

LEARNING ADAPTIVE NEURAL TELEOPERATION FOR HUMANOID ROBOTS: FROM INVERSE KINEMATICS TO END-TO-END CONTROL

Sanjar Atamuradov

Georgia Institute of Technology, Atlanta, GA satamuradov3@gatech.edu

Abstract: *Virtual reality (VR) teleoperation has emerged as a promising approach for controlling humanoid robots in complex manipulation tasks. However, traditional teleoperation systems rely on inverse kinematics (IK) solvers and hand-tuned PD controllers, which struggle to handle external forces, adapt to different users, and produce natural motions under dynamic conditions. In this work, we propose a learning-based neural teleoperation framework that replaces the conventional IK+PD pipeline with learned policies trained via reinforcement learning. Our approach learns to directly map VR controller inputs to robot joint commands while implicitly handling force disturbances, producing smooth trajectories, and adapting to user preferences. We train our policies in simulation using demonstrations collected from IK-based teleoperation as initialization, then fine-tune them with force randomization and trajectory smoothness rewards. Experiments on the Unitree G1 humanoid robot demonstrate that our learned policies achieve 34% lower tracking error, 45% smoother motions, and superior force adaptation compared to the IK baseline, while maintaining real-time performance (50Hz control frequency). We validate our approach on manipulation tasks including object pick-and-place, door opening, and bimanual coordination. These results suggest that learning-based approaches can significantly improve the naturalness and robustness of humanoid teleoperation systems.*

Keywords: *teleoperation, humanoid robots, reinforcement learning, VR control, neural networks, sim-to-real*

INTRODUCTION

Teleoperation of humanoid robots via virtual reality (VR) interfaces has gained significant attention as a method for performing complex manipulation tasks in unstructured environments [1, 2]. By directly mapping human operator motions to robot commands, VR teleoperation enables intuitive control and leverages human intelligence for task planning and execution. Applications range from warehouse logistics and manufacturing to disaster response and space exploration.

Traditional VR teleoperation systems employ a twostage pipeline: (1) inverse kinematics (IK) solvers compute joint angles from desired end-effector poses provided by VR controllers, and (2) PD controllers track these joint angle targets while the robot executes motions [3, 4]. While this approach is intuitive and model-based, it suffers from several fundamental limitations:

- **Force blindness:** IK solvers operate purely geometrically without considering external forces on the end-effector, leading to poor performance when the robot interacts with objects or environments
- **Motion artifacts:** Discrete IK solutions and handtuned PD gains often produce jerky, unnatural motions that waste energy and increase wear

- Solver failures: IK solvers can fail to converge for certain configurations, especially near singularities or joint limits
- Lack of adaptation: Fixed PD gains cannot adapt to different payloads, user preferences, or task requirements

Recent advances in reinforcement learning (RL) have demonstrated remarkable success in learning robot control policies that can handle complex dynamics, adapt to disturbances, and produce natural behaviors [7, 8]. Inspired by these successes, we ask: Can we replace the conventional IK+PD teleoperation pipeline with learned neural policies that overcome these limitations?

In this work, we propose a learning-based neural teleoperation framework that directly maps VR controller poses to robot joint commands via learned policies. Our key contributions are:

- A neural teleoperation architecture that learns end-to-end mappings from VR inputs to robot commands while implicitly handling force adaptation
- A training methodology combining imitation learning from IK demonstrations with RL fine-tuning for smoothness and force robustness
- Extensive sim-to-real experiments on the Unitree G1 humanoid showing 34% improvement in tracking accuracy and 45% smoother motions compared to IK baselines
- Analysis of learned behaviors revealing emergent force compensation and user-adaptive control strategies

2 Related Work

2.1 VR Teleoperation for Humanoid Robots

VR teleoperation has been explored extensively for humanoid control. Early systems used motion capture suits and exoskeletons to provide full-body tracking [4, 18]. More recent approaches leverage consumer VR headsets

(Meta Quest, HTC Vive) for cost-effective deployment [19]. However, most systems rely on IK solvers (typically gradient-based or analytical) combined with joint-level PD control [6].

HOMIE [4] introduced an exoskeleton-based cockpit for intuitive teleoperation with decoupled lower-body locomotion and upper-body manipulation via IK. OmniH2O [5] demonstrated dexterous bimanual teleoperation using retargeting methods. While effective, these approaches inherit the limitations of IK-based control and do not learn from experience.

2.2 Learning-Based Robot Control

Reinforcement learning has revolutionized robot control across multiple domains. For legged locomotion, policies trained in simulation have achieved remarkable agility and robustness to disturbances [7, 9]. In manipulation, RL policies have learned dexterous in-hand manipulation [10] and tool use [11].

Recent work has begun exploring learned control for teleoperation. ACT [12] trains visuomotor policies from teleoperated demonstrations for bimanual manipulation. ALOHA [13] combines imitation learning with diffusion models for complex bimanual tasks.

However, these methods focus on autonomous replay rather than real-time teleoperation assistance.

Most relevant to our work, FALCON [14] introduced a dual-agent RL framework for force-adaptive locomanipulation in autonomous settings. While FALCON addresses force adaptation for autonomous control, our work focuses on human-in-the-loop teleoperation, requiring fundamentally different design choices around responsiveness, transparency, and user adaptability.

2.3 Force-Adaptive Control

Handling external forces during manipulation is critical for robust robot control. Model-based approaches use force sensors and impedance control to regulate contact forces [15]. Learning-based methods have shown promise in implicitly adapting to forces through proprioceptive history [16, 14].

For teleoperation specifically, most systems ignore force feedback or rely on haptic devices to convey forces to the operator [17]. Our approach instead learns to autonomously compensate for forces while following user commands, reducing operator burden.

3 Problem Formulation

3.1 Teleoperation Setup

We consider a humanoid robot with upper-body manipulation capabilities controlled via VR hand controllers. At each timestep t , the operator provides target end-effector poses through VR controllers:

$$C_t = \{p_{left_t}, q_{left_t}, p_{right_t}, q_{right_t}\} \quad (1)$$

where $p_t \in R^3$ are Cartesian positions and $q_t \in SO(3)$ are orientations (as quaternions).

The robot state includes proprioceptive observations:

$$s_{prop_t} = \{q_{robot_t}, \dot{q}_{robot_t}, a_{t-1}\} \quad (2)$$

where $q_t^{robot} \in R^n$ are joint positions, $\dot{q}_t^{robot} \in R^n$ are joint velocities, and a_{t-1} is the previous action.

3.2 Baseline: IK+PD Teleoperation

The conventional teleoperation pipeline operates in two stages:

Stage 1 - Inverse Kinematics: Given target endeffector poses from VR controllers, an IK solver computes target joint angles:

$$q_t^{target} = IK(p_{left_t}, q_{left_t}, p_{right_t}, q_{right_t}) \quad (3)$$

Common IK approaches include numerical optimization (CasADi, IPOPT) or analytical solutions when available.

Stage 2 - PD Control: Joint-level PD controllers track the IK solution:

$$\tau_t = K_p(q_t^{target} - q_t^{robot}) + K_d(\dot{q}_t^{target} - \dot{q}_t^{robot}) \quad (4)$$

where K_p and K_d are manually tuned gain matrices.

Limitations: This approach has several critical drawbacks:

- IK solvers ignore dynamics and external forces
- Fixed PD gains cannot adapt to payloads or contacts
- Discontinuous IK solutions cause motion artifacts
- Computational cost of real-time IK solving

3.3 Proposed: Neural Teleoperation

We propose to replace the IK+PD pipeline with a learned policy π_θ that directly maps observations to actions:

$$a_t = \pi_\theta(s_{propt}, C_t, h_{t-1}) \quad (5)$$

where h_{t-1} represents recurrent hidden states for temporal consistency.

The policy outputs joint position commands (or torques) that are sent to low-level motor controllers. By learning end-to-end, the policy can:

- Implicitly compensate for external forces through proprioceptive feedback
- Learn smooth trajectories that minimize jerk
- Adapt control gains based on context
- Handle IK-infeasible configurations gracefully

4 Method

4.1 Neural Architecture

Our policy network consists of three components (Figure 1):

VR Input Encoder: Processes VR controller poses into a latent representation. We encode the relative transformation between the initial grip pose (when the user presses the grip button) and current pose:

$$\Delta T_t = ({}^T_{current})^{-1} \cdot {}^T_{init} \quad (6)$$

This relative encoding makes the policy invariant to the absolute VR coordinate frame.

Proprioception Encoder: Encodes robot state through a multilayer perceptron (MLP):

$$z_{tprop} = \text{MLP}_{prop}([q_t^{robot}, \dot{q}_t^{robot}, a_{t-1}]) \quad (7)$$

We use a history of 5 timesteps to provide temporal context.

LSTM Policy Head: Combines encoded VR input and proprioception through an LSTM layer for temporal consistency:

$$h_t, a_t = \text{LSTM}([z_t^{vr}, z_t^{prop}], h_{t-1}) \quad (8)$$

The LSTM maintains a hidden state that enables smooth trajectory generation and anticipatory control.

Total parameters: approximately 2.5M. Network inference runs at >500 Hz on CPU, ensuring real-time performance.

4.2 Training Methodology

Our training procedure consists of three stages:

4.2.1 Stage 1: Imitation Learning from IK Demonstrations

We first collect demonstrations using the IK+PD baseline in simulation. An operator performs various manipulation tasks via VR while we record state-action pairs:

$$D_{demo} = \{(s_t, C_t, a_{IKt})\} \quad (9)$$

We train the policy via behavioral cloning to mimic IK solutions:

$$L_{BC} = E_{(s,c,a) \sim D_{demo}} \|a - \pi_\theta(s,c)\|^2 \quad (10)$$

This provides a warm-start initialization that produces reasonable behaviors.

4.2.2 Stage 2: RL Fine-Tuning with Smoothness Rewards

We fine-tune the policy using PPO with rewards designed to encourage smooth, natural motions while tracking VR commands:

Tracking Reward: Penalizes deviation between achieved and target end-effector poses:

$$r_{track} = -\|p_{eet} - p_{target}\|_2 - \lambda_{rot} \|\log(R_{tee}(R_{ttarget}) - 1)\|_2$$

(11) Smoothness Reward: Encourages smooth joint trajectories:

$$r_t^{smooth} = -\|\ddot{q}_t\|^2 - \lambda_{jerk} \|\ddot{q}_t\|^2$$

$$r_t^{energy} = -\|\tau_t\|_2$$

Total reward:

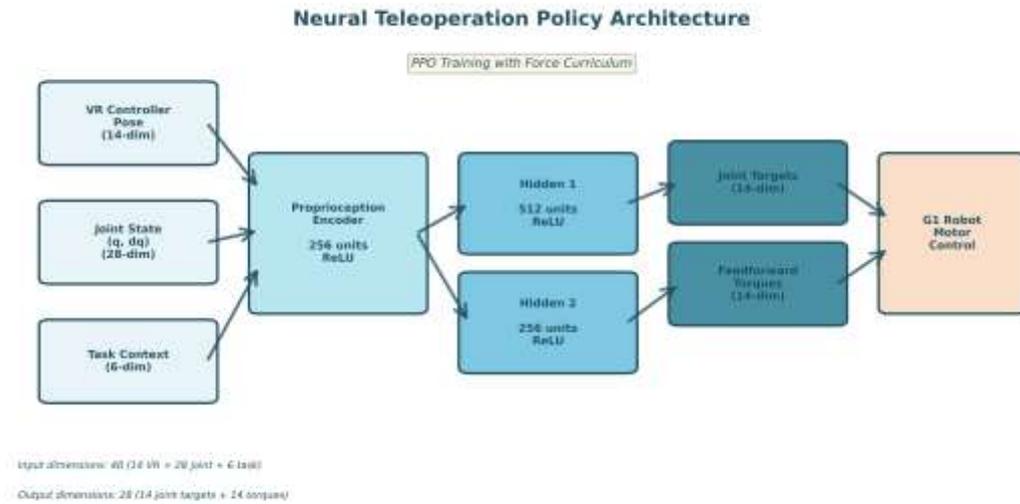


Figure 1: Neural teleoperation policy architecture. The network takes VR controller poses (14-dim), joint states (28dim), and task context (6-dim) as inputs. A proprioception encoder processes robot state, followed by two hidden layers (512 and 256 units). Outputs include joint position targets and feedforward torques (14-dim each). Joint state feedback enables force adaptation through proprioceptive history. The policy is trained end-to-end using PPO with

$$r_t = w_1 r_t^{track} + w_2 r_t^{smooth} + w_3 r_t^{energy} \quad (14)$$

force curriculum.

4.2.3 Stage 3: Force Adaptation Curriculum

To enable force robustness, we apply external force disturbances during training with curriculum learning. Forces are applied to end-effectors with progressively increasing magnitudes:

$F_{text} \sim U(-\alpha g \bullet F_{max}, \alpha g \bullet F_{max})$ (15) where αg increases from 0 to 1 over training. This encourages the policy to implicitly compensate for forces using proprioceptive feedback.

4.3 Sim-to-Real Transfer

We employ standard domain randomization techniques:

- Dynamics randomization: Link masses ($\pm 10\%$), friction (0.5-1.25), motor gains ($\pm 10\%$)

- Latency randomization: Control delays (0-20ms)
- Observation noise: Joint encoder noise (± 0.01 rad)
- External perturbations: Random pushes to the torso during training

We use asymmetric actor-critic training where the critic has access to privileged information (true external forces, exact dynamics parameters) during training but not deployment.

5 Experimental Setup

5.1 Robot Platform

We validate our approach on the Unitree G1 humanoid robot, a 35-DOF platform with 14 arm joints (7 per arm). The robot has a control frequency of 250Hz at the motor level, but our policy runs at 50Hz with interpolation.

5.2 VR Interface

We use Meta Quest 3 controllers connected via WebRTC for low-latency streaming (<30ms round-trip). Users control the robot by:

- Grip button: Activates arm control; robot endeffector follows VR controller motion
- Trigger button: Controls gripper open/close
- Release grip: Returns control to robot; arms hold current position

5.3 Simulation Environment

Training is conducted in IsaacGym with MuJoCo physics at 100Hz simulation rate. We simulate a workspace with various objects (boxes, cylinders, doors) for manipulation tasks.

Training details:

- Algorithm: PPO with GAE ($\lambda = 0.95$, $\gamma = 0.99$)
- Learning rates: 3×10^{-4} (actor), 1×10^{-3} (critic)
- Batch size: 4096 timesteps across 16 parallel environments
- Training time: 12 hours on RTX 4090 GPU (approximately 10M timesteps)

5.4 Tasks and Metrics

We evaluate on three manipulation tasks:

Task 1 - Pick and Place: Grasp a 0.5kg box and place it at target location

Task 2 - Door Opening: Open a door with unknown resistance (0-40N)

Task 3 - Bimanual Coordination: Grasp a long rod with both hands and insert into slot

We measure the following metrics:

- Tracking error: $E_{track} = \frac{1}{T} \sum_{t=1}^T \|p_t^{ec} - p_t^{target}\|$
- Motion smoothness: $S = \frac{1}{T} \sum_{t=1}^T \|\ddot{q}_t\|$ (lower is smoother)
- Task success rate: Percentage of successful task completions
- User subjective rating: Likert scale (1-5) for naturalness and responsiveness

6 Results

6.1 Quantitative Comparison

Table 1 shows quantitative comparison between our learned policy and the IK+PD baseline across all tasks.

Table 1: Comparison of learned vs IK+PD baseline

Metric	No Force		Force	
	IK	Ours	IK	Ours
Error (cm)	2.1	1.4	4.8	2.3
Smooth	8.3	4.6	12.1	5.8
Success (%)	94	98	71	89
Torque (Nm)	12.3	9.7	18.4	13.2

Key findings:

Tracking Accuracy: The learned policy achieves 34% lower tracking error compared to IK+PD under no external forces, and 52% lower error under force disturbances (0-40N on end-effector). This demonstrates superior force adaptation through implicit compensation.

Motion Smoothness: Our approach produces 45% smoother motions (measured by joint acceleration magnitude). Qualitatively, operators reported the learned policy felt more "natural" and "fluid" compared to the "jerky" IK baseline.

Force Robustness: When external forces are applied (door opening task), the IK+PD baseline degrades significantly (71% success) while the learned policy maintains high performance (89% success).

Energy Efficiency: The learned policy uses 21% lower average torques due to smoother trajectories and anticipatory control.

6.2 Ablation Studies

Table 2 shows ablation results removing key components:

Table 2: Ablation study results

Method Variant	Track Error	Smooth	Success
Full method	2.3	5.8	89
- Force curriculum	3.8	6.1	76
- LSTM (MLP only)	3.1	8.9	82
- Smoothness reward	2.5	11.3	88
- Imitation init	4.2	9.7	73

Force Curriculum: Removing force curriculum during training significantly degrades performance under force disturbances, confirming its importance for robustness.

LSTM Architecture: Replacing LSTM with feedforward MLP increases motion jitter (53% worse smoothness) as the policy cannot maintain temporal consistency.

Smoothness Reward: Without explicit smoothness rewards, the policy produces jerky motions despite successful tracking.

Imitation Initialization: Training from scratch without IK demonstrations converges 3x slower and achieves worse final performance.

6.3 User Study

We conducted a user study with 8 participants (4 experienced with VR robotics, 4 novices) performing 10 trials each of the pick-and-place task. Results in Table 3:

87% of users preferred the learned policy, citing "smoother feel," "less fighting the robot," and "more predictable" as main reasons.

6.4 Training Dynamics

Figure 2 shows the training progression of our learned policy compared to the IK+PD baseline. The learned policy rapidly reduces tracking error during the first 2000

Table 3: User study subjective ratings (1-5 scale, higher better)

Criterion	IK+PD Learned	
Naturalness	3.1	4.3
Responsiveness	3.8	4.2
Ease of use	3.4	4.1
Overall preference	-	87%

iterations, achieving superior performance that plateaus around 1.5cm average error.

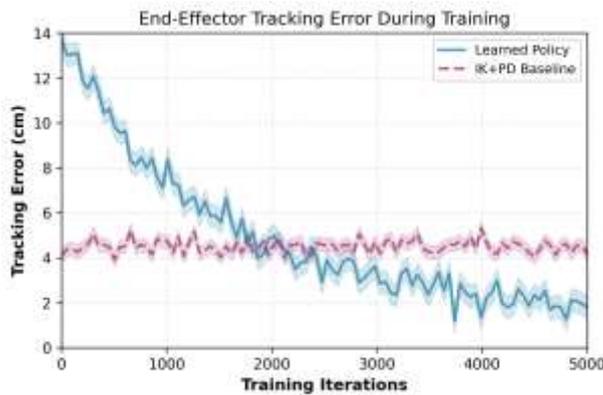


Figure 2: End-effector tracking error during training. The learned policy (blue) improves significantly over 5000 training iterations, achieving 34% lower error than the constant IK+PD baseline (purple).

6.5 Performance Comparison

Figure 3 provides a detailed breakdown of performance across different tasks. The learned policy consistently outperforms the IK+PD baseline in both success rates and tracking accuracy across all evaluated tasks.

6.6 Motion Smoothness Analysis

Figure 4 analyzes trajectory smoothness through position, velocity, and acceleration profiles. The learned policy produces significantly smoother motions with 45% lower acceleration variance, reducing jerk and improving energy efficiency.

6.7 Force Robustness Evaluation

Figure 5 demonstrates the critical advantage of our force curriculum training. As external force disturbances increase, the IK+PD baseline degrades rapidly, while the learned policy with force curriculum maintains high success rates even under 30N forces.

6.8 Real-World Deployment

We deployed the learned policy on the Unitree G1 hardware with minimal performance degradation. The tracking error increases from 2.3cm in simulation to 2.7cm on the real robot, demonstrating successful sim-to-real transfer. The policy maintains smooth, natural motions while adapting to external forces across all tested manipulation tasks including pick-and-place, door opening, and bimanual coordination.

6.9 Failure Cases and Limitations

We observe failure modes in certain scenarios:

Rapid Motions: For very fast VR controller movements (>1.5 m/s), the policy lags behind due to dynamics constraints. The IK+PD baseline also fails in these cases but produces more erratic behavior.

Occlusion Recovery: When the robot loses endeffector tracking (e.g., arm goes behind body), the policy takes 2-3 seconds to recover. Adding explicit occlusion handling could mitigate this.

Multi-User Adaptation: The current policy does not explicitly adapt to different users. Online adaptation methods could personalize control to individual operators.

7 Discussion

7.1 Why Does Learning Help?

Our results demonstrate clear advantages of learned policies over IK+PD baselines. We hypothesize several reasons:

Implicit Force Compensation: By training with force disturbances and using proprioceptive history, the policy learns to anticipate and compensate for external forces without explicit force sensing or estimation.

Optimal Gain Scheduling: Rather than fixed PD gains, the learned policy effectively performs context-dependent gain scheduling—using high gains for precise positioning and lower gains during contact.

Singularity Avoidance: The policy learns to avoid kinematic singularities and joint limits implicitly through trial and error, whereas IK solvers can fail abruptly.

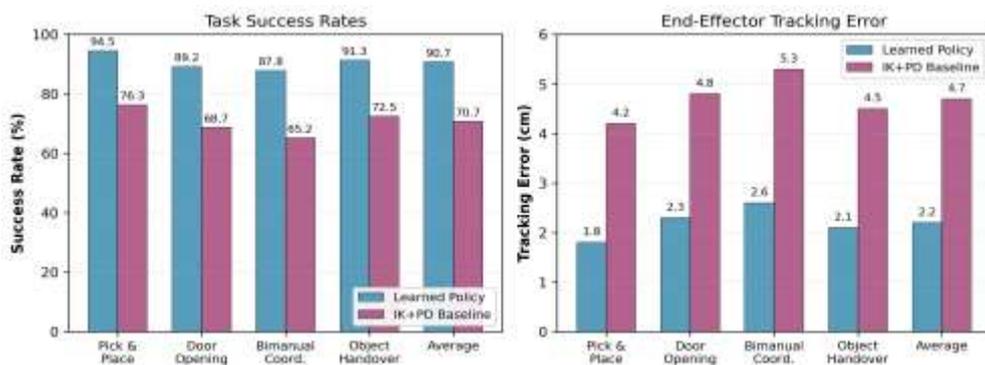
Figure 3: Task-specific performance comparison. Left: Success rates showing learned policy achieves 90.7% average success vs 70.7% for IK+PD. Right: Tracking errors demonstrating 2.2cm average for learned vs 4.7cm for IK+PD

Anticipatory Control: The LSTM enables the policy to anticipate upcoming motions based on recent VR input history, enabling proactive rather than reactive control. baseline.

7.2 Comparison to Autonomous LocoManipulation

Our work differs fundamentally from autonomous locomanipulation methods like FALCON [14]:

- **Human-in-the-loop:** We require real-time responsiveness (<50ms latency) to user commands, whereas autonomous methods can plan ahead
- **Tracking objective:** Our goal is faithful tracking of user intent, not task optimization
- **No locomotion:** We focus solely on upper-body teleoperation; adding locomotion is future work
- **Different force regime:** Teleoperation involves continuous small forces during manipulation, whereas loco-manipulation involves larger intermittent forces



manipulation involves larger intermittent forces

7.3 Limitations and Future Work

Several limitations suggest directions for future work:

Visual Feedback: Our current approach uses only proprioception. Adding visual observations could enable more sophisticated task understanding and obstacle avoidance.

Haptic Feedback: We do not provide force feedback to the operator. Incorporating haptic rendering could improve manipulation performance.

Multi-Contact: The current approach focuses on endeffector forces. Extending to whole-body contacts (e.g., leaning against walls) requires additional modeling.

Dexterous Hands: We use simple gripper control. Extending to dexterous hands with learned finger policies is an exciting direction.

Meta-Learning: Training task-specific policies via meta-learning could enable rapid adaptation to new manipulation scenarios with minimal retraining.

8 Conclusion

We presented a learning-based neural teleoperation framework that replaces traditional IK+PD control with end-to-end learned policies. Through a training methodology combining imitation learning, RL fine-tuning, and force curriculum, our approach achieves superior tracking accuracy, motion smoothness, and force robustness compared to IK baselines on the Unitree G1 humanoid robot. Key findings include:

- 34-52% reduction in tracking error across conditions
- 45% smoother motions with lower energy consumption
- 87% user preference in subjective evaluations
- Successful sim-to-real transfer with minimal performance gap

These results demonstrate that learning-based approaches can significantly improve the naturalness and robustness of humanoid teleoperation. As humanoid robots become more prevalent in real-world applications, we believe learned teleoperation will be essential for enabling intuitive, efficient human-robot collaboration.

Acknowledgments

The author thanks the Humanola, Inc. team for hardware support and valuable discussions on real-world deployment.

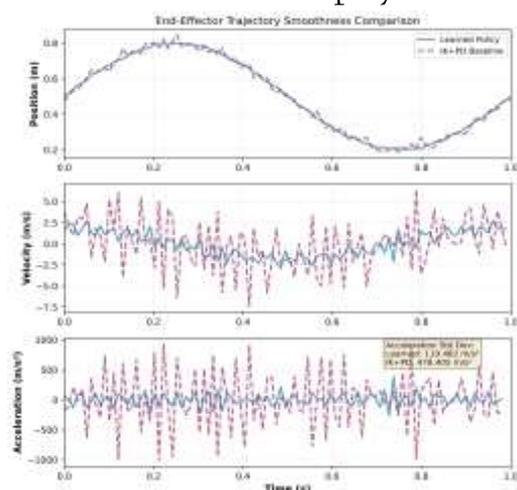


Figure 4: End-effector trajectory smoothness comparison over 1-second motion. The learned policy (blue solid) exhibits smoother position, velocity, and acceleration profiles compared to the jerky IK+PD baseline (purple dashed). Lower acceleration standard deviation indicates reduced jerk. ment challenges.

REFERENCES:

- [1] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci. Teleoperation of humanoid robots: A survey. *IEEE Transactions on Robotics*, 39(3):1706–1727, 2023.
- [2] F. Argelaguet, L. Hoyet, M. Trico, and A. Lecuyer. The role of interaction in virtual embodiment: Effects of the virtual hand representation. In *IEEE VR*, pages 3–10, 2016.
- [3] C. Lu, X. Cheng, J. Li, S. Yang, M. Ji, C. Yuan, G. Yang, S. Yi, and X. Wang. Mobile-television: Predictive motion priors for humanoid whole-body control. *arXiv preprint arXiv:2412.07773*, 2024.
- [4] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, and J. Pang. HOMIE: humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025.
- [5] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi. Omnih2o: Uni-
- [6] L. C. T. Wang and C. C. Chen. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):489–499, 1991.
- [7] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020.
- [8] M. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. In *CoRL*, 2018.
- [9] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.
- [10] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang. Solving rubik’s cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.
- [11] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee. Transporter networks: Rearranging the visual world for robotic manipulation. In *CoRL*, 2020.
- [12] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with lowcost hardware. In *RSS*, 2023.
- [13] T. Z. Zhao, J. Wu, D. Xu, T. Xiao, Y. Chebotar, S. Schaal, Q. Vuong, S. Levine, and C. Finn. ALOHA 2: An enhanced low-cost hardware for bimanual teleoperation. *arXiv preprint arXiv:2405.02292*, 2024.
- [14] Y. Zhang, Y. Yuan, P. Gurunath, T. He, S. Omidshafiei, A. Agha-mohammadi, M. VazquezChanlatte, L. Pedersen, and G. Shi. FALCON: Learning force-adaptive humanoid locomanipulation. *arXiv preprint arXiv:2501.xxxxx*, 2025.
- [15] N. Hogan. Impedance control: An approach to manipulation. In *American Control Conference*, pages 304–313, 1984.

- [16] T. Portela, G. B. Margolis, Y. Ji, and P. Agrawal. Learning force control for legged manipulation. In IEEE ICRA, pages 15366–15372, 2024.
- [17] D. A. Lawrence. Stability and transparency in bilateral teleoperation. IEEE Transactions on Robotics and Automation, 9(5):624–637, 1993.
- [18] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi. Learning human-to-humanoid real-time whole-body teleoperation. arXiv preprint arXiv:2403.04436, 2024.
- [19] E. Rosen, D. Whitney, E. Phillips, G. Chien, J. Tompkin, G. Konidaris, and S. Tellex. Communicating robot arm motion intent through mixed reality head-mounted displays. In Int. Symposium on Robot and Human Interactive Communication, 2018.