# A Preliminaries

## A.1 Formulation

We formulate the humanoid teloperation as a Markov Decision Process (MDP) $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}\}$. $\mathcal{S}$ includes proprioceptive states $s$ and task-oriented observations $o^{task}$. The action space $\mathcal{A} \in \mathcal{R}^{29}$ represents the humanoid's joint angles in our method. $\mathcal{T}$ is the transition function conditioned on $\mathcal{S}$ and $\mathcal{A}$. The reward functions $\mathcal{R}$ is defined based on $\mathcal{S}, \mathcal{A}$. A policy $\pi$ is proposed to maximize the overall reward $\mathcal{R}$ using the Proximal Policy Optimization (PPO) algorithm.

## A.2 LiDAR Odometry and Closed-Loop Error Correction

LiDAR odometry is designed to accurately determine the robot's state, including its orientation and position. In this paper, we adopt FAST-LIO2 [27], which utilizes onboard LiDAR and IMU to estimate the humanoid's global position. FAST-LIO2 [27] leverages IMU data and LiDAR point clouds to construct and update a 3D map in real time. It then registers the current LiDAR point clouds with the map to estimate the robot's current position.

Previous teleoperation systems [18, 22, 23] often operate in an open-loop manner, primarily due to the absence of the humanoid's global position. Consequently, stepwise tracking errors accumulate over time, leading to significant drift during long-horizon tasks. In this work, we leverage LiDAR odometry to determine the robot's global position. Similarly, we obtain the operator's global position with the odometry from Apple Vision Pro. We integrate the difference between the two positions into the task observation $o^{task}$, and design a reward function for our teleoperation policy that minimizes this difference. Notably, the LiDAR operates at 10 Hz, and our policy runs at 50 Hz. Our policy uses the latest available odometry position at each timestep.

# B Reward Functions and Domain Randomization

Tab. A1 provides a detailed overview of the reward structure utilized in this study, while Tab. A2 outlines the domain randomization scheme employed.

# C Implement Details

## C.1 Data Augmentation

In our implementation, we filter out physically infeasible AMASS data and select motions with large upper- and lower-body workspaces as oracle motions. These motions are further modified by concatenating body parts or accelerating sequences.

## C.2 Model Architecture

The student policy is composed of $L = 3$ MoE layers, each containing $N = 4$ experts, where each expert is implemented as an MLP with dimensions (2048, 512, 512, 256). The policy uses a history length of $H = 25$ frames and activates the top $k = 2$ experts based on the highest weights determined by the router. The AMP discriminator is a 3-layer MLP (256, 256, 256) that is updated online during training on **CLONED**. For comparison, the baseline model **CLONE** $^\dagger$ uses a single MLP with architecture (2048, 1024, 512, 512).

## C.3 Policy Training

We train our policy in IsaacGym using a single A800 GPU. The teacher policy is trained for 1M iterations with 8192 parallel environments, while the student policy is trained for $600K$ iterations with 4096 parallel environments. Training the teacher policy requires $\sim$ 480K simulation steps

Table A1: **Reward functions.** The details of the primary reward function used in our training process.

| Term | Expression | Weight |
|---|---|---|
| Torque | $\|\boldsymbol{\tau}\|_2^2$ | $-0.0001$ |
| Torque limits | $[\tau \notin [\tau_{\text{min}}, \tau_{\text{max}}]]_1$ | -2 |
| DoF position limits | $[\mathbf{p}_t \notin [\mathbf{p}_{\text{min}}, \mathbf{p}_{\text{max}}]]_1$ | -625 |
| DoF velocity limits | $[\dot{\mathbf{p}}_t \notin [\dot{\mathbf{p}}_{\text{min}}, \dot{\mathbf{p}}_{\text{max}}]]_1$ | -50 |
| Termination | $\text{termination}_1$ | $-e^4$ |
| DoF acceleration | $\|\ddot{\mathbf{q}}_{\mathbf{t}}\|_2^2$ | $-1.1e^{-5}$ |
| DoF velocity | $\|\dot{\mathbf{q}}_{\mathbf{t}}\|_2^2$ | $-0.004$ |
| Lower-body action rate | $\|\mathbf{a}_t^{\text{lower}} - \mathbf{a}_{t-1}^{\text{lower}}\|_2^2$ | $-1.0$ |
| Upper-body action rate | $\|\mathbf{a}_t^{\text{upper}} - \mathbf{a}_{t-1}^{\text{upper}}\|_2^2$ | $-0.3$ |
| Feet air time | $T_{\text{air}} - 0.3$ | 2500 |
| Stumble | $[(\mathbf{F}_{\text{feet}}^{xy} > 5 \times \mathbf{F}_{\text{feet}}^{z})]_1$ | $-1.25e^4$ |
| Slippage | $\|v_t^{\text{feet}}\|_2^2 \cdot [(\mathbf{F}_{\text{feet}} \geq 1)]_1$ | $-80$ |
| Feet orientation | $\|\mathbf{R}_z^{\text{feet}}\|$ | $-62.5$ |
| In the air | $[(\mathbf{F}_{\text{feet}}^{\text{left}}, \mathbf{F}_{\text{feet}}^{\text{right}} < 1)]_1$ | $-750$ |
| Orientation | $\|\mathbf{R}_z^{\text{root}}\|$ | $-50$ |
| DoF position | $\exp(-0.25\|\hat{\mathbf{p}} - \mathbf{p}\|_2)$ | 100 |
| DoF velocity | $\exp\Big(-0.25\|\hat{\dot{\mathbf{p}}} - \dot{\mathbf{p}}\|_2^2\Big)$ | 10 |
| Extend body position | $\exp\big(-0.5\|\hat{\mathbf{q}} - \mathbf{q}\|_2^2\big)$ | 250 |
| Body position (MR) | $\exp\big(-0.5\|\mathbf{q}_{\text{vr}} - \hat{\mathbf{q}}_{\text{vr}}\|_2^2\big)$ | 150 |
| Body rotation | $\exp\Big(-0.1\|\theta \ominus \hat{\theta}\|\Big)$ | 400 |
| Body velocity | $\exp(-10.0\|\mathbf{v} - \hat{\mathbf{v}}\|_2)$ | 80 |
| Body angular velocity | $\exp(-0.01\|\boldsymbol{\omega} - \hat{\boldsymbol{\omega}}\|_2)$ | 8 |
| Body hand rotation | $(\theta_{\text{hand}} - \hat{\theta}_{\text{hand}})^2$ | 500 |
| AMP | Sec. 3.3 | 500 |

Table A2: **Domain Randomization.** The details of the primary domain randomization used in our training process.

| Term | Value |
|---|---|
| Friction | $\mathcal{U}(0.6, 2.0)$ |
| Base CoM offset | $\mathcal{U}(-0.04, 0.04)$m |
| Link mass | $\mathcal{U}(0.7, 1.25) \times$ default kg |
| P Gain | $\mathcal{U}(0.85, 1.15) \times$ default |
| D Gain | $\mathcal{U}(0.85, 1.15) \times$ default |
| Torque RFI | $0.05 \times$ torque limit N $\cdot$ m |
| Control delay | $\mathcal{U}(0.0, 20)$ms |
| Global step delay | $\mathcal{U}(0.0, 80)$ms |
| Rand born distance | $\mathcal{U}(0.0, 2.0)$m |
| Rand heading degree | $\mathcal{U}(-20.0, 20.0)$degree |
| Push robot | interval $= 5s$, $v_{xy} = 1.5$m/s |
| Terrain type | flat, rough, low obstacles [18] |

($\sim$ 20K PPO steps) and $\sim$ 24 hours on a single A800 GPU. The student policy requires $\sim$ 48 hours on a single 3090 Ti.

# D Experiments

## D.1 Evaluation Metrics

We evaluated **CLONE** on motion tracking tasks from **CLONED** using five metrics: success rate **SR** (%), mean per-keybody position error (MPKPE) $E_{\text{mpkpe}}$ (mm), root-relative mean per-keybody position error (R-MPKPE) $E_{\text{r-mpkpe}}$ (mm), average joint velocity error $E_{\text{vel}}$ (mm/s), and hand orientation tracking error $E_{\text{hand}}$. Success rate (**SR**) represents the proportion of episodes where: (i) the robot maintains balance without falling, and (ii) the average per-keybody distance between the robot and reference motion remains below 1.5m across the three controlled joints. We defined the hand orientation tracking error as $E_{\text{hand}} = 1 - \langle \hat{\text{q}}, \text{q} \rangle^2$, where $\hat{\text{q}}$ and q represent the reference and robot hand quaternions.

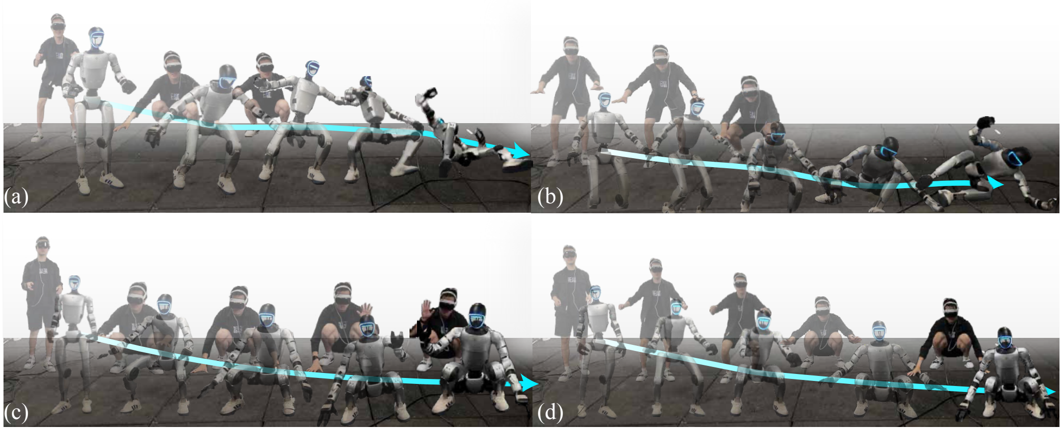

Figure A1: Qualitative Results of **CLONE** and **CLONE** *. (a) and (b) show the "crouch" tracking results of **CLONE** *, while (c) and (d) present the results of **CLONE**.

## D.2 Ablation Study

We investigated the impact of key design choices, specifically history length and MoE parameters, through systematic ablation experiments reported in Tab. A3. Our results indicate that a configuration using 25 timesteps of history, three MoE layers, and four experts per layer yields optimal performance across most evaluation metrics. We observed that shorter history lengths and increased expert counts can produce marginally lower R-MPKPE values and larger global tracking errors, suggesting a trade-off between local and global motion fidelity.

Table A3: Ablation study on history length and architecture components.

| **Method** | $E_{\textbf{mpkpe}}$ | $E_{\textbf{r-mpkpe}}$ | $E_{\textbf{vel}}$ | $E_{\textbf{hand-rot}}$ |
|---|---|---|---|---|
| **(a) History Length Analysis** | | | | |
| History5 | 93.97 | **31.99** | 236.12 | 3.80 |
| History50 | 135.60 | 41.33 | 286.66 | 12.23 |
| History25(CLONE) | **87.84** | 33.30 | **227.17** | **3.61** |
| **(b) Architecture Ablation** | | | | |
| **CLONE** ($L = 1$) | 134.06 | 37.56 | 270.14 | 7.22 |
| **CLONE** ($N = 8$) | 89.21 | **30.90** | 251.10 | 4.26 |
| **CLONE** | **87.84** | 33.30 | **227.17** | **3.61** |

## D.3 Qualitative Results Comparsion

We analyze the qualitative results of **CLONE** and **CLONE** * in Fig. A1. Subfigures (a) and (b) show that **CLONE** *, trained on OmniH2O [18], fails to track motions like "crouch" or "squat to pick up an object" and falls down. In contrast, subfigures (c) and (d) present the results of **CLONE**, which tracks these motions accurately and robustly. Although **CLONE** * is trained on a larger dataset (more than $8k$ motions, compared to **CLONED**'s 345 motions), it struggles with these tasks. Meanwhile, our model effectively tracks these motions and performs manipulation skills using only about 20% of the data. Since the OmniH2O [18] dataset also includes motions like "squat," this result suggests that a smaller dataset can still yield excellent tracking performance, as large-scale training data may cause the policy to overly generalize and compromise certain skills.

**Expert Activation Analysis** To better understand the specialization within our mixture-of-experts architecture, we visualized expert activation weights across nine distinct motion types in Fig. A2. Results reveal clear specialization patterns where motions requiring similar skills activate specific experts. In the first layer, experts 1 and 2 are predominantly activated during standing motions, while experts 3 and 4 show stronger activation during squatting motions. Notably, all four experts in the first layer become activated during dynamic motions such as jumping and punching, suggesting collaborative processing of complex movements. Similar specialization patterns emerged in subsequent layers, albeit with reduced variance across different motion categories.

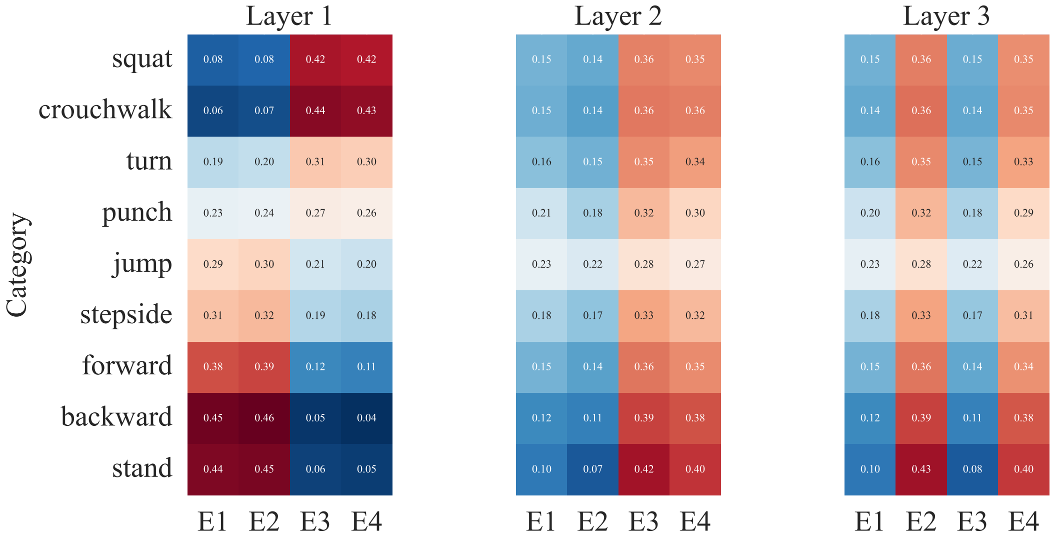

Figure A2: The activation status of each expert.

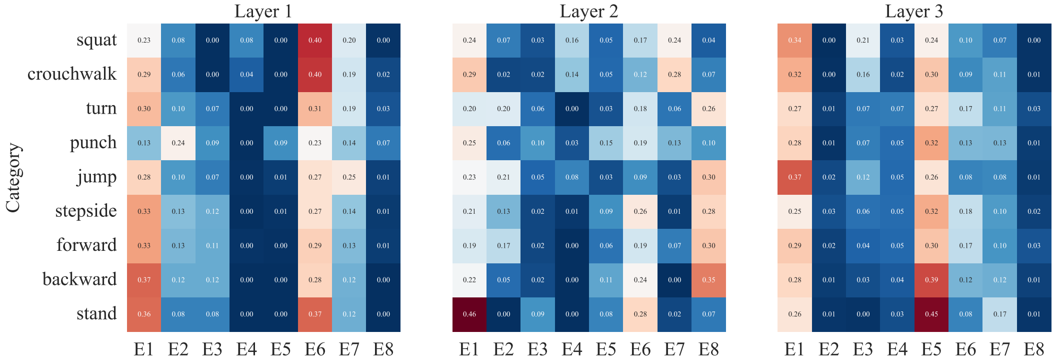

Figure A3: Experts activation when $N = 8$

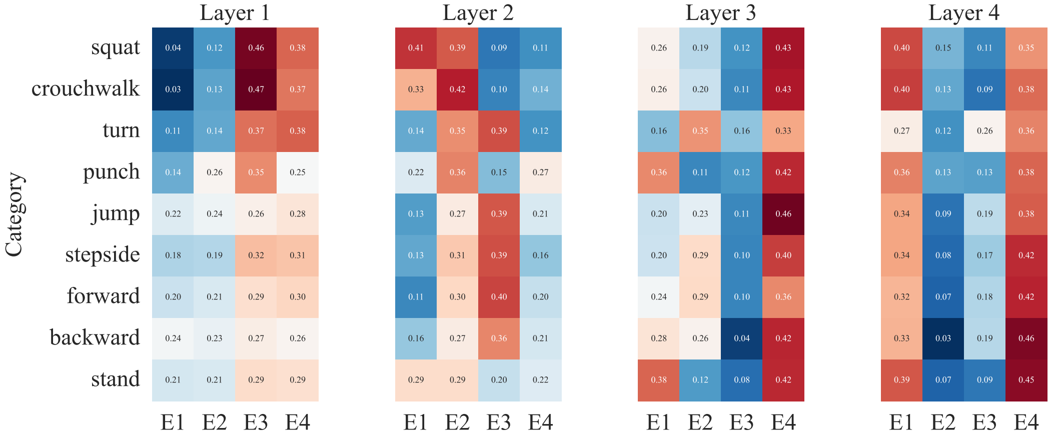

Figure A4: Experts activation when $L = 4$

### D.4 The Choice of the Number of MoE Layers and Number of Experts

We visualize the activation patterns of experts in Fig. A2 to A6. Fig. A3 shows that MoE layers with $N = 8$ experts activate only half of the experts in each layer, revealing that 8 experts are redundant for the current training data distribution, while 4 experts are sufficient. Fig. A6 demonstrates that **Clone** $^*(L = 1)$, which uses only one MoE layer, is still capable of activating different experts. However, as shown in Tab. A3, its tracking performance is inferior to that of **Clone**. This is

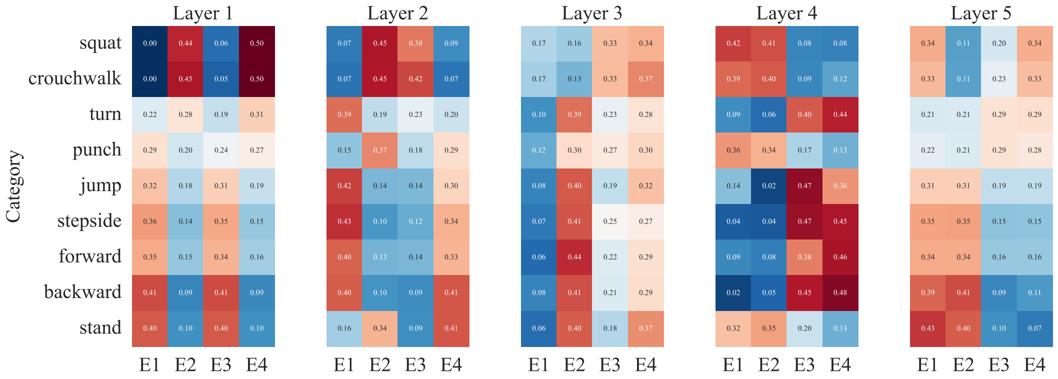

Figure A5: Experts activation when $L = 5$

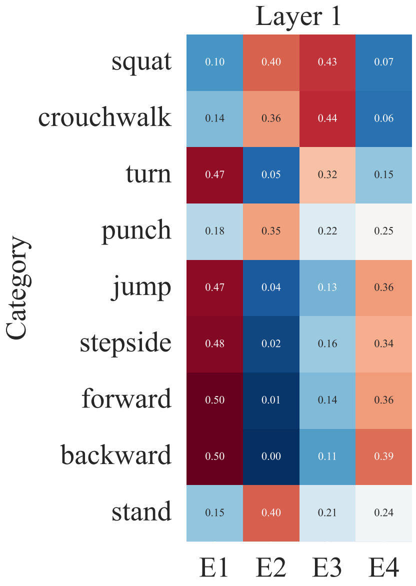

Figure A6: Experts activation when $L = 1$

primarily attributed to the model's parameters being too limited to effectively learn such diverse motions. Though 4 MoE layers and 5 MoE layers also has same activation patterns, like shown in Fig. A4 and A5, we choose 3 MoE layers for a balance of training cost and policy performance. Therefore, we select the MoE policy with three MoE layers and four experts as our final model.