A Implementation Details

Policy Architecture. We adopt the same network structure for the student policy and the teacher policy. The policies and the critics are all MLPs, consisting of a sequence of SiLU, linear layers, and layer-wise normalization layers, with hidden dimensions $[512, 512, 256, 128]$.

Goal Observations. The goal observation of the teacher policy consists of: 1) the root height, 2) the root rotation, 3) the root velocity, 4) the root yaw angle's angular velocity, 5) the joint positions, 6) the key body positions. We then stack 20 frames of future reference motions that span 2 seconds as the goal observations of the teacher policy. The goal observation of the student policy differs in two parts: 1) not using the key body positions to save time to compute forward kinematics, and 2) not using the future reference motions but only using the single frame.

Proprio Observations. The proprioception of the teacher policy consists of: 1) the root angular velocity, 2) the root rotation, 3) the joint positions, 4) the joint velocities, and 5) the last action. The teacher only takes 1 frame of the proprioception, while the student policy takes 10 history frames plus 1 current frame.

Reward functions and domain randimization parameters are given in Table 1 and Table 2.

Table 1: **Reward terms and their weights.** The left table lists tracking rewards, while the middle table lists penalty terms.

Tracking Reward Terms	Weights
KeyBody Position Tracking	2.0
Joint Position Tracking	0.6
Joint Velocity Tracking	0.2
Root Pose Tracking	0.6
Root Velocity Tracking	1.0

Penalty Terms	Weights
Feet Contact Penalty	-5e-4
Feet Slipping Penalty	-0.1
Joint Velocities Penalty	-1e-4
Action Rate Penalty	-0.01
Feet Air Time	5.0

Table 2: **Domain randomization parameters.**

Domain Rand Params	Range
Base Mass (kg)	$[-3, 3]$
Friction	$[0.1, 2.0]$
Motor Strength	$[0.8, 1.2]$
Gravity Change (m/s^2)	$[-0.1, 0.1]$
Push Robot Base (m/s)	$[-0.1, 0.1]$
Push End-Effector (N)	$[0, 20]$

Policy Training. We use Adam optimizer with the learning rate 1e-4. We train the teacher policy with 100k iterations and further train the student policy with 200k iterations.

Control Parameters. The control parameters used in simulation and the real world are given in Table 3. For the lower bodies, we adopt large stiffnesses for obtaining large torque.

Table 3: **Joint stiffness and damping coefficients for Unitree G1.**

Joint	Stiffness (N·m/rad)	Damping (N·m·s/rad)
Hip Yaw	100	2
Hip Roll	100	2
Hip Pitch	100	2
Knee	150	4
Ankle	40	2
Waist	150	4
Shoulder	40	5
Elbow	40	5

Real-World Infrastructure. Our system integrates a joystick controller into the real-world setup to allow seamless pausing and resuming of the teleoperation process. This feature is crucial for long-horizon tasks, where the human operator may occasionally need to pause control and adjust their own position.

B Baseline Details

In this work, we primarily study three algorithms: (1) RL+BC, (2) RL only, and (3) DAgger. Among them, RL+BC serves as the final training algorithm for TWIST. We provide further details below.

RL+BC is a two-stage framework. It first trains a teacher policy using reinforcement learning (RL), and then trains a student policy with a combination of the original task reward and a Kullback–Leibler (KL) divergence loss from the teacher.

RL only corresponds to the student policy in the RL+BC framework, but without the KL divergence term from the teacher. We find that this variant can also achieve competitive tracking performance despite the lack of supervision from the teacher.

DAgger adopts a different approach to distill the teacher. Instead of relying on RL, the student policy interacts with the environment and learns to mimic the teacher's actions directly. We use a mean squared error (MSE) loss as the loss function.