ULTRA: Unified Multimodal Control for Autonomous Humanoid Whole-Body Loco-Manipulation

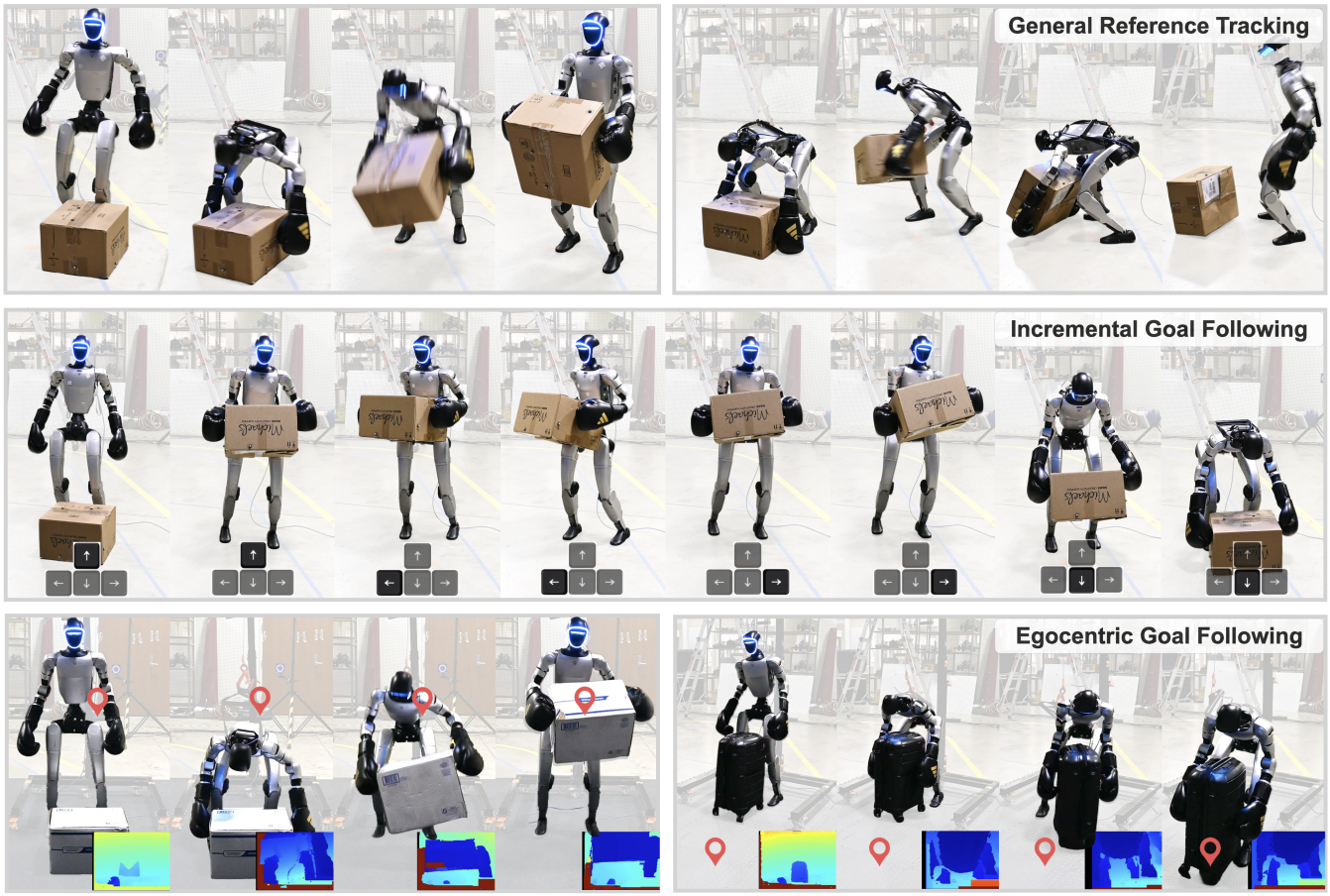


Fig. 1: ULTRA is an **all-in-one** controller for humanoid loco-manipulation that supports: **Top**. dense motion tracking from *arbitrary* reference. **Middle**. *fine-grained control* from operator commands. **Bottom**. *long-horizon goal* following with *egocentric perception*. It demonstrates autonomous whole-body behavior without relying on test-time motion references under real sensing.

Achieving autonomous and versatile whole-body loco-manipulation remains a central barrier to making humanoids practically useful. Yet existing approaches are fundamentally constrained: retargeted data are often scarce or low-quality; methods struggle to scale to large skill repertoires; and, most importantly, they rely on tracking predefined motion references rather than generating behavior from perception and high-level task specifications. To address these limitations, we propose ULTRA, a unified framework with two key components. First, we introduce a physics-driven neural retargeting algorithm that translates large-scale motion capture to humanoid embodiments while preserving physical plausibility for contact-rich interactions. Second, we learn a unified multimodal controller that supports both dense references and sparse task specifications, under sensing ranging from accurate motion-capture state to noisy egocentric visual inputs. We distill a universal tracking policy into this controller, compress motor skills into a compact latent space, and apply reinforcement learning finetuning to expand coverage and improve robustness under out-of-distribution scenarios. This enables coordinated whole-body behavior from sparse intent without test-time reference motions. We evaluate ULTRA in simulation and on a real Unitree G1 humanoid. Results show that ULTRA generalizes to autonomous, goal-conditioned whole-body loco-manipulation from egocentric perception, consistently outperforming tracking-only baselines with limited skills.

I. Introduction

Real-world loco-manipulation requires autonomy beyond replaying fixed reference motions. In unstructured environments, a humanoid must span a continuum: from dense motion references to sparse task goals, and from accurate state estimation to purely onboard sensing. Yet many controllers treat these as separate regimes and focus mainly on

reference tracking [36], [41]. This fragmentation creates a precision-flexibility trade-off: dense-tracking policies break down when references are missing or infeasible, while purely goal-conditioned policies often lack the fine-grained coordination needed for complex tasks. We therefore seek a unified controller that produces whole-body loco-manipulation and smoothly transitions between dense plans and sparse intent as information changes.

Despite progress in co-tracking humanoid and object dynamics [38], two bottlenecks hinder unified autonomy. First, kinematic retargeting can yield physically inconsistent demonstrations that fail in contact-rich tasks. Second, existing architectures typically assume a fixed conditioning structure tailored to one input type, and cannot interpret diverse or partial supervision within a consistent framework. Under shifting observability and goals at deployment, this rigidity leads to systemic instability. We address both barriers: limited, physically implausible demonstrations and policies designed mainly for tracking predefined trajectories rather than operating with subsets of conditioning signals.

To overcome the demonstration bottleneck, we introduce a physics-driven, *neural* retargeting algorithm that transfers large-scale motion capture (MoCap) to humanoid embodiments at scale. Unlike kinematic retargeting [1], [41], which struggles to maintain physical consistency in contact-rich tasks, our retargeting is dynamics- and contact-aware by construction. We cast retargeting as simulation-constrained optimization with kinematic, dynamic, and contact constraints, and solve it with reinforcement learning (RL) at scale. Once trained, the policy generates large-scale physically feasible trajectories and generalizes to arbitrary data, enabling augmentation by scaling both objects and motions.

Building on this expanded corpus, we learn a *U*nified mu*LT*imodal cont*R*oller for *A*utonomous humanoid control (ULTRA) that shifts from reference replay to perception-driven, goal-conditioned control. We first train a privileged universal tracker, then distill it into a student that follows diverse goal specifications, from dense references to sparse long-horizon targets (Fig. 1). This is enabled by (**i**) unified tokenization with availability masking [29], which keeps a single policy stable when references or modalities are missing; and (**ii**) a variational skill bottleneck plus RL finetuning [39] geared toward deployment with realistic perception and sensor noise. The bottleneck resolves ambiguity under sparse goals by maintaining coherent motion, while RL finetuning shifts control from reference-conditioned tracking to closed-loop goal stabilization under partial observability and distribution shift. Together, ULTRA yields one policy that tracks references when available and executes from egocentric perception and sparse intent when they are not.

In summary, ULTRA presents a unified system for practical whole-body loco-manipulation with three components: (**i**) a physics-driven neural retargeting pipeline that scales MoCap to humanoid embodiments and supports zero-shot augmentation; (**ii**) a versatile *multimodal* controller distilled from a privileged tracker that supports reference tracking and goal following across sensing modalities, including blind, MoCap-based, and depth-perception settings; and (**iii**) simulation and real-world evaluation on Unitree G1, showing a single unified model can outperform tracking-only baselines when references exist while enabling broader goal-conditioned behaviors as shown in Fig. 1.

II. Related Work

A. Motion Retargeting

Retargeting transfers motion across embodiments with different morphologies. It originated in animation, where inverse-kinematics optimization adapted motions under kinematic constraints [10], and later evolved into learning-based mappings that amortized transfer for better generalization [33]. Humanoid retargeting requires stronger constraints because executability is contact-dependent and further limited by joint limits and dynamics. As a result, existing robot retargeting methods trade off efficiency and physical fidelity: kinematic approaches are fast but often under-model dynamics and degrade in contact-rich settings [1], [15], [21], [29], [41], while physics-based retargeting enforces contact and dynamics for physically plausible motions, but relies on non-convex, expensive optimization, typically per-trajectory RL [24], [37] or costly sampling-based methods [20]. We target the missing regime: *physics-driven yet scalable* retargeting that preserves interaction semantics without per-trajectory RL. We perform dataset-scale retargeting with a single unified policy in one pass, and enable zero-shot augmentation to expand coverage.

B. Humanoid Whole-body Locomotion

Leveraging human motion data to teach humanoid robots complex skills has been widely studied. Early methods often use model-based control (*e.g.*, trajectory optimization and MPC) to bridge embodiment and dynamics, while recent learning-based systems achieve precise tracking and agile motion replay [3], [5]–[8], [36], [43], [44]. Beyond pure tracking, recent work moves toward foundation-style control by distilling large motion corpora into reusable priors, where a single model tracks diverse motions and supports multiple control modes [14], [16], [42], [45]. Others shape latent priors with adversarial RL [13], [17], [27], [40], but have not shown reliable scaling to large, heterogeneous loco-manipulation corpora. ULTRA follows the scalable teacher-student distillation paradigm but addresses a key bottleneck: offline distillation is limited by the state coverage of teacher rollouts. While less severe for humanoid-only control with more structured spaces, it becomes acute in high-dimensional robot-object interaction. To address this, we draw inspiration from animation practice [39], but focus on real-world deployment: we perform large-scale distillation followed by RL fine-tuning that *expands* interaction-state coverage and improves robustness to out-of-distribution goals and executions.

C. Humanoid Whole-body Loco-Manipulation

Most humanoid motion tracking emphasizes reproducing human motion on the robot and treats environmental dynamics as secondary [2], [6], [12], [28], which is brittle

for contact-rich loco-manipulation. Recent work couples humanoid motion and object interaction via co-tracking and shows strong agility [4], [36], [41], [46], but often assumes limited data replay or relies on external object state estimation (*e.g.*, motion capture), limiting autonomy under onboard egocentric sensing. Other approaches use hierarchical designs that generate trajectories/keypoints and track them with a universal controller [9], [43]; however, stacking a high-level planner on a low-level controller can accumulate error and violate physical constraints. Adversarial motion priors broaden coverage but are typically task-specific, requiring careful objective engineering and scaling poorly to large, heterogeneous loco-manipulation corpora [34]. ULTRA addresses these issues by learning a goal-conditioned policy that unifies dense tracking and sparse task specifications in a *shared* latent space, and by using RL finetuning to induce *closed-loop* behaviors that expand interaction-state coverage. This yields a *versatile* single-policy controller under real-world perception and a *scalable* paradigm that leverages broad motion corpora.

III. Problem Formulation and Preliminaries

A. Task Interface

We study whole-body loco-manipulation tasks where a humanoid interacts with a manipulated object, specified by a *goal* signal $\boldsymbol{c} \in \mathcal{C}$ that defines the task objective. A rollout succeeds if the terminal outcome satisfies $\boldsymbol{c}$, *e.g.*, the humanoid root and/or the object reaches target transformations within a tolerance. At each time step t, the policy receives (**i**) an observation $\boldsymbol{o}_t \in \mathcal{O}$ and (**ii**) task conditioning $\boldsymbol{c}_t$, and outputs an action $\boldsymbol{a}_t \in \mathcal{A}$. Here $\boldsymbol{a}_t$ specifies target joint positions executed by a PD controller.

Goal specification. We consider two forms of $\boldsymbol{c}$: (**i**) *dense reference conditioning*, which provides a time-indexed motion reference and thus specifies intermediate motions; and (**ii**) *sparse goal conditioning*, which specifies long-horizon target transformations for the humanoid root and/or object while leaving intermediate motions underdetermined.

Perception. Beyond proprioception, we consider two regimes for object sensing: (**i**) *MoCap-based sensing*, where $\boldsymbol{o}_t$ includes accurate object pose (*e.g.,* from motion capture); and (**ii**) *egocentric depth perception*, where $\boldsymbol{o}_t$ includes an egocentric point cloud from a depth sensor (*e.g.,* head-mounted), from which object state must be inferred.

B. Preliminaries

Since the controller may rely on partial onboard sensing, we model loco-manipulation as a goal-conditioned *Partially Observable Markov Decision Process* (POMDP). Let $\boldsymbol{s}_t \in \mathcal{S}$ be the underlying system state (humanoid and scene, including the object), with dynamics $\boldsymbol{s}_{t+1} \sim \mathcal{T}(\boldsymbol{s}_t, \boldsymbol{a}_t)$. The policy acts from $\boldsymbol{o}_t = \Omega(\boldsymbol{s}_t)$ and conditioning $\boldsymbol{c}_t$, producing $\boldsymbol{a}_t \in \mathcal{A}$. We optimize $\pi(\boldsymbol{a}_t \mid \boldsymbol{o}_t, \boldsymbol{c}_t)$ to maximize expected discounted return: $\max_\pi \ \mathbb{E}\big[\sum_{t\geq 0} \gamma^t\, r(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{c}_t)\big]$. where γ is the discount factor that exponentially down-weights future rewards, The following sections describe how we use PPO [26] and imitation to learn policies, including the observation/reward design and key techniques for our tasks.

IV. Method

As shown in Fig. 2, ULTRA follows a four-stage training paradigm that couples physics-driven motion retargeting with teacher-student learning. In Stage 1, we learn a *retargeting policy* that maps human MoCap motions to physically feasible humanoid loco-manipulation rollouts. In Stage 2, we train a privileged *teacher policy*, leveraging full state and dense reference trajectories from the retargeted rollouts. In Stage 3, we distill the teacher into a *multimodal student* that operates under perception and sparse goal specifications. Finally, we deploy the student with separated control mode.

A. General Motion Tracking for Neural Retargeting

Given a human-object demonstration represented by an SMPL-X [22] motion sequence and an object pose trajectory, our goal is to generate a physically feasible rollout on the target humanoid (*e.g.,* Unitree G1) that preserves the overall motion and intended interaction. Traditional retargeting solves inverse kinematics under kinematic constraints. We instead cast retargeting as RL-based trajectory optimization: rewards encode tracking, while simulator transitions enforce kinematics, dynamics, and contacts. Following [38], this is well suited for contact-rich loco-manipulation, where contacts are hard to express as kinematic constraints. As preprocessing, we scale the human-object trajectory to match G1 and define a fixed correspondence from human key links to humanoid counterparts. We then train a *unified* retargeting policy across all motions, producing physically consistent rollouts without per-motion optimization or retraining. Dense, full-body tracking is brittle under embodiment mismatch and becomes especially fragile during object interaction, where exact link-wise targets may be infeasible and contact often requires deliberate deviations. Our key insight is to combine (**i**) relaxed tracking that prioritizes end effectors critical for loco-manipulation with (**ii**) interaction and contact rewards that correct mismatch-induced errors.

Reward. We define $r_{\text{track}} = r_p \cdot r_r \cdot r_{\text{obj}} \cdot r_{\text{int}} \cdot r_{\text{ct}} \cdot r_{\text{eng}}$, with all terms computed in a heading-aligned humanoid frame. Let $\mathcal{F}$ include only feet and palms. r_p tracks end-effector positions as sparse anchors; r_r matches normalized link directions over a fixed key edge set; and r_{eng} regularizes joint effort and foot placement. To reduce ambiguity, r_{obj} tracks object pose/velocities and r_{int} matches palm-to-surface offsets over sampled object points. We also align contact events by mapping contacts on human links to corresponding humanoid links, yielding r_{ct}. Full definitions are in Sec. A.

Observation. The policy uses a privileged, reference-aware observation containing simulator state and its deviation from the SMPL-X reference. Since preprocessing establishes a fixed correspondence after scaling/alignment, residuals are well-defined: $\boldsymbol{o}_t = \big[\boldsymbol{o}_t^{\text{sim}},\ \boldsymbol{o}_t^{\text{ref}},\ \boldsymbol{o}_t^{\Delta}\big]$. $\boldsymbol{o}_t^{\text{sim}}$ includes proprioception and contact signals; $\boldsymbol{o}_t^{\text{ref}}$ provides selected correspondence-defined reference quantities (including object state); and $\boldsymbol{o}_t^{\Delta}$ encodes heading-aligned simulation-

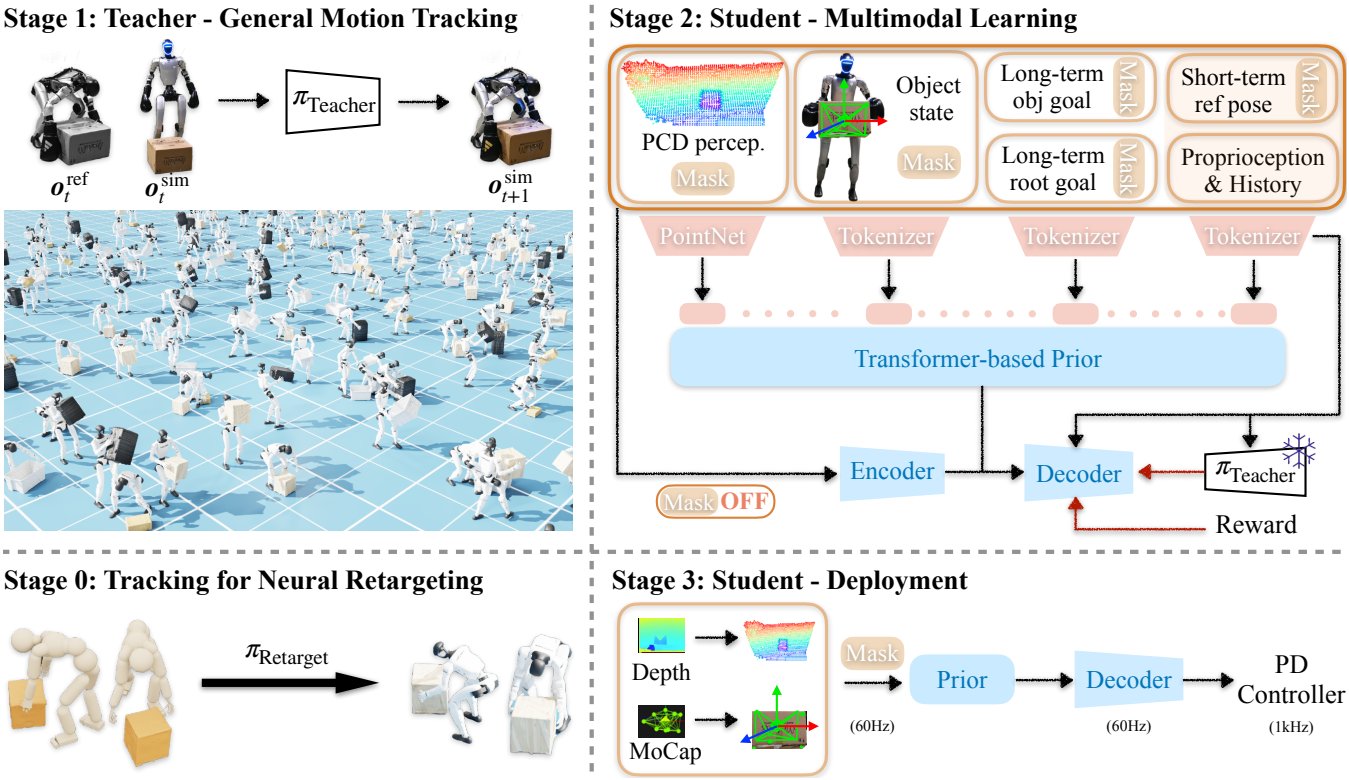


Fig. 2: ULTRA follows four stages: (**i**) **Neural Retargeting:** an RL policy converts MoCap data into physically feasible G1 rollouts with augmentation; (**ii**) **Tracking:** a privileged teacher tracks these rollouts using full state and references; (**iii**) **Distillation:** we distill the teacher into a multimodal student for realistic sensing and sparse goals, with additional RL finetuning; (**iv**) **Deployment:** the student runs under real sensing, supporting depth input or MoCap-based state estimation.

reference differences. All quantities are expressed in a heading-aligned frame to remove global yaw. See Sec. A.

State initialization and early termination. Because we cannot reliably initialize the humanoid from an SMPL-X pose, we do not use reference-state initialization [23]. Each episode starts from a default standing pose, initially tracking the first reference frame to stabilize the humanoid before transitioning to full tracking with smoothly varying weights. We terminate on falls, excessive deviation, or contact mismatch for 20 frames [38] to improve sample efficiency.

Simplified actuation. Since retargeting is used only to *generate reference rollouts*, we prioritize motion quality and throughput over hardware-faithful control. We use an *idealized* low-level controller in simulation (I., control frequency equal to simulation frequency), enabling stronger, more responsive tracking than onboard PD control. We train without domain randomization or perturbations, and address robustness later in Sec. IV-B.

Trajectory and object augmentation. RL-based retargeting also enables *flexible augmentation* (Fig. 4). Since preprocessing already scales positions, we can (**i**) apply anisotropic scaling along coordinate axes and (**ii**) scale the manipulated object with different coefficients, while interaction/contact rewards correct imperfections and the simulator enforces physical feasibility. Crucially, these augmentations are handled by a *single retargeting policy without retraining*.

B. Dense Motion Tracking for Teacher Policy

Sec. IV-A converts human-object demonstrations into physically feasible G1 rollouts. For downstream imitation, we train a separate privileged teacher π_{teacher} to track these rollouts. The teacher uses the deployment control interface and actuation limits, but trains with privileged state and dense reference residuals to accelerate learning. We randomize physics and inject perturbations to broaden state visitation and teach recovery, producing stable behaviors that provide high-quality supervision for the student.

Observation. The teacher uses the same reference-aware observation as retargeting, but does not require cross-embodiment correspondence since the reference is already in the humanoid embodiment. (Table D).

Dense tracking objective. The teacher uses the same reward template, but replaces sparse anchoring with *full* link tracking, together with object, interaction, and contact reward. (Tables B and C).

Reference initialization and robustness training. We initialize from randomly sampled reference frames and include occasional stand still episodes that track standing references, reflecting deployment from a stable standing pose. To improve robustness, we randomize humanoid/object physical properties and inject perturbations, with a short grace period to allow recovery. We use the same early-termination criteria as retargeting and add no observation noise at this stage. See

Tables I and J.

C. Multimodal Student Policy

We distill the privileged teacher into a multimodal student policy π_{student}. Unlike the teacher, the student observes only partial state and conditions on whatever modalities are available at test time via an availability mask randomly sampled during training. This retains teacher behavior as a prior while enabling goal-reaching under missing observations.

Multimodal observation with availability mask. The student consumes heterogeneous inputs: $\boldsymbol{o}_t^{\text{student}} = \left[\boldsymbol{o}_t^{\text{proprio}}, \boldsymbol{o}_t^{\text{goal}}, \boldsymbol{o}_t^{\text{object}}, \boldsymbol{o}_t^{\text{pcd}}, \boldsymbol{m}_t\right]$. $\boldsymbol{o}_t^{\text{proprio}}$ contains proprioception (*e.g.,* joint states, IMU), $\boldsymbol{o}_t^{\text{object}}$ provides object state (*e.g.,* MoCap), and $\boldsymbol{o}_t^{\text{pcd}}$ is an egocentric point cloud (*e.g.,* egocentric camera). $\boldsymbol{o}_t^{\text{goal}}$ encodes task objectives and commands, including (**i**) long-horizon object transforms, (**ii**) long-horizon humanoid root transforms, and (**iii**) next-frame humanoid local state changes for tracking. We also include discretized commands (*e.g.*, stand still) for deployment. $\boldsymbol{m}_t$ indicates which modalities are present. (Table G).

Distillation. We collect data with a DAgger-style loop [25]: we roll out with the teacher initially, gradually shift to the student, and query the teacher on visited states to obtain $\boldsymbol{a}_t^{\text{teacher}}$. During training, an encoder $q_\phi(\boldsymbol{z}_t^{\text{res}} \mid \boldsymbol{o}_t^{\text{student}}, \boldsymbol{o}_t^{\text{teacher}})$ infers a latent residual [29] using privileged teacher inputs, while a prior $p_\theta(\boldsymbol{z}_t^{\text{prior}} \mid m(\boldsymbol{o}_t^{\text{student}}))$ predicts a latent from masked student observations ($m(\cdot)$ applies $\boldsymbol{m}_t$). We combine them as $\boldsymbol{z}_t = \boldsymbol{z}_t^{\text{prior}} + \boldsymbol{z}_t^{\text{res}}$ and sample actions $\boldsymbol{a}_t^{\text{student}} \sim \pi_{\text{student}}(\boldsymbol{a}_t \mid \boldsymbol{o}_t^{\text{student}}, \boldsymbol{z}_t)$. We implement π_{student} with a transformer-based encoder [32] that projects each modality into shared tokens; $\boldsymbol{m}_t$ gates tokens and modulates cross-modal attention to ignore missing inputs. At deployment, we sample $\boldsymbol{z}_t$ from the prior only.

Training objective. We match teacher actions while aligning the prior with the privileged posterior:

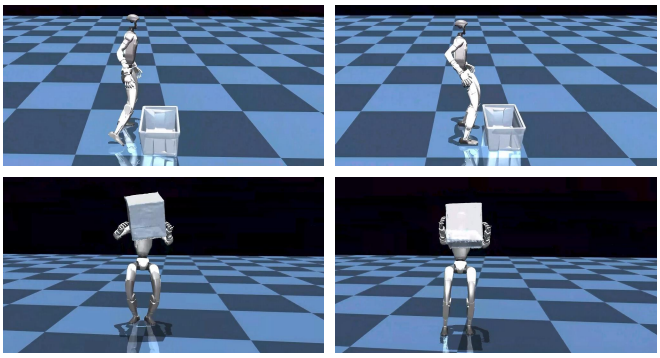
$$\mathcal{L} = \|\boldsymbol{a}_t^{\text{student}} - \boldsymbol{a}_t^{\text{teacher}}\|_2^2 + \mathcal{L}_{\text{aux}} + \lambda_{\text{KL}} D_{\text{KL}}\big(q_\phi(\boldsymbol{z}_t \mid \boldsymbol{o}_t^{\text{student}}, \boldsymbol{o}_t^{\text{teacher}}) \parallel p_\theta(\boldsymbol{z}_t \mid \boldsymbol{o}_t^{\text{student}})\big). \tag{1}$$

$\mathcal{L}_{\text{aux}}$ uses reconstruction heads (recovering masked modalities) to encourage $\boldsymbol{z}_t$ to retain task-relevant information.

Curriculum learning. Beyond DAgger, we use two curricula to keep the prior effective under partial observability: we progressively increase modality-masking probability, and anneal λ_{KL} and auxiliary weights to avoid posterior collapse while preserving latent skill diversity.

Shortcut for tracking. For local-goal tracking, behavior is largely deterministic, so a stochastic latent helps less. We add a residual shortcut (with the mask) from the full-body goal directly to the decoder, preserving low-level reference information and stabilizing decoding (Fig. 2).

RL finetuning. We perform RL finetuning on top of the distilled student by switching a subset of parallel environments to a goal-reaching objective while continuing distillation updates. Following [39], we partition simulators into (**i**) distillation environments replaying reference motions with imitation losses, and (**ii**) RL environments optimizing task success under state/goal perturbations. We sample random offsets for the object goal, humanoid root goal, and their initializations. Reward details are in Table H.



OmniRetarget ULTRA (**Ours**)

Fig. 3: Qualitative comparison of our retargeting and OmniRetarget [41] at the same frame/sequence. **Top:** final frame; the baseline shows undesired standing foot placement. **Bottom:** a contact frame; ours yields more stable contacts.

Deployment versatility. At test time, the student receives only $\boldsymbol{o}_t^{\text{student}}$ and samples $\boldsymbol{z}_t$ from the prior. With the same parameters, modality masking enables: (**i**) high-fidelity tracking by unmasking local reference (Fig. 1 **Top**), (**ii**) goal-conditioned control by masking local reference and unmasking long-horizon goals (Fig. 1 **Middle**), and (**iii**) vision-based manipulation by masking MoCap object state while unmasking point clouds (Fig. 1 **Bottom**).

V. Experimental Results

We evaluate ULTRA end-to-end for autonomous whole-body loco-manipulation, from data generation to real-world transfer. We ask: (**i**) Can we retarget human-object MoCap into physically consistent rollouts with stable contacts and minimal sliding/penetration? (**ii**) Under dense references, can the student match a privileged teacher and specialized trackers? (**iii**) Under sparse goals, does RL finetuning improve robustness and yield a semantically organized latent skill space? (**iv**) Can one policy transfer to a real humanoid without test-time references? We evaluate four axes: retargeting, tracking, goal execution, and real deployment on Unitree G1.

A. Experimental Setup

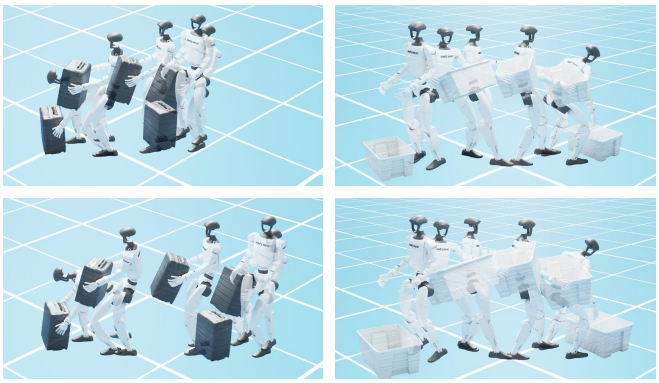
Simulation. We train in IsaacGym [19] with GPU-parallel environments and validate key results in MuJoCo [30]. Real trials use a physical Unitree G1 [31].

Dataset. We use OMOMO [11] human-object MoCap, using the corrected subset from [38] for a fair comparison with [41]. We focus on 4 box-shaped objects (others require dexterous hands). We retarget all sequences with our RL-based pipeline (Sec. IV-A) and augment via anisotropic trajectory scaling and object resizing, yielding a $\sim 6\times$ larger corpus (Fig. 4). We use the same train/test split for in-distributional (**ID**) evaluation and define out-of-distribution

TABLE I: Motion-tracking evaluation in IsaacGym. All methods are trained/evaluated on our data unless noted. **Green** highlights our primary tracking controller.

	In-Distribution (ID)							Out-of-Distribution (OOD)						
	Succ ↑	Succ ↑	Humanoid			Object		Succ ↑	Succ ↑	Humanoid			Object	
Method	(Humanoid)	(+Object)	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{jitter} ↓	E_{pos} ↓	E_{rot} ↓	(Humanoid)	(+Object)	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{jitter} ↓	E_{pos} ↓	E_{rot} ↓
(a) Unified Multimodal Controller														
ULTRA (**Ours**)	67.30±0.12	57.44±0.40	13.49±0.14	5.89±0.02	6.27±0.00	53.42±0.21	65.44±0.37	70.57±0.54	52.00±0.44	35.55±0.23	14.67±0.08	6.81±0.01	56.52±0.35	67.60±0.58
(b) Privileged Teacher														
ULTRA Teacher	97.57±0.05	89.79±0.11	12.98±0.30	5.64±0.05	14.81±0.08	17.15±0.03	23.28±0.33	97.12±0.43	81.33±0.78	19.14±0.48	7.94±0.11	15.91±0.08	25.57±0.28	33.49±0.37
(c) General Motion Tracking														
ULTRA (RL)	54.47±0.43	41.78±0.31	49.30±0.32	16.23±0.11	20.04±0.15	47.48±0.31	60.53±0.09	53.38±0.98	23.54±0.24	68.11±0.26	22.22±0.01	17.46±0.09	66.17±0.54	59.44±0.73
ULTRA (Distillation)	**85.03±3.00**	**77.15±0.57**	**15.45±0.08**	**6.84±0.04**	**8.12±0.01**	**25.48±0.48**	**33.97±0.58**	**86.63±0.50**	**52.74±0.04**	**35.01±0.31**	**13.48±0.10**	**9.35±0.01**	**36.18±0.30**	**38.18±0.29**
HDMI [36]	13.07±0.20	9.94±0.38	92.77±0.56	26.90±0.10	26.13±0.60	78.93±0.42	70.23±0.62	13.92±0.78	12.95±0.30	87.07±0.44	27.54±0.06	29.19±0.38	77.33±2.27	71.16±0.48
OmniRetarget† [41]	41.27±1.17	21.90±0.29	62.96±1.43	15.37±0.17	39.35±0.57	77.94±3.52	66.47±1.15	33.36±0.39	20.78±0.13	74.80±0.34	16.23±0.15	49.52±0.52	55.11±2.32	62.44±0.77
OmniRetarget [41]	51.34±0.67	20.91±0.52	67.12±0.86	7.43±0.07	39.92±1.44	60.67±0.54	67.03±0.19	46.71±0.74	25.82±0.52	68.34±0.82	8.98±0.19	40.08±1.77	58.57±2.37	66.70±0.84

† Trained/evaluated on original OmniRetarget dataset.



Motion Augmentation Object Augmentation

Fig. 4: Zero-shot augmentation with the retargeting policy. **Left:** trajectory scaling. **Right:** object scaling. Motions remain plausible, enabling scalable data augmentation.

(**OOD**) by held-out motions and novel object scales from our zero-shot augmentation.

B. Motion Retargeting

Baselines. We compare against: (**i**) PHC [15] (kinematics-based retargeting for G1), (**ii**) GMR [1] (humanoid motion retargeting without objects), and (**iii**) OmniRetarget [41] (interaction-preserving kinematic engine with interaction-mesh augmentation). All retarget from the same OMOMO subset processed by [38].

Metrics. We measure: (**i**) *penetration* (duration, max depth) between humanoid/object/environment [41]; (**ii**) *foot skating* (sliding duration, max tangential stance velocity), with stance defined geometrically (foot within 2 cm of ground) to avoid noisy MoCap stance labels; and (**iii**) *contact floating*, the duration of lost hand-object contact during transport, detected via MuJoCo contact queries.

Quantitative evaluation. Table II shows ULTRA outperforms baselines across nearly all metrics/categories: lowest foot-skating duration/velocity and much less contact floating (near-zero on Largebox/Suitcase), while also reducing penetration. We attribute this to physics-aware retargeting that enforces contact/dynamics, keeping stance feet planted and preserving hand-object contact when lifting.

Qualitative evaluation. Fig. 3 shows more accurate hand/foot placement than OmniRetarget, whose kinematic formulation often breaks contact consistency and yields unnatural configurations relative to the object and ground.

Effectiveness of data augmentation. Our augmentation diversifies motions without retraining the retargeter and applies along the full trajectory (not only the initial frame), producing temporally consistent variations (Fig. 4). This improves downstream generalization: in Table I, OmniRetarget retrained on our augmented data attains substantially higher OOD tracking success than when trained on its original dataset, confirming broader state/skill coverage.

C. General Motion Tracking

Baselines. We evaluate dense tracking (full reference provided) against: (**i**) OmniRetarget† (original data), (**ii**) OmniRetarget retrained on our augmented set, and (**iii**) HDMI [36] adapted to our setting. We also report ULTRA ablations: (**i**) direct RL under student observations (tracking only), (**ii**) tracking-only distillation, and (**iii**) all-task unified training. The privileged teacher is an upper bound.

Metrics. We report success (Succ): no fall and per-frame $E_{\text{g-mpjpe}} < 0.3\,\text{m}$ and $E_{\text{pos}} < 0.3\,\text{m}$; we also report humanoid-only success. Tracking errors include $E_{\text{g-mpjpe}}$, E_{mpjpe}, E_{jitter}, and object errors E_{pos}, E_{rot}.

Results. Table I shows ULTRA strongly outperforms baselines for humanoid-object tracking, especially under OOD motions/object scales. HDMI often becomes unstable at our scale and fails to converge. OmniRetarget trains smoothly but frequently fails manipulation: humanoid success is reasonable, but drops when object tracking is required, likely due to missing explicit object observations and a default-to-locomotion failure mode. ULTRA closes this gap via a privileged teacher with object/contact signals and distillation that preserves closed-loop tracking under partial observability.

Distillation vs. direct RL under partial observation. Table I shows a clear gap between ULTRA trained with direct RL under student observations and the distilled student, in both ID and OOD tracking. In contact-rich loco-manipulation, direct RL must simultaneously learn whole-

TABLE II: Physical interaction quality for retargeting. Ours is better.

Method	Penetration		Foot Skating		Contact Floating
	Duration ↓	Max Depth (cm) ↓	Duration ↓	Max Vel. (cm/s) ↓	Duration ↓
Largebox					
PHC [15]	0.908 ± 0.125	0.073 ± 0.048	0.303 ± 0.145	0.032 ± 0.022	0.025 ± 0.054
GMR [1]	0.522 ± 0.259	0.086 ± 0.053	0.366 ± 0.317	0.029 ± 0.020	0.111 ± 0.171
OmniRetarget [41]	**0.000 ± 0.002**	0.013 ± 0.002	0.205 ± 0.106	0.035 ± 0.019	0.231 ± 0.224
ULTRA (**Ours**)	0.008 ± 0.030	**0.012 ± 0.002**	**0.061 ± 0.031**	**0.018 ± 0.010**	**0.015 ± 0.063**
Suitcase					
PHC [15]	0.914 ± 0.119	0.077 ± 0.051	0.286 ± 0.147	0.035 ± 0.024	0.032 ± 0.065
GMR [1]	0.571 ± 0.265	0.105 ± 0.050	0.399 ± 0.368	0.028 ± 0.018	0.142 ± 0.175
OmniRetarget [41]	0.003 ± 0.016	**0.012 ± 0.002**	0.264 ± 0.141	0.040 ± 0.021	0.404 ± 0.279
ULTRA (**Ours**)	**0.002 ± 0.013**	0.017 ± 0.019	**0.062 ± 0.045**	**0.017 ± 0.008**	**0.008 ± 0.040**

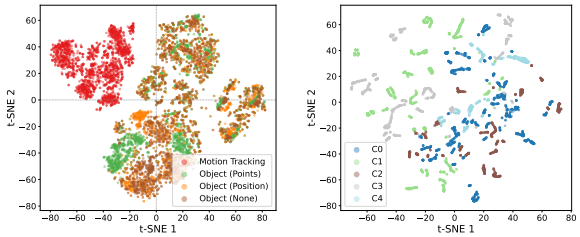


Fig. 5: **Left:** skill latent under different modalities; aside from tracking, embeddings largely mix, indicating a shared skill space. **Right:** skill latent cluster by text labels (C0–C4), showing semantic structure.

body stabilization and sustained object contact from partial observations, so early failures dominate rollouts and training often collapses. In contrast, the privileged teacher leverages full simulator state and dense references to learn contact-aware corrections with stable optimization, and distillation transfers this behavior to the student under realistic sensing, yielding higher success and lower object errors.

Distillation regularizes control. Although the teacher has access to more information, Table I shows the student can achieve *lower jitter* than the privileged teacher, this is significant for both all task student or student specialized for tracking. We attribute this to distillation acting as an implicit regularizer: matching teacher actions suppresses high-frequency, overly reactive RL corrections that reduce instantaneous error but introduce jitter and contact chattering. The student therefore learns a smoother, more contact-stable approximation that preserves the teacher's dominant strategy while discarding brittle micro-corrections.

All-task training induces a motion prior. Comparing ULTRA (Distillation) to ULTRA (**Ours**) in Table I, unified training reduces ID tracking success while *largely preserving OOD performance*. We hypothesize that jointly optimizing dense tracking and sparse goal completion encourages the policy to learn a more trajectory-invariant motion prior that remains stabilizable under partial observability. This can reduce ID tracking fidelity, since the unified controller is not trained exclusively for reference replay, but it does not harm OOD performance, where success depends more on contact-stable primitives and robust stabilization than on exact replay.

D. Goal-Conditioned Following

Metric. Success (Succ): no fall and terminal state within 0.3 m of the goal.

Comparisons. Tracking-only baselines (*e.g.,* HDMI [36], OmniRetarget [41]) require dense references and are inapplicable; we compare ULTRA to ablations.

Tasks. We deploy ULTRA on a physical Unitree G1. The student runs onboard at the control frequency with proprioception and, when available, OptiTrack object pose (Fig. A). For dense tracking, we test OMOMO subsets (bimanual box lift/carry, suitcase transport) with household objects (Fig. B). For goal-conditioned control, we provide no motion references and specify future object transforms via simple keyboard commands.

RL finetuning expands OOD coverage. Table III and Fig. 6 show finetuning yields modest ID gains but large OOD gains under random goal offsets (nearly doubling under point clouds and tripling under position-only). This suggests finetuning expands interaction-state coverage and reinforces closed-loop recovery beyond the demonstration manifold.

Latent space shows control modes and motion semantics. We visualize the learned motion embeddings with t-SNE [18] to interpret what the motor latent capture. Fig. 5 (left) shows that the latent space cleanly separates dense reference tracking from sparse goal following across input modalities, while remaining within a shared manifold. Motion tracking stays distinct because we do not force it through the stochastic latent: when a local tracking goal is given, we pass a residual shortcut from the full-body goal directly to the decoder (Sec. IV-C). This leaves the latent to capture mainly ambiguity and multimodality under sparse goals. Fig. 5 (right) further shows *semantic structure*: we encode each motion's text description with MiniLM [35], cluster the resulting text embeddings into 5 classes with K-Means, and then plot the corresponding latents. The latent projections align with these semantic clusters, suggesting that the transformer encoder organizes motor skills by both control regime and high-level motion intent, reducing ambiguity under sparse goals by mapping them to appropriate regions of the skill manifold.

E. Real-World Deployment

Tasks. We deploy ULTRA on a physical Unitree G1. The student runs onboard at the control frequency with proprioception and, when available, OptiTrack object pose. For dense tracking, we test OMOMO subsets (bimanual box lift/carry, suitcase transport) with household objects. For goal-conditioned control, we provide no motion references and specify future object transforms via keyboard commands.

Point cloud extraction. For egocentric perception, we extract object point clouds from depth only: back-project depth pixels using calibrated intrinsics, crop a forward ROI, remove the ground plane, take the dominant cluster as the box, and downsample to a fixed size for policy input.

Quantitative evaluation. Table IV reports success rates: the policy reliably grasps/transports on hardware and achieves reasonable sparse-goal success under out-of-distribution operator commands, including composed motions.

Failure analysis. Failures mainly arise from (**i**) friction gaps causing occasional grasp slip, (**ii**) depth noise/occlusion

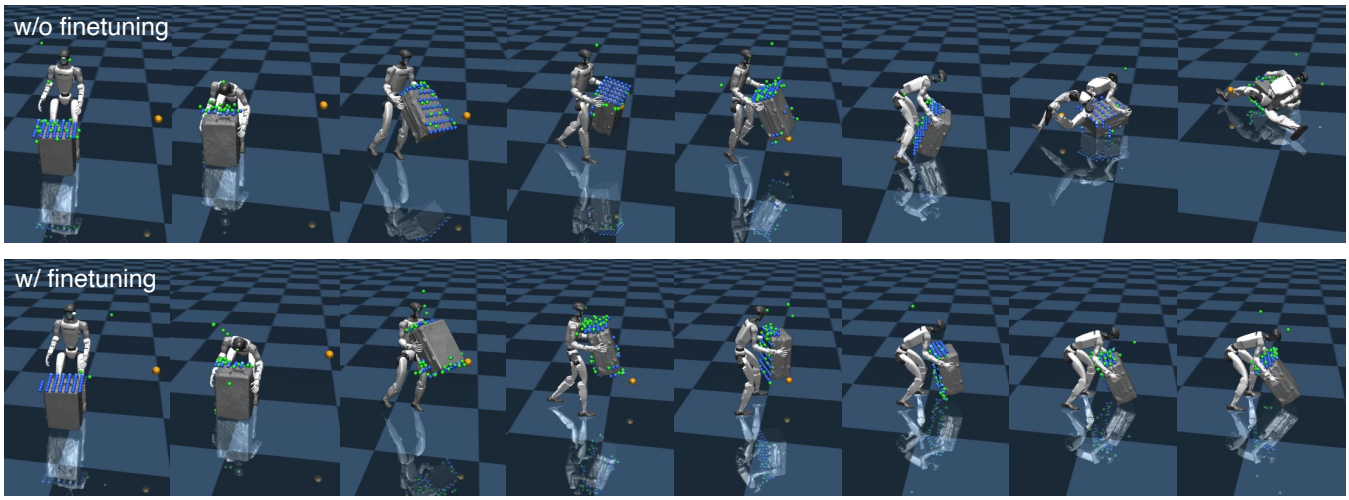


Fig. 6: Sim-to-sim comparison for egocentric goal following. Blue/green: point cloud observation without/with noise; yellow: object goal. **Top:** without RL finetuning. **Bottom:** with RL finetuning.

TABLE III: Sim-to-sim success rate on Mujoco across goal type with in-distributional (ID) goals from training and out-of-distributional (OOD) goals with random offsets, and across perception with egocentric point clouds or object position with no shape. Policies are trained in IsaacGym and evaluated in MuJoCo with 20 selected motion per setting.

RL fine-tuning	ID Goals		OOD Goals	
	Points	Position	Points	Position
✗	16 / 20	14 / 20	5 / 20	4 / 20
✓	19 / 20	16 / 20	**9 / 20**	**12 / 20**
Δ (RL gain)	+18.8%	+14.3%	**+80.0%**	**+200.0%**

TABLE IV: Real-world success rates on the OMOMO subset using a Unitree G1 humanoid. Each task is evaluated over two trials. MoCap provides object pose tracking for non-egocentric control modes, while the egocentric setting relies only on onboard sensing. MoCap is used for success evaluation in all settings. Dense reference tracking is direction-agnostic and thus reported as a single success rate.

Setting	Vertical	Lateral
Dense Reference Tracking	**73%** (19/26)	
Sparse Goal Following (MoCap)	**80%** (8/10)	**90%** (9/10)
Sparse Goal Following (Egocentric)	**50%** (5/10)	**60%** (6/10)

breaking point-cloud extraction, and (**iii**) disturbances beyond the recovery margin learned with domain randomization, motivating future tactile integration.

VI. Conclusion

ULTRA is a unified framework for practical humanoid whole-body loco-manipulation that moves beyond reference replay toward perception- and goal-driven autonomy. It combines an RL-formulated, physics-driven retargeting policy that scales human-object MoCap into physically consistent humanoid rollouts with a distilled multimodal controller that unifies dense tracking and sparse goal specification. Experiments show improved interaction fidelity from retargeting, a student that matches tracking performance while remaining robust under distribution shift, and RL finetuning that boosts success on out-of-distribution goals. We further validate sim-to-real transfer on Unitree G1, demonstrating reliable dense tracking and sparse goal following. Overall, ULTRA points to a scalable path for versatile loco-manipulation that adapts online from realistic sensing without test-time references.

References

[1] J. P. Araujo, Y. Ze, P. Xu, J. Wu, and C. K. Liu, "Retargeting matters: General motion retargeting for humanoid motion tracking," *arXiv preprint arXiv:2510.02252*, 2025. 2, 6, 7

[2] 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. 2

[3] Z. Chen, M. Ji, X. Cheng, X. Peng, X. B. Peng, and X. Wang, "Gmt: General motion tracking for humanoid whole-body control," *arXiv preprint arXiv:2506.14770*, 2025. 2

[4] Y. Fu, F. Xie, C. Xu, J. Xiong, H. Yuan, and Z. Lu, "DemoHLM: From one demonstration to generalizable humanoid loco-manipulation," *arXiv preprint arXiv:2510.11258*, 2025. 3

[5] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbab, C. Pan, *et al.*, "Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills," *arXiv preprint arXiv:2502.01143*, 2025. 2

[6] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," *arXiv preprint arXiv:2406.08858*, 2024. 2

[7] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang, *et al.*, "Hover: Versatile neural whole-body controller for humanoid robots," in *ICRA*, 2025. 2

[8] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang, "Exbody2: Advanced expressive humanoid whole-body control," *arXiv preprint arXiv:2412.13196*, 2024. 2

[9] D. Kalaria, S. S. Harithas, P. Katara, S. Kwak, S. Bhagat, S. Sastry, S. Sridhar, S. Vemprala, A. Kapoor, and J. C.-K. Huang, "Dreamcontrol: Human-inspired whole-body humanoid control for scene interaction via guided diffusion," *arXiv preprint arXiv:2509.14353*, 2025. 3

[10] J. Lee and S. Y. Shin, "A hierarchical approach to interactive motion editing for human-like figures," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 39–48. 2

[11] J. Li, J. Wu, and C. K. Liu, "Object motion guided human motion synthesis," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 6, pp. 1–11, 2023. 5

[12] J. Li, Y. Zhu, Y. Xie, Z. Jiang, M. Seo, G. Pavlakos, and Y. Zhu, "OKAMI: Teaching humanoid robots manipulation skills through single video imitation," in *CoRL*, 2024. 2

[13] Y. Li, Z. Luo, T. Zhang, C. Dai, A. Kanervisto, A. Tirinzoni, H. Weng, K. Kitani, M. Guzek, A. Touati, *et al.*, "Bfm-zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning," *arXiv preprint arXiv:2511.04131*, 2025. 2

[14] Q. Liao, T. E. Truong, X. Huang, Y. Gao, G. Tevet, K. Sreenath, and C. K. Liu, "Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion," *arXiv preprint arXiv:2508.08241*, 2025. 2

[15] Z. Luo, J. Cao, K. Kitani, W. Xu, *et al.*, "Perpetual humanoid control for real-time simulated avatars," in *ICCV*, 2023. 2, 6, 7

[16] Z. Luo, Y. Yuan, T. Wang, C. Li, S. Chen, F. Castañeda, Z.-A. Cao, J. Li, D. Minor, Q. Ben, *et al.*, "Sonic: Supersizing motion tracking for natural humanoid whole-body control," *arXiv preprint arXiv:2511.07820*, 2025. 2

[17] L. Ma, Z. Meng, T. Liu, Y. Li, R. Song, W. Zhang, and S. Huang, "Styleloco: Generative adversarial distillation for natural humanoid robot locomotion," *arXiv preprint arXiv:2503.15082*, 2025. 2

[18] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008. 7

[19] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," in *NeurIPS*, 2021. 5

[20] C. Pan, C. Wang, H. Qi, Z. Liu, H. Bharadhwaj, A. Sharma, T. Wu, G. Shi, J. Malik, and F. Hogan, "Spider: Scalable physics-informed dexterous retargeting," *arXiv preprint arXiv:2511.09484*, 2025. 2

[21] S. Park, H. Bharadhwaj, and S. Tulsiani, "Demodiffusion: One-shot human imitation using pre-trained diffusion policy," *arXiv preprint arXiv:2506.20668*, 2025. 2

[22] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3D hands, face, and body from a single image," in *CVPR*, 2019. 3

[23] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018. 4

[24] D. Reda, J. Won, Y. Ye, M. van de Panne, and A. Winkler, "Physics-based motion retargeting from sparse inputs," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 6, no. 3, pp. 1–19, 2023. 2

[25] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635. 5

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017. 3

[27] J. Shi, X. Liu, D. Wang, O. Lu, S. Schwertfeger, C. Zhang, F. Sun, C. Bai, and X. Li, "Adversarial locomotion and motion imitation for humanoid policy learning," *arXiv preprint arXiv:2504.14305*, 2025. 2

[28] W. Sun, L. Feng, B. Cao, Y. Liu, Y. Jin, and Z. Xie, "Ulc: A unified and fine-grained controller for humanoid loco-manipulation," *arXiv preprint arXiv:2507.06905*, 2025. 2

[29] C. Tessler, Y. Guo, O. Nabati, G. Chechik, and X. B. Peng, "Maskedmimic: Unified physics-based character control through masked motion inpainting," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 6, pp. 1–21, 2024. 2, 5

[30] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IROS*, 2012. 5

[31] Unitree, "Unitree g1 humanoid agent ai avatar," https://www.unitree.com/g1/. 5

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017. 5

[33] R. Villegas, J. Yang, D. Ceylan, and H. Lee, "Neural kinematic networks for unsupervised motion retargetting," *CVPR*, 2018. 2

[34] H. Wang, W. Zhang, R. Yu, T. Huang, J. Ren, F. Jia, Z. Wang, X. Niu, X. Chen, J. Chen, *et al.*, "Physhsi: Towards a real-world generalizable and natural humanoid-scene interaction system," *arXiv preprint arXiv:2510.11072*, 2025. 3

[35] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," in *NeurIPS*, 2020. 7

[36] H. Weng, Y. Li, N. Sobanbabu, Z. Wang, Z. Luo, T. He, D. Ramanan, and G. Shi, "Hdmi: Learning interactive humanoid whole-body control from human videos," *arXiv preprint arXiv:2509.16757*, 2025. 2, 3, 6, 7

[37] S. Xu, Y.-W. Chao, L. Bian, A. Mousavian, Y.-X. Wang, L.-Y. Gui, and W. Yang, "Dexplore: Scalable neural control for dexterous manipulation from reference-scoped exploration," in *CoRL*, 2025. 2

[38] S. Xu, H. Y. Ling, Y.-X. Wang, and L.-Y. Gui, "InterMimic: Towards universal whole-body control for physics-based human-object interactions," in *CVPR*, 2025. 2, 3, 4, 5, 6

[39] S. Xu, S. Schulter, M. Ziyadi, X. He, X. Fei, Y.-X. Wang, and L. Gui, "InterPrior: Scaling generative control for physics-based human-object interactions," *arXiv preprint arXiv:2602.06035*, 2026. 2, 5

[40] H. Xue, X. Huang, D. Niu, Q. Liao, T. Kragerud, J. T. Gravdahl, X. B. Peng, G. Shi, T. Darrell, K. Sreenath, *et al.*, "Leverb: Humanoid whole-body control with latent vision-language instruction," *arXiv preprint arXiv:2506.13751*, 2025. 2

[41] L. Yang, X. Huang, Z. Wu, A. Kanazawa, P. Abbeel, C. Sferrazza, C. K. Liu, R. Duan, and G. Shi, "Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction," *arXiv preprint arXiv:2509.26633*, 2025. 2, 3, 5, 6, 7

[42] K. Yin, W. Zeng, K. Fan, M. Dai, Z. Wang, Q. Zhang, Z. Tian, J. Wang, J. Pang, and W. Zhang, "Unitracker: Learning universal whole-body motion tracker for humanoid robots," *arXiv preprint arXiv:2507.07356*, 2025. 2

[43] S. Yin, Y. Ze, H.-X. Yu, C. K. Liu, and J. Wu, "Visualmimic: Visual humanoid loco-manipulation via motion tracking and generation," *arXiv preprint arXiv:2509.20322*, 2025. 2, 3

[44] Y. Ze, Z. Chen, J. P. Araújo, Z.-a. Cao, X. B. Peng, J. Wu, and C. K. Liu, "Twist: Teleoperated whole-body imitation system," *arXiv preprint arXiv:2505.02833*, 2025. 2

[45] W. Zeng, S. Lu, K. Yin, X. Niu, M. Dai, J. Wang, and J. Pang, "Behavior foundation model for humanoid robots," *arXiv preprint arXiv:2509.13780*, 2025. 2

[46] S. Zhao, Y. Ze, Y. Wang, C. K. Liu, P. Abbeel, G. Shi, and R. Duan, "Resmimic: From general motion tracking to humanoid whole-body loco-manipulation via residual learning," *arXiv preprint arXiv:2510.05070*, 2025. 3

APPENDIX

In this appendix, we provide additional details of our ULTRA:

1) Sec. A provides additional details on retargeting and the teacher policy, including the observation and reward design.
2) Sec. B provides additional details on the student policy, covering both the distillation stage and the RL finetuning stage.
3) Sec. C provides additional experimental details and setups for both simulation and real-world experiments.

A. Additional Details on Retargeting and Teacher Policy

We provide additional details on retargeting and teacher policy training. Both follow the same general procedure. The key difference is that the retargeting policy tracks reference motions from a different embodiment, which requires a predefined key-joint mapping when constructing rewards and observations. In this appendix, we focus on the teacher policy reward and observation; This can be easily extended to the retargeting policy, which uses the same formulations, with the additional cross-embodiment link alignment applied (Table A).

Rewards. The teacher policy reward combines tracking terms with smoothness and regularization penalties to facilitate sim-to-real transfer for the distilled student policy. The total reward is defined as

$$r_{\text{teacher}} = r_{\text{track}} + \sum_i w_i \, r_i^{\text{smooth}}. \tag{2}$$

Table B lists all tracking reward terms, and Table C summarizes all smoothness and regularization terms.

TABLE A: Correspondence between Unitree G1 key links and SMPL-X body indices (52-body model) used for motion retargeting.

Unitree G1 Link	SMPL-X Index	SMPL-X Body Part
left_hip_yaw_link	1	Left hip
left_knee_link	2	Left knee
left_ankle_roll_link	3	Left ankle
right_hip_yaw_link	5	Right hip
right_knee_link	6	Right knee
right_ankle_roll_link	7	Right ankle
torso_link	9	Pelvis / lower torso
mid360_link	13	Upper torso / spine
left_shoulder_yaw_link	15	Left shoulder
left_elbow_link	16	Left elbow
left_wrist_yaw_link	17	Left wrist
right_shoulder_yaw_link	34	Right shoulder
right_elbow_link	35	Right elbow
right_wrist_yaw_link	36	Right wrist

Observation. At time step t, the teacher policy receives an observation vector

$$\boldsymbol{o}_t = \left[\boldsymbol{o}_t^{\text{sim}},\ \boldsymbol{o}_t^{\text{ref}},\ \boldsymbol{o}_t^{\Delta},\ \boldsymbol{o}_t^{\text{ig}}\right], \tag{3}$$

where the four blocks correspond to simulated state, reference targets, simulation–reference residuals, and interaction-graph features, respectively. Table D details the observation components and variable definitions.

TABLE B: **Tracking reward components for the teacher policy.** $\mathbf{p}_l, \mathbf{q}, \dot{\mathbf{p}}_l, \dot{\mathbf{q}}$ denote the simulated body joint positions, joint rotation, and their velocities; $\hat{\mathbf{p}}_l, \hat{\mathbf{q}}, \hat{\dot{\mathbf{p}}}_l, \hat{\dot{\mathbf{q}}}$ are the corresponding reference quantities. $\mathbf{p}_o, \mathbf{q}_o, \dot{\mathbf{p}}_o$ denote the simulated object position, rotation (quaternion), and linear velocity; $\hat{\mathbf{p}}_o, \hat{\mathbf{q}}_o, \hat{\dot{\mathbf{p}}}_o$ are the corresponding references. $\angle(\mathbf{q}_o, \hat{\mathbf{q}}_o)$ is the relative rotation angle and $\text{huber}(\cdot)$ is the Huber loss. $\boldsymbol{\delta}_{ij}$ denotes the palm-to-surface distance between palm point i and object surface sample j, with weights w_{ij}; hats denote reference values. $c_l \in \{0, 1\}$ is a binary contact indicator for link l. The scalars $k_\cdot$ are temperature coefficients. All reward terms are multiplied together.

Term	Expression	Weight
Body Tracking:		
Joint position	$\exp\left(-k_p \cdot \text{mean}(\|\mathbf{p}_l - \hat{\mathbf{p}}_l\|_2^2)\right)$	$k_p = 10.0$
Joint rotation	$\exp\left(-k_r \cdot \text{mean}(\|\mathbf{q} - \hat{\mathbf{q}}\|_2^2)\right)$	$k_r = 5.0$
Body velocity	$\exp\left(-k_{pv} \cdot \text{mean}(\|\dot{\mathbf{p}}_l - \hat{\dot{\mathbf{p}}}_l\|_2^2)\right)$	$k_{pv} = 0.1$
Joint velocity	$\exp\left(-k_{rv} \cdot \text{mean}(\|\dot{\mathbf{q}} - \hat{\dot{\mathbf{q}}}\|_2^2)\right)$	$k_{rv} = 0.001$
Object Tracking:		
Object position	$\exp\left(-k_{op} \cdot \text{mean}(\|\mathbf{p}_o - \hat{\mathbf{p}}_o\|_2^2)\right)$	$k_{op} = 5.0$
Object rotation	$\exp\left(-k_{or} \cdot \text{huber}(\angle(\mathbf{q}_o, \hat{\mathbf{q}}_o))\right)$	$k_{or} = 0.5$
Object linear velocity	$\exp\left(-k_{opv} \cdot \text{mean}(\|\dot{\mathbf{p}}_o - \hat{\dot{\mathbf{p}}}_o\|_2^2)\right)$	$k_{opv} = 0.1$
Interaction:		
Palm-to-surface	$\exp\left(-k_{\text{int}} \cdot \sum_{i,j} w_{ij} \|\boldsymbol{\delta}_{ij} - \hat{\boldsymbol{\delta}}_{ij}\|_2^2\right)$	$k_{\text{int}} = 20.0$
Contact matching	$\exp\left(-k_{\text{ct}} \cdot \text{mean}(\|c_l - \hat{c}_l\|)\right)$	$k_{\text{ct}} = 5.0$

Architecture. The teacher policy uses a three-layer MLP with hidden dimensions 1024, 1024, and 512, and ReLU activations. It follows a separate actor–critic design, producing a 29-dimensional action output. The action mean is output with no activation (*i.e.,* a linear head), while the action standard deviation is fixed rather than learned and is initialized to -2.9. All weights use the default Xavier initialization.

Training. We summarize PPO hyperparameters in Table E.

B. Additional Details on Student Policy

Student Policy Observation. The student policy observation extends the student observation with multimodal inputs including object point clouds and goal phase information. The observation is structured as $\boldsymbol{o}_t^{\text{student}} = [\boldsymbol{o}^{\text{global}}, \boldsymbol{o}^{\text{cmd}}, \boldsymbol{o}^{\text{local}}, \boldsymbol{o}^{\text{proprio}}, \boldsymbol{o}^{\text{task}}, \boldsymbol{m}]$. We summarize all components in Table G.

Student Policy Architecture. ULTRA enables learning a latent representation that captures task-relevant information while maintaining robustness to noise and missing modalities.

Specifically, the student is a latent-variable policy with a 64-dimensional latent z and a modality-fusion transformer. Each input modality, as summarized in Table G, is first encoded into a 256-dimensional token; point-cloud perception uses a PointNet over 64 3D points that outputs a 256-dimensional feature via a point MLP and global pooling/statistics, while all other modalities are embedded with MLP encoders into the same token space. These modality tokens are fused by a lightweight transformer with token dimension 256, two layers, four attention heads, and a 1024-dimensional feed-forward network, using GELU activations,

TABLE C: **Smoothness and regularization rewards for the teacher policy.** All terms are penalties with negative weights $w_i < 0$. $\mathbf{v}_{\text{base}}$ and $\boldsymbol{\omega}_{\text{base}}$ are the base linear and angular velocities. $\mathbf{a}_t$ is the action at time t; $\dot{\mathbf{q}}_t$ is the joint velocity at time t; and $\boldsymbol{\omega}_t$ is the base angular velocity at time t. $\boldsymbol{\tau}$ denotes joint torques and $\odot$ is elementwise multiplication. $\mathbf{q}_{\text{lim}}$ and $\boldsymbol{\tau}_{\text{lim}}$ are per-joint position and torque limits. $\mathbb{1}_{\text{contact}}$ is an indicator for foot contact; $\mathbf{f}_{xy}$ and $\mathbf{f}_z$ are the horizontal and vertical components of the contact force. d_{feet} and d_{knee} are the distances between the two feet and the two knees, and $\text{clamp}(\cdot)$ clips the distance to the specified interval. $\mathbf{g}_{\perp}^{\text{feet}}$ measures foot tilt relative to gravity. ref_stand and sim_stand indicate standing phases in the reference and simulation. h_{term} is a swing-foot height term, c is a clearance/contact-related term, and g_{swing} gates the swing penalty, which is active only during reference indicating swing.

Term	Expression	Weight
Velocity Penalties:		
Base linear velocity	$\|\mathbf{v}_{\text{base}}\|_2$	-0.1
Base angular velocity	$\|\boldsymbol{\omega}_{\text{base}}\|_2^2$	-0.01
Joint velocity	$\|\dot{\mathbf{q}}\|_2^2$	-0.0004
Smoothness Penalties:		
Action rate	$\|\mathbf{a}_t - \mathbf{a}_{t-1}\|_2$	-0.1
Joint velocity change	$\|\dot{\mathbf{q}}_t - \dot{\mathbf{q}}_{t-1}\|_2^2$	-2×10^{-5}
Angular velocity change	$\|\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t-1}\|_2^2$	-5×10^{-4}
Torque & Energy Penalties:		
Torque magnitude	$\|\boldsymbol{\tau}\|_2$	-0.001
Energy consumption	$\|\boldsymbol{\tau} \odot \dot{\mathbf{q}}\|_2$	-0.0001
Joint position limits	$\sum \max(0, \|\mathbf{q}\| - \mathbf{q}_{\text{lim}})$	-5.0
Joint torque limits	$\sum \max(0, \|\boldsymbol{\tau}\|/\boldsymbol{\tau}_{\text{lim}} - 0.95)$	-1.0
Foot & Stability Penalties:		
Feet orientation	$\|\mathbf{g}_{\perp}^{\text{feet}}\|_2$	-0.35
Foot slip	$\sqrt{\|\mathbf{v}_{\text{foot}}\|_2} \cdot \mathbb{1}_{\text{contact}}$	-0.1
Feet stumble	$\mathbb{1}(\|\mathbf{f}_{xy}\| > 4\|\mathbf{f}_z\|)$	-10.0
Feet distance	$\text{clamp}(d_{\text{feet}} - [0.25, 0.65])$	-0.1
Knee distance	$\text{clamp}(d_{\text{knee}} - [0.25, 0.65])$	-0.1
Stand on feet	$\mathbb{1}(\text{ref\_stand} \wedge \neg\text{sim\_stand})$	-1.0
Swing clearance	$(w_h \cdot h_{\text{term}} + w_c \cdot c) \cdot g_{\text{swing}}$	-0.6
Termination	$\mathbb{1}_{\text{terminated}}$	-50.0

sinusoidal positional encoding, and zero dropout.

For the latent model, a prior network predicts the Gaussian parameters of z from the transformer context token using two MLP heads, each mapping 256 to 128 to 64 with ReLU. During training, a privileged encoder takes the teacher observation together with additional observations and privileged signals, processes them with an MLP of widths 2048, 1024, 512, and 256, and outputs the posterior with the same 256 to 128 to 64 heads. An auxiliary latent decoder maps z through a small MLP from 64 to 256 to 16 for reconstruction or regularization. Finally, z conditions the policy via FiLM by predicting per-layer scale and shift parameters with a linear projection, and the FiLM modulation is scaled by 0.1.

Distillation Training. ULTRA is first pretrained via a distillation loop that combines on-policy rollouts with teacher supervision. We apply a DAgger mixing schedule to transition from teacher-driven rollouts to student-driven rollouts. Specifically, we use full teacher rollout below 500 epoch, then linearly anneal to over epochs 1500 epoch for full student rollout. We use 4096 parallel environments with horizon length 8, mini-batch size 4096, and 2 mini-epochs per update, which updates more frequently compared to PPO setup in Table E, given that supervised distillation is much more stable. The learning rate follows a warmup-and-decay schedule: it starts at 2×10^{-4}, ends warmup at epoch 500, and decays to 5×10^{-5} by epoch 5500. The latent dimension is 64.

Distillation Loss. In the distillation stage, the optimization uses supervised and latent-regularization objectives. The PPO actor-critic terms are disabled in the provided implementation, so the total loss is a weighted sum of (**I**) expert supervision on the action mean, (**II**) KL alignment between the privileged posterior and the student prior, (**III**) a temporal smoothness penalty on the latent mean, (**IV**)

TABLE D: **Teacher policy observation space.** At time t, $\boldsymbol{o}_t = [\boldsymbol{o}_t^{\text{sim}}, \boldsymbol{o}_t^{\text{ref}}, \boldsymbol{o}_t^{\Delta}, \boldsymbol{o}_t^{\text{ig}}]$. $\boldsymbol{o}_t^{\text{sim}}$ contains simulated quantities: root height; per-body local positions $\mathbf{p}$, rotations $\mathbf{R}$ in 6D tan-norm, linear velocities $\dot{\mathbf{p}}$, angular velocities $\boldsymbol{\omega}$, contact flags $c \in \{0, 1\}$; and joint states (positions $\mathbf{q}$, velocities $\dot{\mathbf{q}}$, actions $\mathbf{a}$, torques $\boldsymbol{\tau}$, and history). $\boldsymbol{o}_t^{\text{ref}}$ contains corresponding reference targets (hats), *e.g.,* body pose $(\hat{\mathbf{p}}, \hat{\mathbf{R}})$ and object pose/velocity $(\hat{\mathbf{x}}_o, \hat{\mathbf{q}}_o, \hat{\mathbf{v}}_o, \hat{\boldsymbol{\omega}}_o)$. $\boldsymbol{o}_t^{\Delta}$ stores residuals between simulation and reference (*e.g.,* $\mathbf{p} - \hat{\mathbf{p}}$ and velocity errors), and $\boldsymbol{o}_t^{\text{ig}}$ stores interaction-graph features based on SDF distances between body points and the object surface and their residuals. Two observations based on next 1-frame and next 16-frame reference are concatenated, yielding total dimension 4052.

Category	Feature	Dimension	Description
$\boldsymbol{o}^{\text{sim}}$	Root height	1	Root height above ground
	Local body positions $\mathbf{p}$	114	39 bodies $\times$ 3 (root removed)
	Local body rotations $\mathbf{R}$	234	39 bodies $\times$ 6 (tan-norm)
	Local body velocities $\dot{\mathbf{p}}$	117	39 bodies $\times$ 3
	Local body angular vel. $\boldsymbol{\omega}$	117	39 bodies $\times$ 3
	Contact indicators c	39	Binary contact flags
	Joint states $(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{a}, \boldsymbol{\tau})$	145	Proprioception + short history
$\boldsymbol{o}^{\text{ref}}$	Body reference $(\hat{\mathbf{p}}, \hat{\mathbf{R}})$	351	Positions (117) + rotations (234)
	Object reference	13	Pose $(\hat{\mathbf{x}}_o, \hat{\mathbf{q}}_o)$ (7) + vel. $(\hat{\mathbf{v}}_o, \hat{\boldsymbol{\omega}}_o)$ (6)
$\boldsymbol{o}^{\Delta}$	Body residuals	585	Pose/velocity residuals vs. reference
	Object residuals	21	Object pose/velocity residuals vs. reference
$\boldsymbol{o}^{\text{ig}}$	Interaction graph	234	39 $\times$ 3 SDF distances + residuals
Total		4052	Concatenate 2 frames (current + future)

TABLE E: PPO hyperparameters.

Hyperparameter	Value
Learning rate	2×10^{-5}
Clip ratio ϵ	0.2
GAE λ (tau)	0.95
Discount γ	0.99
Horizon length	32
Mini-batch size	16384
Mini epochs per update	6
Entropy coefficient	0.0
Critic loss coefficient	5.0
Bounds loss coefficient	10.0
Max gradient norm	1.0
Number of parallel envs	4096
Normalize input	True
Normalize value	False
Normalize advantage	True

TABLE F: **Distillation-stage loss terms.** $\boldsymbol{\mu}$ is the student action mean from the full model forward pass, and $\boldsymbol{\mu}^{\text{prior}}$ is the action mean from a prior-only forward pass (encoder disabled). $\boldsymbol{a}^{\text{exp}}$ denotes the expert action mean (teacher target) used for supervision. The prior and privileged encoder output diagonal Gaussians over the latent z, denoted by $p_\theta(z \mid \boldsymbol{o}) = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p^2)$ and $q_\phi(z \mid \boldsymbol{o}, \boldsymbol{o}^{\text{priv}}) = \mathcal{N}(\boldsymbol{\mu}_e, \boldsymbol{\sigma}_e^2)$. Let ep be the epoch index and $s = \text{clip}\big(\frac{\text{ep}-500}{3000}, 0, 1\big)$. The cosine ramp is $g(s) = \frac{1-\cos(\pi s)}{2}$.

Term	Definition	Weight / schedule
Total loss	$\mathcal{L} = \lambda_E\mathcal{L}_E + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_S\mathcal{L}_S + \lambda_A\mathcal{L}_A + \lambda_G\mathcal{L}_G + \lambda_P\mathcal{L}_P$	see below
Expert imitation	$\mathcal{L}_E = \lVert\boldsymbol{\mu} - \boldsymbol{a}^{\text{exp}}\rVert_2^2$	$\lambda_E = 1.0$
Latent KL alignment	$\mathcal{L}_{\text{KL}} = D_{\text{KL}}\big(q_\phi(z) \parallel p_\theta(z)\big)$	$\lambda_{\text{KL}} = 0.001 + (0.1 - 0.001)\, g(s)$
Latent smoothness	$\mathcal{L}_S = \big\lVert(\boldsymbol{\mu}_p + \boldsymbol{\mu}_e)_t - (\boldsymbol{\mu}_p + \boldsymbol{\mu}_e)_{t-1}\big\rVert_2^2$	$\lambda_S = 0.0001 + (0.001 - 0.0001)\, g(s)$
Auxiliary prediction	$\mathcal{L}_A = \text{MSE}(\hat{\boldsymbol{y}}_{\text{aux}}, \boldsymbol{y}_{\text{aux}})$ with mask-weighting	$\lambda_A = 1.0$
Local-goal prediction	$\mathcal{L}_G = \text{MSE}(\hat{\boldsymbol{g}}_{\text{local}}, \boldsymbol{g}_{\text{local}})$ with mask-weighting	$\lambda_G = 1.0$

TABLE G: Student policy observation space.

Category	Feature	Dimension	Description
$\boldsymbol{o}^{\text{global}}$	Root position residual (xy)	2	Horizontal position error
	Heading residual (yaw)	1	Yaw angle error
$\boldsymbol{o}^{\text{cmd}}$	End-of-episode flag	1	Near episode end indicator
	Approaching flag	1	Moving toward object
	Leaving flag	1	Moving away from object
	Time-to-go	1	Normalized remaining time
$\boldsymbol{o}^{\text{local}}$	IMU residual (roll, pitch)	2	Local orientation error
	Joint position residual	29	Joint angle error to local target
$\boldsymbol{o}^{\text{proprio}}$	Root angular velocity	3	Base angular velocity
	IMU (roll, pitch)	2	Current orientation
	Joint positions	29	Current joint angles
	Joint velocities	29	Scaled by 0.05
	Previous action	29	Last action command
$\boldsymbol{o}^{\text{history}}$	Proprioceptive history	920	92 dims × 10 steps
Object Observations:			
$\boldsymbol{o}^{\text{task}}$	Object position residual	3	Local frame position error
	Object rotation residual	6	Tan-norm rotation error
	Object position	3	Local frame position
	Point cloud (PCA sampled)	192	64 points × 3 in head frame
Observation Masks:			
$\boldsymbol{m}$	Global goal mask	3	Keep probability for global goal
	Local goal mask	31	Keep probability for local goal
	Object masks	204	Trans(3) + rot(6) + pos(3) + pcd(192)
	Goal mask	4	Keep probability for command
Total		1496	

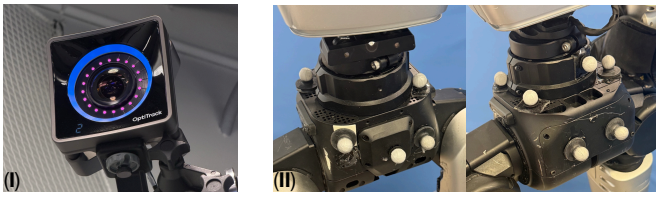


Fig. A: (**I**) The *OptiTrack* camera setup used to support ULTRA control in MoCap mode. (**II**) Markers attached to the humanoid root (front and back).

Fig. B: We visualize the objects used in our real-world deployment.

auxiliary masked-prediction losses, and (**V**) a prior-only action matching loss computed from an additional forward pass that disables the privileged encoder. We set the expert loss coefficient, auxiliary loss coefficient, and local goal prediction coefficient to 1.0. To stabilize training and encourage a useful prior, the KL coefficient increases from 0.001 to 0.1 with a cosine schedule. More details are presented in Table F.

RL Finetuning Training. We finetune the distilled student with PPO. Importantly, PPO is applied to the *deployable* student policy, *i.e.,* the prior-only policy that does not use privileged encoder inputs. And to keep the prior, we preserve $1/4$ of environment still for distillation update. The PPO objective follows the standard loss. To keep finetuning stable, we use a conservative update regime. First, we scale the overall PPO contribution by a small constant relative to the distillation objectives. Second, we apply a short warm-up schedule for the critic: the policy-gradient term is gradually enabled over an initial window of 100 training epochs, while the critic loss remains active throughout. This warm-up reduces abrupt distribution shift from the distilled policy and improves optimization stability in the early finetuning phase.

RL Finetuning Reward. At each step, we define the total reward as the sum of a dense goal-reaching term, a progress term, and auxiliary regularizers: the dense term encourages matching the object and root goals via an exponentially decayed function of their current distances, while the progress term rewards step-to-step reduction in those distances, which is clipped to prevent large spikes; importantly, both goal-related terms are visibility-gated by binary masks so that if the object or root goal is hidden by the observation mask, the corresponding reward contribution is set to zero, and we add a terminal success bonus when all visible goal constraints fall below preset thresholds. More details are discussed in Table H.

C. Additional Experimental Details.

Simulation Configuration. We summarize the simulation hyperparameters in Table M.

Domain Randomization and Observation Noise. We summarize the domain randomization settings for the humanoid and the object in Tables I and J, respectively. Observation noise is summarized in Table L. We additionally apply domain randomization and noise to egocentric perception, summarized in Table K.

TABLE H: **RL finetuning reward components**. We mask-gate goal-related terms: if a target is unobserved, we set its reward to zero and exclude it from success checks.

Term	Active when	Description
Goal-reaching	target visible	Reward proximity to the sampled goal (dense).
Progress	target visible	Reward step-to-step improvement toward the goal (clipped).
Terminal bonus	success on visible targets	Add a bonus when all visible thresholds are met.
Termination	always	Same as in Table C.
Smoothness	always	Mostly same as in Table C without reference related terms

TABLE I: Domain randomization ranges.

Category	**Parameter**	**Range**
Humanoid	Added base mass (kg)	$[-3.0, 3.0]$
	CoM offset (m)	$[-0.05, 0.05]$
	Motor strength scale	$[0.8, 1.2]$
Environment	Ground friction	$[0.5, 2.0]$
	Gravity perturbation (m/s^2)	$[-0.1, 0.1]$
	Gravity randomization interval	4 s
Perturbations	Max push velocity (m/s)	1.0
	Push interval	4 s
Action	Action delay buffer length	8 steps

TABLE J: Object domain randomization ranges.

Parameter	**Range**
Object mass (kg)	$[0.15, 1.5]$
Object mass rare (kg)	$[0.05, 0.15]$ (every 10 episodes)
Object CoM offset (m)	$[-0.05, 0.05]$
Object inertia scale	$[0.5, 2.0]$
Object friction	$[0.2, 1.2]$
Object restitution	$[0.0, 0.3]$
Object rolling friction	$[0.0, 0.05]$
Object torsion friction	$[0.0, 0.05]$

TABLE K: Point cloud domain randomization.

Parameter	**Value**
Gaussian noise std (m)	0.02
Point dropout probability	0.15
Outlier probability	0.05
Outlier max distance (m)	0.5
Depth noise scale	0.01
Density range	[0.5, 1.0]
Cluster noise std (m)	0.005
Scale range	[0.95, 1.05]
Translation noise (m)	0.02
Occlusion probability	0.1
Camera rotation noise (rad)	0.05
Camera position noise (m)	0.02

TABLE L: Observation noise scales.

Observation	**Noise Scale**
Joint positions (rad)	0.01
Joint velocities (rad/s)	0.1
Angular velocity (rad/s)	0.05
IMU	0.05
Root position (m)	0.05

TABLE M: Simulation configuration.

Simulation	**Parameter**	**Value**
Physics	Simulation substeps	1
	Control frequency inverse	17 ($\approx$59 Hz)
	Number of parallel envs	4096
	PhysX solver type	1 (TGS)
	Position iterations	4
	Velocity iterations	1
	Contact offset (m)	0.02
	Rest offset (m)	0.0
	Bounce threshold vel. (m/s)	0.2
	Max depenetration vel. (m/s)	1.0
Ground Plane	Static friction	1.0
	Dynamic friction	1.0
	Restitution	0.0

Real-World Deployment. Figure A illustrates the motion-capture system used to support MoCap-driven control, and Figure B shows the objects used in our real-world experiments.

Limitations. Our method still has several limitations. It can fail under out-of-distribution conditions, such as severe point-cloud occlusion that removes critical geometric cues. In real-world experiments, MoCap-driven control is sensitive to marker occlusions, which can introduce jitter and drift in the estimated object pose and cascade into unstable tracking. Performance can also degrade when a human operator specifies overly aggressive or inconsistent goals (*e.g.*, large, discontinuous target jumps or goals that are physically infeasible given the current contacts).