Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware

RSS 2023

양현서

July 12, 2024

About

- Author: Tony Z. Zhao, Vikash Kumar, Sergey Levine, Chelsea Finn

- Conference: RSS 2023

Motivation

- **Fine manipulation tasks** such as
  - threading cable ties
  - slotting a battery

Motivation

- **Fine manipulation tasks** such as
  - threading cable ties
  - slotting a battery

- **Requires**
  - precision
  - careful coordination of contact forces
  - closed-loop visual feedback

Low cost and imprecise hardware for fine manipulation tasks

Suggestion

Low-cost system that performs end-to-end **imitation learning** directly from real demonstrations, collected with a custom teleoperation interface

Challenges in Imitation Learning

- **Errors** in the policy can compound over time

Challenges in Imitation Learning

- **Errors** in the policy can compound over time

- Human demonstrations can be **non-stationary**

Introduction

- Fine manipulation tasks require **precision and coordination**

Introduction

- Fine manipulation tasks require **precision and coordination**
- Current systems are **expensive and complex**

Introduction

- Fine manipulation tasks require **precision and coordination**
- Current systems are **expensive and complex**
- Goal: Develop a low-cost, effective system for bimanual manipulation

Introduction

- Fine manipulation tasks require **precision and coordination**
- Current systems are **expensive and complex**
- Goal: Develop a low-cost, effective system for bimanual manipulation
- Key contributions:
  - Low-cost hardware setup
  - Novel imitation learning algorithm (ACT)
  - Successful demonstration on various tasks

System Design - Low-cost hardware

- ViperX 6-DoF robot arms
- 3D printed components
- Cost: <$20k

System Design - Design principles

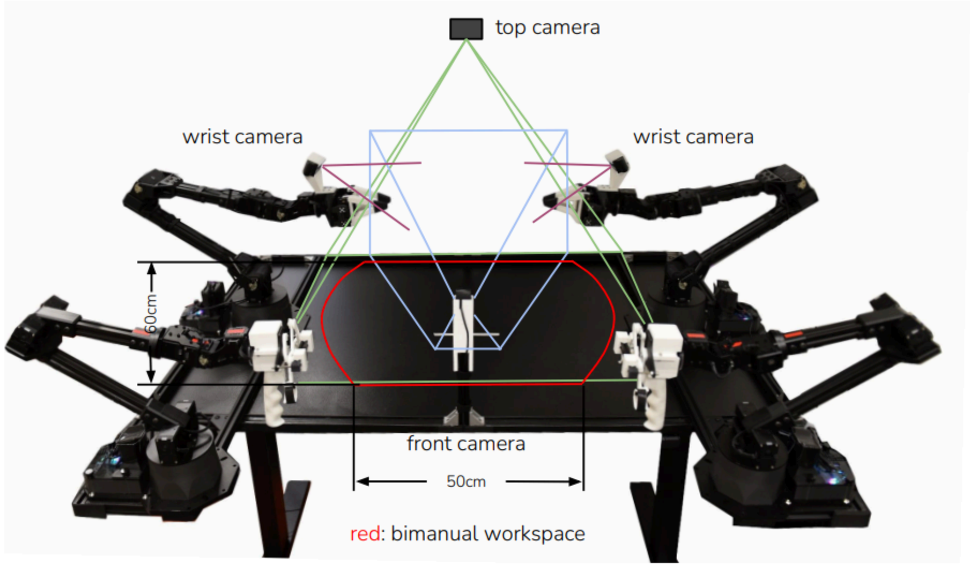
- Versatile
- User-friendly
- Repairable
- Easy-to-build

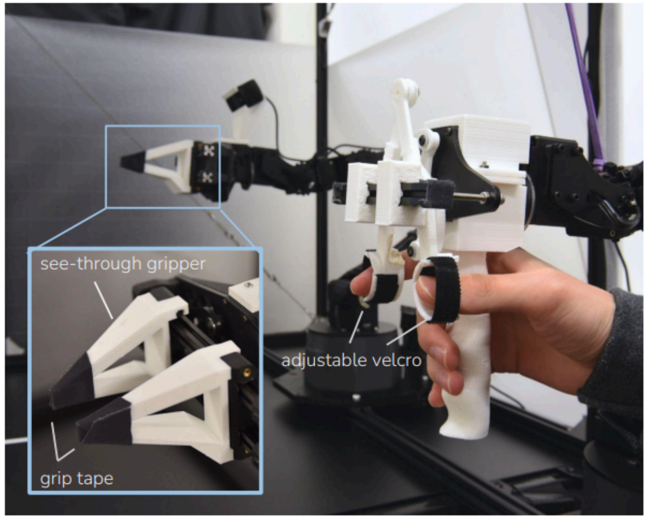
System Design - Design principles - Teleoperation setup

- Joint-space mapping for control
- High-frequency control (50Hz)
- 3D printed “see-through” fingers, gripping tape

Joint space mapping for control

- Directly maps "Leader" joint angles to "Follower" joint angles
- Solves IK failing problem





Imitation Learning Algorithm

- **Challenges**:
  - Compounding errors in policy
  - Non-stationary human demonstrations

- **Solution: Action Chunking with Transformers (ACT)**:
  - Predicts sequences of actions (chunks)
  - Reduces effective horizon of tasks
  - Uses temporal ensembling for smoothness

Training ACT on a New Task

- Record leader joint positions as actions
- Observations: follower joint positions, 4 camera feeds
- Train ACT: predict future actions from observations
- Test: use policy with lowest validation loss
- **Challenge: Compounding errors**

Action Chunking

- **Groups individual actions into units** for efficient storage and execution

- Reduces the **effective horizon** of long trajectories

- Every k steps, the agent receives an observation and generates k actions

- Mitigates issues with non-stationary demonstrations

Temporal Ensemble

- Creates **overlapping action chunks**
- Queries the policy at **every step** for precise and smoother motions
- **Combines** predictions using a weighted average
- No additional training cost, only extra inference-time computation

Modeling Human Data

- Human demonstrations are **noisy and inconsistent**

Modeling Human Data

- Human demonstrations are **noisy and inconsistent**

- **Different** trajectories can be used for the same observation

Modeling Human Data

- Human demonstrations are **noisy and inconsistent**
- **Different** trajectories can be used for the same observation
- Human actions are more **stochastic** where precision matters less

Modeling Human Data

- Human demonstrations are **noisy and inconsistent**
- **Different** trajectories can be used for the same observation
- Human actions are more **stochastic** where precision matters less
- Policy must **focus on high precision** areas

Conditional Variational Autoencoder (CVAE)

- Train action chunking policy as a generative model

Conditional Variational Autoencoder (CVAE)

- Train action chunking policy as a generative model

- Only decoder (policy) used in deployment

Conditional Variational Autoencoder (CVAE)

- Train action chunking policy as a generative model
- Only decoder (policy) used in deployment
- Maximize log-likelihood of demonstration action chunks

Implementation of ACT: Encoder

- CVAE encoder and decoder implemented with transformers
- BERT-like transformer encoder used
- Inputs: current joint positions and target action sequence
- Outputs: mean and variance of "style variable" z
- Encoder only used during training

Implementation of ACT: Decoder

- Predicts next k actions
- Inputs: current observations and z
- Observations: 4 RGB images and joint positions of 2 robot arms
- ResNet18 used for image processing

Implementation of ACT: Decoder

- Transformer encoder synthesizes information
- Transformer decoder generates action sequence
- L1 loss used for precise action sequence modeling

Experiment Tasks

- Slide Ziploc: Grasp and open ziploc bag slider
- Slot Battery: Insert battery into remote controller slot
- Open Cup: Open lid of small condiment cup
- Thread Velcro: Insert velcro cable tie into loop
- Prep Tape: Cut and hang tape on box edge
- Put On Shoe: Put shoe on mannequin foot and secure velcro strap
- Transfer Cube (sim): Transfer red cube to other arm
- Bimanual Insertion (sim): Insert peg into socket in mid-air

Challenges

- Requires ine-grained bimanual control
- Perception challenges (e.g., transparency, low contrast)
- Random initial placement of objects
- Need for visual feedback to correct perturbations
- Precise manipulation needed

Data Collection

- Collected demonstrations using ALOHA teleoperation for 6 real-world tasks
- Each episode: 8-14 seconds (400-700 time steps at 50Hz)
- 50 demonstrations per task (100 for Thread Velcro)
- Total: 10-20 minutes of data per task

- Two types of demonstrations for simulated tasks: scripted policy and human demonstrations

- Human demonstrations are stochastic:
  - Mid-air handover example: position varies each time
  - Policy must learn dynamic adjustments, not memorization

Experiment Comparison

- Compared ACT with four methods:
  - BC-ConvMLP: Simple baseline with convolutional network
  - BeT: Uses Transformers, no action chunking, separate visual encoder
  - RT-1: Transformer-based, predicts one action from history
  - VINN: Non-parametric, uses k-nearest neighbors

- ACT directly predicts continuous actions

Experiment Results

- ACT outperforms all prior methods in both simulated and real tasks
- Simulated tasks: ACT shows 20%-59% higher success rates
- Real-world tasks: Slide Ziploc (88%), Slot Battery (96%)
- ACT's performance in Thread Velcro was lower (20%) due to precision challenges

Experimental Results - Performance

- Success rates of 80-90%

- Comparison with baselines:
  - ACT significantly outperforms other methods
  - Effective in both simulated and real-world tasks

Conclusion and Future Work

- **Conclusion**:
  - Developed a low-cost, effective system for fine manipulation
  - Proposed a novel imitation learning algorithm (ACT)

- **Future Work**:
  - Improving generalization to new tasks
  - Enhancing hardware precision
  - Exploring more complex manipulation tasks