, 実機でハンカチ折り畳みを実現

様々な課題への対応

- IK・無線・USB・コネクタなど, **モデル外の要因も実機性能に直結**
- 動くだけでなく, **安定して動き続ける基盤**が必要