

EmbodyHOI: Embodiment-Aware Generation of Hand–Object Interactions for Humanoid Dexterous Manipulation

Chaitanya Chawla*, Kartik Sharma*, Ananya Bal, Laszlo A. Jeni, Guanya Shi
Carnegie Mellon University

United States

{cchawla, kartiksh, abal, laszloaj, guanyas}@andrew.cmu.edu

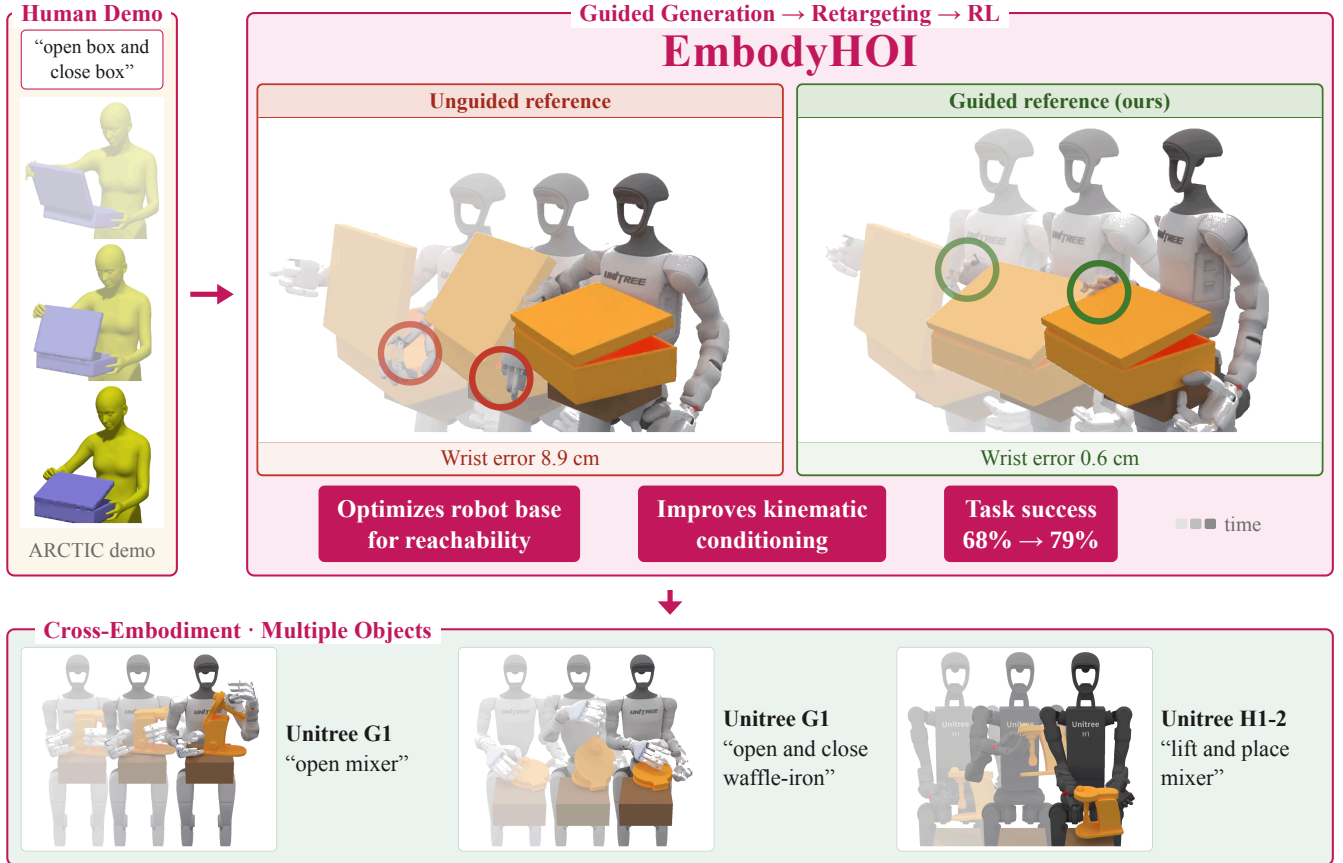


Fig. 1: EMBODYHOI steers a human-space HOI trajectory generation method toward motions that are more reachable for specific humanoid embodiments. During diffusion sampling in the HOI generation, a differentiable IK solve on the target robot scores wrist trajectories, and its gradient is fed back into the sampler while the robot’s base placement is optimized jointly. The generator is never retrained, so the same model and the same IK objective serve the Unitree G1 and H1-2.

Abstract—Humanoid dexterous manipulation needs reference motions that are both semantically rich and kinematically feasible for the target robot. Text-conditioned hand–object interaction (HOI) generators supply diverse, language-specified demonstrations, but they synthesize motion in human (MANO) space. The embodiment gap is therefore deferred to post-hoc retargeting, where trajectories deployed on the robot can fail to maintain contact with objects and also degrade policy learning. Our key idea is that *robot reachability* should constrain the interaction trajectories *while they are being generated*. We present EMBODYHOI, which runs a differentiable inverse-

kinematics (IK) solver for a specific humanoid embodiment inside the sampling loop of a frozen flow-matching HOI trajectory generator. We feed back gradients from the embodiment-aware IK solver into the HOI sampler and jointly optimize the position of the humanoid with respect to the object. Because guidance acts only at inference, one HOI trajectory generator trained on human hand poses transfers across humanoid embodiments without retraining. On ARCTIC trajectories with two contrasting humanoids (Unitree G1 and H1-2), our proposed guidance-based method cuts the retargeted wrist error from 8.9 cm to 0.6 cm while leaving text-to-HOI generation quality intact, and raises task success from 68% to 79% on 50 random G1 tasks and from 62% to 76% on a four-task H1-2 suite. Critically,

*Equal contribution.

cheaper alternatives for bridging the embodiment gap fail to beat EmbodyHOI. Post-hoc wrist refinement and compute-matched best-of- N selection both reach lower IK energy than our guidance yet train worse policies.

I. INTRODUCTION

Dexterous manipulation with multi-fingered hands remains difficult for humanoids that can reliably walk and loco-manipulate [1]–[4]. A robot must coordinate reaching, finger contact, and whole-arm motion under embodiment-specific kinematic constraints to execute trajectories that actually manipulate objects. A human trajectory when kinematically retargeted to a humanoid is often not executable, due to differences in finger morphology, joint limits, and wrist reachability.

We study how to *generate* human interactions which can easily close the above gap. Given a language prompt and an articulated object, we produce bimanual hand–object trajectories specific to a humanoid, which are easily executable by the robot.

A natural route to this goal of end-to-end policies is generate-then-track, but today’s Hand-Object Interaction (HOI) generators have no notion of the robot that will execute their output. In this route, a generator proposes diverse task-conditioned reference motions and a physics-based RL policy learns to track them under contact and actuation limits [5].

Hand-centric HOI generators are the natural source of dexterous references as they model finger-object contact, but they generate in MANO [6] space with no term that couples the wrist to a robot’s kinematic chain. This mismatch only surfaces during retargeting, after the finger and object motions are committed, and the tracking policy inherits wrist targets that the arm cannot reach. Previous works resolve this on either side of generation. Training the motion prior in robot space [7] yields one generator per robot, which rules out cross-embodiment reuse. Whereas repairing a generated human motion [8], [9] only makes delta modifications, and cannot always rectify retargeted wrist trajectories unreachable for the robot. Tracking systems such as DexMachina [10] and ManipTrans [11] sidestep the question by commanding a floating wrist with no arm.

We take a different approach: keep a frozen MANO HOI generator as a *reusable* motion prior and make its samples embodiment-aware *during generation* (Fig. 1). We build on MAD-HOI [12], a text-conditioned HOI generator that fills object and hand tokens over a sequence of unmasking steps, integrating a flow-matching ODE for the unmasked tokens at each step. Inside every ODE step, we decode the current wrist trajectories, solve a differentiable IK problem on the target humanoid, and add a gradient of the IK residual (*IK energy*) to the learned velocity field, in a manner of classifier guidance. A wrist correction made in one unmasking step conditions the finger and object tokens generated in later ones, so the interaction re-forms around the corrected wrist instead of being patched afterward.

A full humanoid raises another problem that floating hands and fixed-base arms do not, where the robot stands. The base

placement decides whether both arms stay reachable over the whole interaction, and a fixed-base IK cannot differentiate between a feasible interaction performed from the wrong standing position from an infeasible one. We therefore solve for the base translation together with the arm joints, and re-solve it at every unmasking step so that the base follows the partially generated motion. Finally, we retarget the guided motions and train a PPO tracking policy with separate imitation rewards for arm and hand, as they differ in error scale and contact requirements.

IK energy is a diagnostic, not a quantity to minimize on its own. Guidance costs $32\times$ a native sample (5 s to 160 s on an A100), paid once per reference, so we compare it with alternatives that lower the energy after generation at the same or lower cost. Post-hoc IK refinement of a finished sample reaches the lowest energy of any method, and compute-matched best-of- N selection also beats guidance on energy, yet both train worse policies (Table I). Refinement moves the wrists after the fingers and object are fixed, and selection returns a different interaction instead of correcting the requested one. Reachability has to be imposed while the interaction is still being generated.

Our contributions are as follows.

- *Training-free IK guidance for a frozen HOI generator.* A differentiable IK solve on the target humanoid runs inside every ODE step of the sampler, and its gradient steers the hand tokens. One objective with one set of weights serves the Unitree G1 and H1-2, and it also lowers the IK energy of a second generator, Text2HOI [13], without modification.
- *Base placement inside guidance.* Solving for the base removes near-singular G1 arm configurations (2.6% to 0.2% of test samples) even when applied once to a finished sample, and re-solving it at every unmasking step accounts for 2.8 of the 5.7 success-rate points that guidance adds over post-hoc base optimization on the four-task G1 suite (Table III).
- *Lower IK energy after generation does not give better references.* Post-hoc refinement and compute-matched best-of-32 selection reach lower energy than guidance (0.35 and 1.21 vs. 1.81), yet their policies succeed less often (32.8% and 74.0% vs. 79.7%, Table I).
- *Downstream gains on two humanoids.* Policies trained on guided references with decoupled hand and arm reward streams, raise success from 68% to 79% on 50 randomly drawn ARCTIC tasks on the G1 and from 62% to 76% on a four-task H1-2 suite. We will release the guidance code, the generated references and the trained policies.

II. RELATED WORK

EMBODYHOI draws on three lines of work, and none of them adapts the reference motion to a humanoid’s arm chain while the interaction is being generated.

Human-space HOI generation and test-time guidance.

Text-to-motion models generate full-body SMPL [14] motion with pose-space diffusion [15], [16] or masked-token

generation [17]. Full-body HOI methods [18], [19] model contact only coarsely at the body surface. Hand-centric methods parameterized by MANO [6] model fine-grained finger-object contact on datasets such as ARCTIC [20], GRAB [21], and OakInk [22], from grasp synthesis and interaction denoising [21], [23] to language-conditioned sequence generation [12], [13], [24]–[26]. Test-time guidance steers such generators toward keyframes, trajectories, and contacts, but always toward targets specified in human space [27], [28]. Our guidance has the same form but no human-space target: it scores the decoded wrist against a robot’s kinematic chain, so its optimum moves when the robot or its base changes.

Retargeting and RL from human demonstrations.

Human-to-robot retargeting typically optimizes joint angles against keypoint and end-effector errors [29], [30], and it underpins teleoperation [30], [31] and robot-space datasets [7]. Dexterous-hand retargeting [32]–[34] is a standard preprocessing step for RL from human demonstrations. ManipTrans [11] pretrains a hand-trajectory imitator and then fine-tunes a residual policy. DexMachina [10] uses retargeted joints and contacts as auxiliary rewards under a decaying virtual-object curriculum. DexMan [35] extends this recipe to RGB video, and HERMES [36] unifies heterogeneous motion sources for mobile bimanual manipulation. In all of them the reference is fixed before the robot is considered. ManipTrans and DexMachina command a floating 6-DoF wrist with no arm chain, and DexMan drives a G1 arm through IK but keeps the robot at the origin, so neither arm reachability nor base placement is ever optimized. We reuse their object-side recipe (Sec. III-C) and make the reference itself the variable.

Generative priors for policy learning. Example-guided RL [37], [38] extends to HOI imitation [39], [40] and to tracking-as-reward execution [5], [41]. The two works closest to ours resolve the embodiment mismatch after the motion exists. Chen et al. [9] guide an RL finger controller with a generative prior that emits only wrists, fixed before the robot is considered, and DemoDiffusion [8] denoises a retargeted human trajectory with a robot diffusion policy that stays close to that trajectory and so inherits its reachability. DreamControl-v2 [7] avoids retargeting by training in robot space, and fine-tuning a human-space generator on reachable data would likewise remove our inference cost; both need robot-feasible data for every embodiment, which a method like ours would have to produce, and yield one generator per robot. However, none of them keeps one frozen human-space generator: we resolve reachability inside its sampling loop and handle base placement for a full humanoid arm chain.

III. METHOD

EMBODYHOI makes a *frozen* HOI generator embodiment-aware at inference time (Fig. 2). The generator only needs to expose a decodable wrist pose and separate hand and object tokens (Sec. III-A). A differentiable IK energy on the target humanoid steers its sampler and solves for the robot’s base

(Sec. III-B), and we retarget the guided motion and train a PPO tracking policy on it (Sec. III-C).

A. Text-to-HOI Generation

We adopt MAD-HOI [12] as the generator and keep it frozen; Sec IV-D applies the same guidance to Text2HOI [13]. Given a text prompt and an articulated object mesh, MAD-HOI produces an object trajectory $O = (\mathbf{T}_{\text{obj}}^k, \mathbf{R}_{\text{obj}}^k, \alpha_{\text{obj}}^k)$ (position, orientation, and articulation angle at frame k) and a MANO motion $H_h = (\tau_h^k, \phi_h^k, \theta_h^k)$ for each hand $h \in \{L, R\}$: a wrist translation $\tau_h^k \in \mathbb{R}^3$, a wrist orientation $\phi_h^k \in SO(3)$, and 15 finger joint rotations θ_h^k , with rotations in 6D form. Motion is represented in a decoupled latent space with separate object, left-hand, and right-hand tokens. A masked autoregressive transformer generates these tokens over a sequence of unmasking steps, and a flow-matching head generates the tokens unmasked at each step. The head is a learned velocity field $\mathbf{v}_\theta(\mathbf{x}, t, \mathbf{z})$ that defines the ODE $\dot{\mathbf{x}} = \mathbf{v}_\theta(\mathbf{x}, t, \mathbf{z})$ over the latent $\mathbf{x}$ of these tokens, with flow time $t \in [0, 1]$ and a context $\mathbf{z}$ that the transformer computes from the already generated tokens and the text. A decoder $\mathcal{D}$ maps latents back to motion. Two properties of this generator make our method possible: the wrist pose can be decoded, differentially, at any point of sampling, and hand and object tokens are separate, so a gradient can be applied to one and withheld from the other. We use the ARCTIC dataset [20] (4.6k sequences, 11 articulated objects), whose bimanual articulated interactions are the hardest case for a humanoid arm chain.

B. Embodiment-Aware IK Guidance

Guidance acts on the wrists alone, through an IK energy on the target humanoid that also solves for the robot’s base. Of the generated motion, only the wrist poses $\tau_h^t \in \mathbb{R}^3$ and $\phi_h^t \in SO(3)$ for $h \in \{L, R\}$ constrain the arm. Nothing ties these trajectories to the deployment robot, so per-frame IK retargeting is often ill-conditioned under large embodiment mismatch. The manipulator Jacobian of arm h with joint vector $\mathbf{q}^t$, $\mathbf{J}_h(\mathbf{q}^t) = \partial \text{vec}(\mathbf{T}_{\text{ee},h}^t) / \partial \mathbf{q}^t \in \mathbb{R}^{6 \times n}$, loses rank near singular configurations. We monitor its minimum singular value $\sigma_{\min}(\mathbf{J}_h)$ and condition number $\kappa(\mathbf{J}_h)$ as proximity measures, and bias *generation* away from such configurations.

Guidance adds an energy gradient to velocity field. Following classifier guidance [42], adapted to a flow-matching velocity field, we subtract the gradient of an energy E during sampling,

$$\dot{\mathbf{x}} = \mathbf{v}_\theta(\mathbf{x}, t, \mathbf{z}) - w \nabla_{\mathbf{x}} E(\mathbf{x}), \quad (1)$$

with guidance weight $w \geq 0$. The energy is evaluated on the decoded wrists: $\mathcal{D}$ maps the current latent, together with the tokens already unmasked, to wrist trajectories (τ, ϕ) , which serve as task-space targets for an IK solve on the robot.

Joint arm and base IK. We solve for the base because a fixed base makes reachability a property of the trajectory alone: a feasible interaction performed from the wrong standing position is then indistinguishable from an infeasible one, and fixed-base IK rejects both.

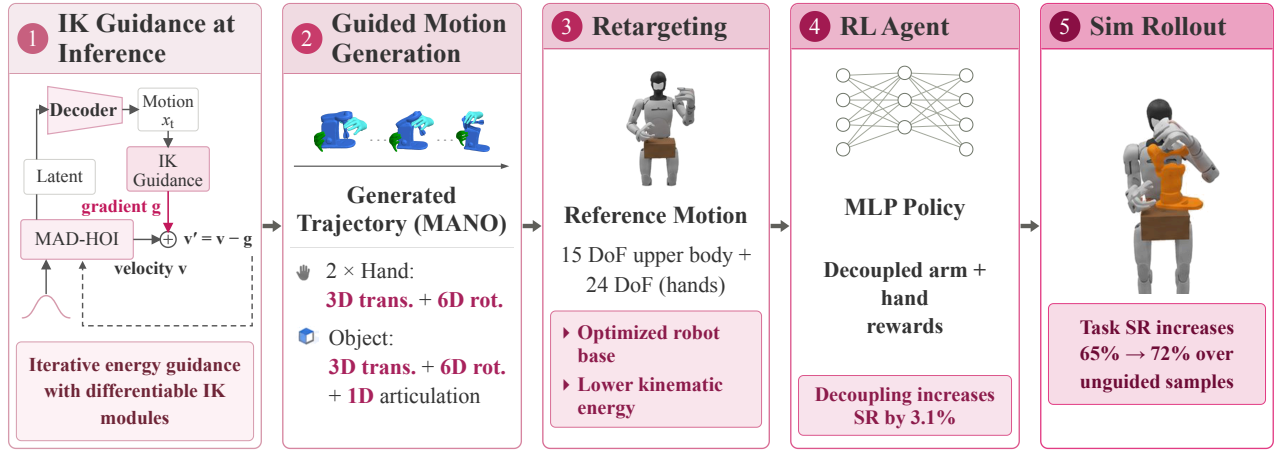


Fig. 2: EMBODYHOI makes a frozen HOI generator embodiment-aware at inference. Inside every unmasking step of the masked autoregressive generator, a differentiable IK solve on the target humanoid scores the decoded wrist trajectories and places the robot’s base; the gradient of the resulting IK energy is added to the flow-matching velocity field of the hand tokens, so the finger and object tokens generated in later steps form around reachable wrists. We then retarget the guided sample and train a PPO tracking policy on it with decoupled arm and hand rewards. Changing the robot changes only the IK module.

At each timestep we optimize $\mathbf{u}^t = [\mathbf{q}^t; \mathbf{b}^t]$, arm joints and a base translation $\mathbf{b}^t \in \mathbb{R}^3$ (base orientation fixed). With world-frame wrist target $\mathbf{T}_{w,h}^t$ built from (τ_h^t, ϕ_h^t) , the base transform $\mathbf{T}^b$ built from $\mathbf{b}^t$, and the forward kinematics $\text{FK}_h(\mathbf{q}^t)$ of wrist h in the base frame, the pose residual of arm h is

$$\mathbf{e}_h^t = \log((\mathbf{T}_{w,h}^t)^{-1} \mathbf{T}^b \text{FK}_h(\mathbf{q}^t)) \in \mathbb{R}^6 \quad (2)$$

weighted by w_{pos} on its translational part and w_{ori} on its rotational part. The regularizers complete the residual and fall into the two groups that Table III ablates. The *base-placement group* acts on $\mathbf{b}^t$: a base prior that penalizes displacement from the current preferred placement $\mathbf{b}_0$ (w_{root}), and a clearance term that keeps the base outside a fixed horizontal radius around the object (w_{clear}). The *arm-conditioning group* acts on $\mathbf{q}^t$: a rest-pose prior (λ_{rest}), a joint-limit penalty (w_{joint}), a penalty on deviation from the previous solve’s configuration, which also warm-starts the solve (w_{smooth}), and a null-space manipulability step (λ_{manip}). The null-space step runs after each solve. It takes a few gradient steps on $1/\sigma_{\min}(\mathbf{J}_h)$, where $\mathbf{J}_h(\mathbf{q}) \in \mathbb{R}^{6 \times n}$ is the geometric Jacobian of wrist h and $\sigma_{\min}$ its minimum singular value, projected onto the null space of both wrist tasks; it thus improves manipulability without moving either wrist, and the refined configuration seeds the next solve. Because $\mathbf{J}_h$ loses rank at singular configurations, $\sigma_{\min}$ and the condition number $\kappa(\mathbf{J}_h)$ are also the conditioning measures we report in Sec. IV.

Stacking residuals into $\mathbf{r}(\mathbf{u})$ and solving with Levenberg–Marquardt (10 iterations) gives $\mathbf{u}^{t*}$, and the IK energy is the mean squared residual over the horizon of T timesteps,

$$E(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T \|\mathbf{r}(\mathbf{u}^{t*})\|_2^2. \quad (3)$$

All weights are set once and shared by both embodiments,

both generators, and all tasks (Table IV).

Although the solve optimizes a base translation per timestep, the base is fixed during execution, so we average the per-timestep estimates into a single execution pose $\mathbf{b}_0 \leftarrow \text{mean}_t(\mathbf{b}^{t*})$, which becomes the root prior for the next unmasking iteration (last line of Alg. 1). The prior starts from one hand-set placement per robot and settles as the sequence fills in: its median change per step falls from about 1.3 cm in the first unmasking steps to 0.06 cm in the last. Because the base is solved per robot, the same generator can serve robots of different proportions: a 45 cm G1 arm and a 68.5 cm H1-2 arm must stand in different places for the same interaction (Sec. IV-D).

Gradients through the solver. We differentiate implicitly through the LM solve in JAX, so $\partial E / \partial(\tau, \phi)$ is defined without unrolling the iterations, and a JAX–PyTorch bridge propagates it through $\mathcal{D}$ to the latent $\mathbf{x}$. Before the update, we zero the gradient on object tokens ($\mathbf{m}_{\text{hand}}$ in Alg. 1), so guidance never acts on object tokens directly and the generator’s decoupled object–hand structure is preserved. Object and finger tokens generated in later unmasking steps still condition on the guided hand tokens, which is how the interaction re-forms around the corrected wrists.

Guidance must act during generation. We inject guidance into the inner ODE of every unmasking step (Alg. 1) and refresh the base prior after each. Guidance therefore edits the wrist while finger and object tokens are still being generated, so the interaction adapts to the new wrist; it changes the sampling distribution instead of ranking its draws. Section IV-A tests two alternatives that lower the same energy after generation, post-hoc refinement of a finished sample and best-of- N selection: both reach lower E than guidance, and both train worse policies.

Algorithm 1 One outer unmasking step with IK guidance

Require: partial latents $\mathbf{L}$, outer step s , IK module $\mathcal{G}$, weight w , base prior $\mathbf{b}_0$

- 1: sample mask $\mathcal{M}$ (cosine schedule); build $\mathbf{L}_{\text{mask}}$
- 2: $\mathbf{z}_{\mathcal{M}} \leftarrow \text{MAD-HOI}(\mathbf{L}_{\text{mask}}, \mathbf{c}_{\text{text}})$; $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 3: **for** flow time $t = 0 \rightarrow 1$ (RK4, 20 steps) **do**
- 4: $\mathbf{v} \leftarrow \text{MLP}(\mathbf{x}, t, \mathbf{z}_{\mathcal{M}})$; $(\boldsymbol{\tau}, \boldsymbol{\phi}) \leftarrow \mathcal{D}(\mathbf{x}; \mathbf{L}_{\text{mask}})$
- 5: $E \leftarrow \mathcal{G}(\boldsymbol{\tau}, \boldsymbol{\phi}; \mathbf{b}_0)$ $\triangleright$ joint arm+base LM solve, Eq. (3)
- 6: $\mathbf{x} \leftarrow \text{ODEStep}(\mathbf{x}, \mathbf{v} - w \nabla_{\mathbf{x}} E \odot \mathbf{m}_{\text{hand}}, t)$
- 7: **end for**
- 8: $\mathbf{L}[\mathcal{M}] \leftarrow \mathbf{x}$
- 9: **if** $s \geq s_{\text{start}}$ **then** decode $\mathbf{L}$; $\mathbf{b}_0 \leftarrow \text{mean}_t(\mathbf{b}^{t*})$
- 10: **end if**

C. Retargeting and Policy Learning

Retargeting turns the guided sample into a robot reference. Unless stated otherwise, the target is a Unitree G1 with two XHands: 15 controllable upper-body DoFs (torso and both arms), fixed lower body, and 12 DoFs per hand. PyRoki [29] solves the guided wrist trajectories to arm joint trajectories from the optimized base; dex-retargeting [32] maps MANO fingers to XHand joints; and contact retargeting following DexMachina [10] yields the reference contacts $\hat{\mathbf{c}}_t$. Generated object motion supplies the reference object pose and articulation used in observations, rewards, and terminations.

The policy tracks the reference with PPO. We train an MLP policy with PPO [43] in Genesis. It observes short histories of joint positions, velocities, and actions; the episode phase; sparse future reference joint offsets $\hat{\mathbf{q}}_{t+\Delta}^{\text{robot}} - \mathbf{q}_t^{\text{robot}}$ for $\Delta \in \{1, 2, 4, 8, 16\}$; and the object’s part poses, articulation, and future tracking errors against the generated trajectory. Actions are 39-D: increments on the previous joint target for the 15 upper-body DoFs, which keep large arm motions smooth, and absolute targets for the 24 finger DoFs, whose joint ranges are small. We never actuate the object, so unlike DexMachina we use no virtual-object-controller curriculum.

Arm and hand get separate imitation rewards and terminations. Coarse free-space reaching and contact-rich finger control differ in error scale, so a single imitation term over all 39 DoFs biases optimization toward whichever subsystem dominates the error. We therefore split imitation into an upper-body reward $r_{\text{imi,upper}}$ over 15 DoFs and a dexterous reward $r_{\text{imi,dex}}$ over 24 DoFs, each a joint-tracking error against the retargeted reference with its own weight and temperature. The total reward adds an object-tracking term r_{task} on position, orientation, and articulation, a contact term that matches the retargeted contact locations, and standard penalties on action magnitude, action rate, joint velocity, joint acceleration, and torque. Terminations are decoupled in the same way, on the half-squared joint error $\frac{1}{2}\Delta q^2$: the upper-body error terminates on its *maximum* over joints, because a large arm failure ends the rollout, and the finger error on its *mean*, because an isolated finger error is not catastrophic. Episodes also end if the object falls below the table or its half-squared articulation error exceeds ϵ_a .

D. Implementation Details

The guided sampler runs 18 unmasking steps. Inside each, it integrates the flow ODE with RK4 in 20 steps and applies guidance with a constant weight $w=0.5$ at every ODE step, from the first unmasking step on ($s_{\text{start}}=0$). Each IK solve runs LM for 10 iterations with implicit differentiation. Appendix I-B lists the objective weights (Table IV) and solver settings (Table V), which are identical for both embodiments, both generators, and every task. Each policy trains for about 4 h on one RTX 4090. Appendix I-D gives the full RL specification: observations (Table VI), reward terms (Table VII), terminations (Table VIII), and all PPO, reward, penalty, threshold, and action-parameterization values (Table IX). The same RL settings are used for every condition in Tables I and III; only the reference differs.

IV. EXPERIMENTS

We answer four questions. (Q1) Does guidance *during* sampling earn its cost relative to post-hoc IK refinement and compute-matched best-of- N selection? (Sec. IV-A) (Q2) How does guidance change kinematic conditioning and generation quality? (Sec. IV-B) (Q3) How much of the downstream gain comes from each component? (Sec. IV-C) (Q4) Does the same objective transfer across embodiments and generators? (Sec. IV-D)

Protocol. Downstream results use two protocols on the G1 with XHands: a fixed four-task suite (box open, waffle-iron lift, mixer open+close, microwave lift+place; Tables I and III) and 50 randomly sampled tasks with no pre-selection. Of the latter, the 10 whose unguided reference falls below $\sigma_{\min} < 0.1$, and consequently carries high IK energy, form a hard subset (Sec. IV-C). Retargeting, rewards, and PPO follow Sec. III-C in every condition; only the reference differs. Q1 uses 51 seed-matched test prompts that include the four-task suite, Q2 400 ARCTIC test prompts, Q3 the four-task suite and the 50 random tasks, and Q4 the prompt sets stated in Sec. IV-D. We recompute every IK metric for every condition with one standalone instance of the IK solver of Sec. III-B (optimized base, two base-prior updates from the same initial prior), so the rows of a table are directly comparable.

Success requires the object’s position, orientation, and articulation to stay within 2 cm, 40° , and 10° of the reference at every step of the episode. Success rates (SR) are means over 5 training seeds, each evaluated on 10 episodes of B parallel environments with randomized object initialization per task (10 B rollouts per seed and task), with the standard error over seeds in Tables I and III. Kinematic metrics are the IK energy E (Eq. 3) and the worst-case Jacobian conditioning over the sequence (minimum singular value $\sigma_{\min}$, maximum condition number) on the target robot’s upper-body chain.

Generation quality uses the standard text-to-motion metrics (FID, KID, diversity, top-3 retrieval accuracy, matching score) and geometric metrics (contact ratio, penetration depth, jitter).

TABLE I: Guidance during sampling trains the best policies, although two alternatives reach lower IK energy. Medians over 51 seed-matched prompts. Cost: sampling time per reference in native samples ($t_1=5$ s); Δ_{wrist} : wrist displacement (cm) from the seed-matched unguided sample; Pen.: penetration loss; SR: success rate on the four-task suite, mean $\pm$ s.e.m. over 5 seeds.

Method	Cost	$E \downarrow$	Δ_{wrist} (cm)	Pen. $\downarrow$	SR % $\uparrow$
Unguided, fixed base	t_1	2.30	–	0.52	68.1 ± 0.3
+ base optimization	t_1	1.89	0	0.52	73.3 ± 0.5
+ IK guidance (ours)	$32 t_1$	1.81	0.3	0.53	79.7 ± 0.4
Post-hoc IK	$t_1 + \text{IK}$	0.35	32.6	0.59	32.8 ± 0.3
Best-of-8 by E	$8 t_1$	1.36	13.2	0.52	73.1 ± 0.6
Best-of-32 by E	$32 t_1$	1.21	13.7	0.51	74.0 ± 0.6

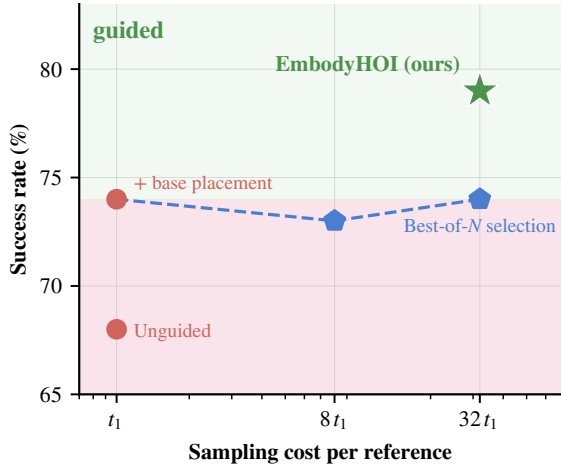


Fig. 3: Only guidance improves on base optimization. Success rate on the four-task suite against sampling cost per reference (Table I). Base optimization lifts the unguided baseline at no sampling cost; best-of- N selection does not measurably improve on it even at 32 times the cost (0.7 ± 0.8 points).

A. Is Guidance During Generation Necessary? (Q1)

Guidance costs $32\times$ a native sample, so each alternative in Table I receives the same budget, the same 51 prompts, and the same IK solver and base optimizer. **Base-placement** optimization alone keeps the sample and re-solves only the base. This cost is paid once per reference and amortized over ≈ 4 h of policy training it supports, so the wall-clock comparison is a worst case rather than a deployment constraint. **Post-hoc IK** refinement optimizes the wrist channels of a finished unguided sample against the same energy (Adam, 100 steps). **Best-of- N** draws N unguided samples with fixed seeds and keeps the one with the lowest terminal E under the same optimized base.

Base placement is the cheapest large gain. Solving for the base with a single unguided sample reduces the median energy from 2.30 to 1.89 and raises the success rate from 68.1 to 73.3 at no sampling cost (Fig. 3). Guidance adds a smaller, consistent correction on top: it lowers the energy

of the seed-matched sample on 40 of the 51 prompts (78%; median 1.81) and raises the success rate to 79.7.

IK energy is not the quantity to optimize. Energy is diagnostic only at fixed interaction content: it is meaningful for a seed-matched guided-vs-unguided pair, and meaningless across independently drawn samples. Post-hoc refinement reaches the lowest energy of any method ($E=0.35$) but moves the wrists by 33 cm after finger and object motion are fixed, raising penetration from 0.52 to 0.59 and collapsing the success rate to 32.8. Selection lowers the energy further than guidance at every budget: 1.36 at $N=8$ and 1.21 at the compute-matched $N=32$, and wins on 88–94% of prompts — 88% on the easiest third by unguided E , 94% on the hardest — so its advantage does not shrink where reachability is worst. (see Section II-B).

The reason is variance. Energy varies across samples of one prompt (s.d. 0.36) far more than guidance shifts it (median per-prompt shift 0.04), so drawing more samples finds a low-energy trajectory more easily than correcting one sample does. Yet the policies trained from selected references succeed less often than from guided ones (73.1 at $N=8$, 74.0 at $N=32$, vs. 79.7), a gap of 5.7 ± 0.7 points at matched compute. The energy measures wrist reachability only; it is blind to whether fingers and object still form a coherent interaction around the wrists it prefers.

Guidance corrects, selection re-draws. The selected best-of- N sample lies as far from the seed-matched one as any other sample (13 cm), whereas guidance moves the wrists of the seed-matched sample by 0.3 cm and leaves penetration and contact unchanged. Only guidance edits the sample already in hand, which is what matters when that sample is one a user chose. Base-placement optimization is therefore the baseline to report alongside guidance: it is free, selection is a no-op on top of it at either budget (-0.2 ± 0.8 at $N=8$, 0.7 ± 0.8 at $N=32$), and raising N from 8 to 32 does not help either (0.9 ± 0.8).

B. Kinematic Conditioning and Generation Quality (Q2)

Conditioning. Base placement and guidance remove different failure modes. On 400 ARCTIC test prompts, optimizing the base alone cuts the share of near-singular unguided samples ($\sigma_{\min} < 0.03$) from 2.6% to 0.2% and the median IK energy from 2.59 to 2.19 (Fig. 5); after that, $\sigma_{\min}$ (≈ 0.086) and the condition number (≈ 26) are as good as the chain allows, and guidance leaves them there. Guidance instead removes the residual error that survives retargeting: it lowers the energy on 83% of prompts, by 2.9% on average and by 8% on the hardest quartile (Fig. 4), and the G1’s wrist tracking error after retargeting falls from 8.9 cm and 20° to 0.6 cm and 4° (Table II). On the remaining 17% of prompts the energy rises by 1.2% at the median with the wrists within 0.5 cm of the unguided sample; on 2 of the 51 seed-matched prompts of Sec. IV-A the guidance step propagated through later unmasking rounds into a different trajectory. No sample is filtered, so the success rates in Secs. IV-A and IV-C include these cases. We do not measure a per-prompt relation between energy and success.

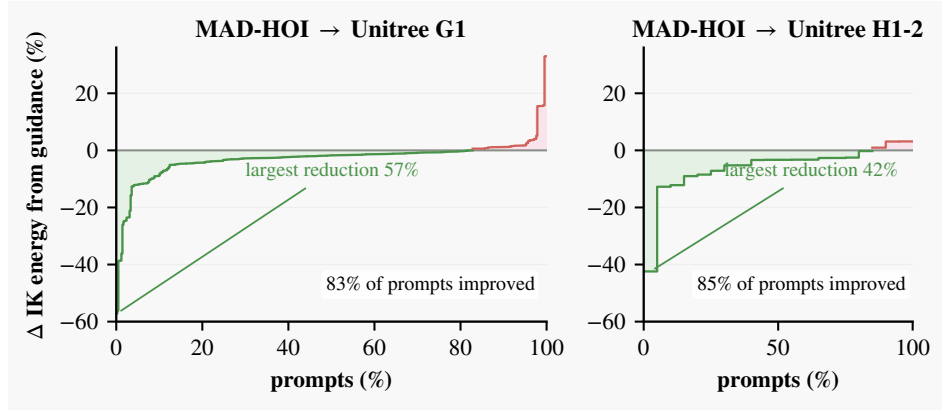


Fig. 4: Change in IK energy that guidance applies to each seed-matched sample. Guidance makes 83–97% of samples more reachable depending on the robot–generator pair; the remaining increases are small (median 1%), with two outliers on the G1 where sampling propagated the correction into a different trajectory (Sec. IV-B).

TABLE II: Guidance removes most of the retargeting error with little change in generation quality. Left: generation quality of unguided and IK-guided MAD-HOI on the *same* 400 ARCTIC test prompts with one evaluator. Right: wrist tracking error of the retargeted G1 after PyRoki IK on the randomly sampled tasks.

	FID↓	KID↓	Div.↑	Top-3↑	Match↓	Contact %↑	Pen. <1 cm %↓	Jitter↓	Wrist err. (cm)↓	Wrist err. (°)↓
Unguided	0.667	0.001	19.77	0.852	3.99	83.5	12.2	0.287	8.9	20.2
IK-guided	0.545	0.003	19.85	0.872	3.97	83.1	11.5	0.293	0.6	4.0

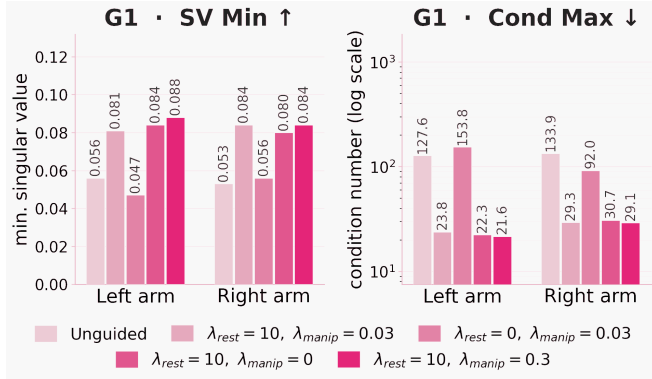


Fig. 5: Base placement, not the IK weights, removes near-singular configurations. Worst-case conditioning of the G1 per arm over the ARCTIC prompts: minimum Jacobian singular value (↑) and maximum condition number (↓, log scale). Unguided samples are solved at the fixed base; guided samples are solved at the optimized base under four IK-weight settings, whose distributions overlap.

Generation quality. Guidance does not degrade the prior (Table II). On the identical 400-prompt subset, guided samples have lower FID than unguided (0.545 vs. 0.667), better top-3 retrieval (0.872 vs. 0.852), unchanged contact ratio (83.1 vs. 83.5%) and diversity (19.85 vs. 19.77), lower penetration (11.5 vs. 12.2%), and marginally higher KID (0.003 vs. 0.001). Guidance necessarily integrates the flow ODE itself (20 RK4 steps, Alg. 1), so this compares our sampler against MAD-HOI’s released one rather than ablating the IK term alone, and the 0.12 FID gain belongs to the guided

sampler as a whole. The IK objective reshapes wrist trajectories toward reachable configurations without collapsing the semantic content of the interaction (see supplementary video).

Sensitivity to weights. The weights are set once before the sweep and frozen across all tasks and both robots (Sec. IV-D). Zeroing the rest-pose prior, zeroing or tripling the manipulability term, or scaling the root prior by 0.3 on the same 400 prompts moves the median IK energy by at most 0.03 and $\sigma_{\min}$ by less than 1% (Fig. 5), and every setting degrades FID (0.615–0.869 vs. 0.545) and top-3 retrieval (0.830–0.869 vs. 0.872). The manipulability term is the most sensitive of the four: tripling λ_{manip} accounts for the 0.869 end of the FID range. We did not retrain policies under perturbed weights, so downstream sensitivity is bounded only by Table III, where dropping a whole weight group costs 2.9 to 4.0 points.

C. Downstream Policy Learning and Ablations (Q3)

Randomly sampled tasks. Guided references raise the average success rate on 50 randomly sampled tasks from 68% to 79%, and on the 10 worst-conditioned of them — selected by $sv_{\min}$ of the **unguided** reference before any policy is trained — from 57.9% to 82.1%. The 50 tasks are drawn with no selection across all 11 ARCTIC objects, and we train policies from a guided and an unguided reference of each (5 seeds). The gain concentrates where reachability is poor: most randomly drawn prompts are already reachable, and the aggregate gain comes from the poorly conditioned minority (Sec. IV-A).

Controlled comparison and ablation. Two changes move downstream success on the four-task suite, and they are

TABLE III: Guidance, in-loop base placement, and the decoupled reward each raise success on the G1 four-task suite (box open, waffle-iron lift, mixer open+close, microwave lift+place); mean $\pm$ s.e.m. over 5 seeds, 10 episodes per task. Rows A–D form a ladder: A $\rightarrow$ B changes the reference source, B $\rightarrow$ C changes the RL recipe (DexMachina-style [10] vs. ours) and adds post-hoc base optimization, and C $\rightarrow$ D adds IK guidance with in-loop base placement. The lower half removes one component at a time from D. Row D and the IK-guidance row of Table I are independent training runs of the same configuration (79.3 vs. 79.7, within seed error).

Reference $\rightarrow$ retarget / RL	Avg. SR % $\uparrow$
A ARCTIC GT, fixed base $\rightarrow$ DexMachina-style	66.8 $\pm$ 0.3
B Unguided, fixed base $\rightarrow$ DexMachina-style	68.2 $\pm$ 0.9
C Unguided, optimized base $\rightarrow$ ours	73.6 $\pm$ 0.9
D Guided (both groups) $\rightarrow$ ours (full)	79.3 $\pm$ 0.6
D – arm-conditioning group	76.4 $\pm$ 0.3
D – base-placement group	75.3 $\pm$ 0.3
D, shared imitation reward	74.1 $\pm$ 0.6
D, shared reward, no contact retarg.	73.0 $\pm$ 0.4

comparable in size (Table III). A $\rightarrow$ B holds reachability fixed and varies only the reference source: swapping the ARCTIC demonstration for an unguided generated sample changes the success rate by 1.4 points, so generation supplies diversity but not reachability. B $\rightarrow$ C then introduces our RL recipe together with base placement and adds 5.4 points, and C $\rightarrow$ D adds IK guidance on top of it for a further 5.7. Neither half alone accounts for the method’s gain. The “DexMachina-style” rows re-implement their object-side recipe on our 15-DoF arm chain; they are not a run of their released floating-wrist system. Because that system commands a free-floating wrist with no arm chain, it never faces the reachability problem and is not a like-for-like baseline: its 86.1% on the same tasks is an upper bound obtained by assuming away the constraint we study, and the gap to our 79.3% is that constraint’s cost rather than a deficiency of the RL recipe.

The two weight groups of the IK objective contribute comparably to the 5.7 points that C $\rightarrow$ D adds. Zeroing one group at a time, dropping base placement costs 4.0 points (79.3 to 75.3) and dropping the four arm-conditioning terms costs 2.9 (79.3 to 76.4); read the other way, base placement alone adds 2.8 over row C and arm conditioning alone 1.7, so the two together achieve more than the sum of their parts. The comparison against row C is the one that matters for the design: row C applies the same base optimization once, after generation, so re-solving the base inside the sampling loop rather than after it (Sec. III-B) accounts for 2.8 of those points. Base placement is therefore not preprocessing that any pipeline could bolt on, but a term in the guidance objective whose value comes from being inside the loop. A shared imitation reward costs 5.2 points, and dropping contact retargeting only costs 1.1 more.

Reward design. Averaged over the suite, imitation alone reaches 44.5%, the object-tracking term alone 58.2%, a single imitation term over all 39 DoFs 74.1%, and our

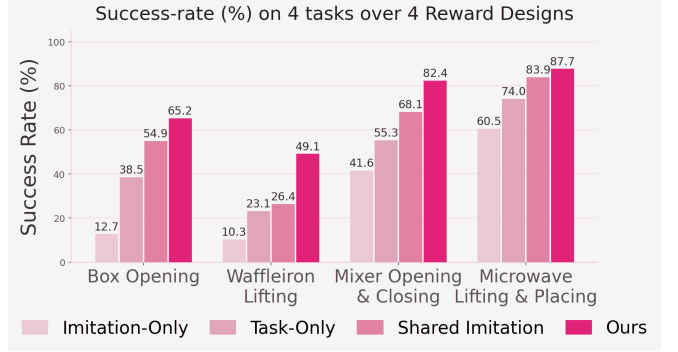


Fig. 6: Decoupled arm and hand imitation rewards give the highest success rate on every task of the G1 four-task suite. Per-task success rate under four reward configurations: imitation only, object tracking only, one imitation term over all 39 DoFs, and our decoupled rewards.

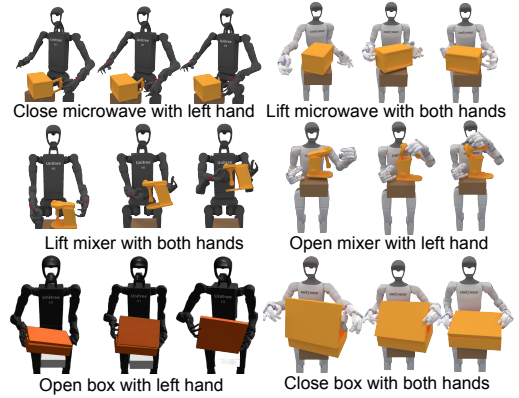


Fig. 7: The same generator and IK objective place two humanoids differently: from the same initial placement, guidance moves the H1-2 base (left; 178 cm, 68.5 cm arms) away from the object to avoid over-extension and the G1 base (right; 132 cm, 45 cm arms) closer and higher to stay reachable.

decoupled arm and hand streams 79.3%. Decoupling wins on every task, with the largest margins where coarse reaching and finger contact differ most in error scale. Figure 6 gives the per-task breakdown behind the aggregate reward-design numbers.

Feasibility beyond kinematics. Guidance does not reason about dynamics, but reachable references are also closer to executable ones: across the prompts of Sec. IV-A, guided references have a lower median peak-velocity ratio against the G1’s joint limits than their seed-matched unguided counterparts (0.20 vs. 0.24).

D. Transfer Across Embodiments and Generators (Q4)

As guidance acts at inference on a frozen generator, shifting to a new robot only needs change in the IK module. On the Unitree H1-2, whose arms are 68.5 cm long compared to G1’s 45 cm, guidance lowers the IK energy of the seed-matched sample on 46 of 50 prompts (median 2.06 $\rightarrow$ 1.95, mean -6%) while moving wrists by 0.3 cm. Both conditions

stay far from singularity ($\sigma_{\min} \approx 0.13$) because H1-2’s longer arms are closer to human proportions.

Notably, the base moves in the opposite direction to G1’s (Fig. 7): the longer arms over-extend and must stand back, where the G1’s must close in. No term in the objective encodes either behaviour; both fall out of the same residual.

The gain carries through to policy learning: on the same four-task suite, guidance raises H1-2 success rate from $62.3 \pm 0.4\%$ to $75.7 \pm 0.5\%$ (5 seeds, 10 episodes per task), a 13.4-point gain comparable to the G1’s 11.6 points (68.1 to 79.7, Table I) despite the different arm geometry.

The module is also generator-agnostic. Text2HOI [13] represents the wrist pose identically to MAD-HOI, so the IK objective and the base prior transfer without modification. Applied to its DDPM sampler as a gradient step on prediction x_0 , the guidance lowers the IK energy on 48 of 50 ARCTIC prompts (median $2.36 \rightarrow 2.34$, mean -1.3%) with at most 0.4 cm of wrist adjustment. The same objective on the H1-2 lowers Text2HOI’s energy on all 50 ARCTIC prompts (median $1.77 \rightarrow 1.75$, mean -1.2%), again with the wrists within 0.5 cm. The effect is smaller than on MAD-HOI because Text2HOI’s contact refiner post-processes the hands, but its direction is consistent across generators and robots. We therefore do not depend on MAD-HOI: its checkpoint is public [12], we will release the guidance code, all generated references, and the trained policies.

V. CONCLUSION AND LIMITATIONS

EMBODYHOI helps a frozen HOI generator in producing motions a target humanoid can reach. The key component is running a differentiable IK solve inside its sampling loop and optimizing the robot’s base. Policies trained on the guided references succeed more often than on unguided ones (79% vs. 68% on 50 random tasks, 76% vs. 62% on the H1-2). Base placement being the larger of the two contributors (4.0 vs. 2.9 points) and also the cheaper, since it adds no sampling cost.

More broadly, our comparison against post-hoc refinement and best-of- N selection is a caution for reference-generation pipelines in general: a kinematic feasibility metric is a valid diagnostic only when the rest of the interaction is held fixed, and driving it down directly buys feasibility at the cost of the interaction it was meant to preserve.

Guidance is kinematic: better reachability does not imply dynamic or contact feasibility, and the RL stage absorbs what guidance does not model. The cost is a $32\times$ slower sampler, paid once per reference; restricting guidance to a subset of the refinement steps would reduce it.

Our scope is limited in two ways. The lower body is fixed: we optimize where the robot stands, and not how it gets there. Integrating a locomotion controller [4], [44] is left for future work. All evaluation is performed in simulation on ARCTIC’s articulated objects; real-world rollouts are the immediate next step. Beyond scope, fine-tuning the generator on its own guided samples would distill the guidance into the prior and remove the inference cost, at the price of one generator per embodiment.

ACKNOWLEDGMENTS

LLMs were used for copy-editing and debugging scripts; all technical content, experiments, and analyses are the authors’ own.

REFERENCES

- [1] 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 *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9989–9996, IEEE, 2025.
- [2] 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.
- [3] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, D. J. Yoon, R. Hoque, L. Paulsen, *et al.*, “Humanoid policy~ human policy,” *arXiv preprint arXiv:2503.13441*, 2025.
- [4] 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.
- [5] D. Kalaria, S. S. Harithas, P. Katara, S. Kwak, S. Bhagat, S. Sastry, S. Sridhar, S. Vemprala, A. Kapoor, and J. C.-K. Huang, “Dream-control: Human-inspired whole-body humanoid control for scene interaction via guided diffusion,” *arXiv preprint arXiv:2509.14353*, 2025.
- [6] J. Romero, D. Tzionas, and M. J. Black, “Embodied hands: Modeling and capturing hands and bodies together,” *ACM Transactions on Graphics (SIGGRAPH Asia)*, vol. 36, no. 6, 2017.
- [7] S. Harithas, S. Kwak, P. Katara, S. Deolasee, D. Kalaria, S. Sridhar, S. Vemprala, A. Kapoor, and J. C.-K. Huang, “Dreamcontrol-v2: Simpler and scalable autonomous humanoid skills via trainable guided diffusion priors,” *arXiv preprint arXiv:2604.00202*, 2026.
- [8] S. Park, H. Bharadhwaj, and S. Tulsiani, “Demodiffusion: One-shot human imitation using pre-trained diffusion policy,” *arXiv preprint arXiv:2506.20668*, 2025.
- [9] Y. Chen, C. Wang, Y. Yang, and C. K. Liu, “Object-centric dexterous manipulation from human motion data,” *arXiv preprint arXiv:2411.04005*, 2024.
- [10] Z. Mandi, Y. Hou, D. Fox, Y. Narang, A. Mandlekar, and S. Song, “Dexmachina: Functional retargeting for bimanual dexterous manipulation,” *arXiv preprint arXiv:2505.24853*, 2025.
- [11] K. Li, P. Li, T. Liu, Y. Li, and S. Huang, “Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6991–7003, 2025.
- [12] A. Bal, K. Sharma, E. Lai, S. Tiwari, L. Dahiya, C. Chawla, and L. A. Jeni, “Mad-hoi: Masked autoregressive diffusion for generating articulated hand object interactions from text,” *arXiv preprint arXiv:2608.10162*, 2026.
- [13] J. Cha, J. Kim, J. S. Yoon, and S. Baek, “Text2HOI: Text-guided 3D motion generation for hand-object interaction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [14] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, “SMPL: A skinned multi-person linear model,” *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 248:1–248:16, 2015.
- [15] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermato, “Human motion diffusion model,” in *International Conference on Learning Representations (ICLR)*, 2023.
- [16] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu, “Motiondiffuse: Text-driven human motion generation with diffusion model,” *arXiv preprint arXiv:2208.15001*, 2022.
- [17] C. Guo, Y. Mu, M. G. Javed, S. Wang, and L. Cheng, “MoMask: Generative masked modeling of 3D human motions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [18] S. Xu, Z. Li, Y.-X. Wang, and L.-Y. Gui, “InterDiff: Generating 3D human-object interactions with physics-informed diffusion,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [19] X. Peng, Y. Xie, Z. Wu, V. Jampani, D. Sun, and H. Jiang, “HOI-Diff: Text-driven synthesis of 3D human-object interactions using diffusion models,” *arXiv preprint arXiv:2312.06553*, 2023.

- [20] Z. Fan, O. Taheri, D. Tzionas, M. Kocabas, M. Kaufmann, M. J. Black, and O. Hilliges, “Arctic: A dataset for dexterous bimanual hand-object manipulation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12943–12954, 2023.
- [21] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas, “Grab: A dataset of whole-body human grasping of objects,” in *European conference on computer vision*, pp. 581–600, Springer, 2020.
- [22] L. Yang, K. Li, X. Zhan, F. Wu, A. Xu, L. Liu, and C. Lu, “OakInk: A large-scale knowledge repository for understanding hand-object interaction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [23] X. Liu and L. Yi, “GeneOH diffusion: Towards generalizable hand-object interaction denoising via denoising diffusion,” in *International Conference on Learning Representations (ICLR)*, 2024.
- [24] M. Li, S. Christen, C. Wan, Y. Cai, R. Liao, L. Sigal, and S. Ma, “Latenthoi: On the generalizable hand object motion generation with latent hand diffusion,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17416–17425, 2025.
- [25] Z. Zhang, Y. Shi, L. Yang, S. Ni, Q. Ye, and J. Wang, “Openhoi: Open-world hand-object interaction synthesis with multimodal large language model,” *Advances in Neural Information Processing Systems*, vol. 38, pp. 166582–166612, 2026.
- [26] S. Christen, S. Hampali, F. Sener, E. Remelli, T. Hodan, E. Sauser, S. Ma, and B. Tekin, “DiffH2O: Diffusion-based synthesis of hand-object interactions from textual descriptions,” in *SIGGRAPH Asia Conference Papers*, 2024.
- [27] K. Karunratanakul, K. Preechakul, S. Suwajanakorn, and S. Tang, “Guided motion diffusion for controllable human motion synthesis,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [28] Y. Xie, V. Jampani, L. Zhong, D. Sun, and H. Jiang, “Omnicontrol: Control any joint at any time for human motion generation,” in *International Conference on Learning Representations*, vol. 2024, pp. 28176–28194, 2024.
- [29] C. M. Kim, B. Yi, H. Choi, Y. Ma, K. Goldberg, and A. Kanazawa, “Pyroki: A modular toolkit for robot kinematic optimization,” in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1312–1319, IEEE, 2025.
- [30] 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.
- [31] Z. Luo, Y. Yuan, T. Wang, C. Li, S. Chen, F. Castaneda, 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.
- [32] Y. Qin, W. Yang, B. Huang, K. Van Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox, “Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system,” *arXiv preprint arXiv:2307.04577*, 2023.
- [33] 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.
- [34] K. Shaw and D. Pathak, “Leap hand v2: Dexterous, low-cost anthropomorphic hybrid rigid soft hand for robot learning,” in *2nd Workshop on Dexterous Manipulation: Design, Perception and Control (RSS)*, 2024.
- [35] J. Hsieh, K.-H. Tu, K.-H. Hung, and T.-W. Ke, “Dexman: Learning bimanual dexterous manipulation from human and generated videos,” *arXiv preprint arXiv:2510.08475*, 2025.
- [36] Z. Yuan, T. Wei, L. Gu, P. Hua, T. Liang, Y. Chen, and H. Xu, “Hermes: Human-to-robot embodied learning from multi-source motion data for mobile dexterous manipulation,” *arXiv preprint arXiv:2508.20085*, 2025.
- [37] 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.
- [38] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, “Amp: Adversarial motion priors for stylized physics-based character control,” *ACM Transactions on Graphics (ToG)*, vol. 40, no. 4, pp. 1–20, 2021.
- [39] Y. Wang, J. Lin, A. Zeng, Z. Luo, J. Zhang, and L. Zhang, “Physhoi: Physics-based imitation of dynamic human-object interaction,” *arXiv preprint arXiv:2312.04393*, 2023.
- [40] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12266–12277, 2025.
- [41] Z. Luo, J. Cao, S. Christen, A. Winkler, K. Kitani, and W. Xu, “Omni-grasp: Grasping diverse objects with simulated humanoids,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 2161–2184, 2024.
- [42] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.
- [43] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.
- [44] Y. Zhang, Y. Yuan, P. Gurunath, I. Gupta, S. Omidshafiei, A.-a. Aghamohammadi, M. Vazquez-Chanlatte, L. Pedersen, T. He, and G. Shi, “Falcon: Learning force-adaptive humanoid loco-manipulation,” *arXiv preprint arXiv:2505.06776*, 2025.
- [45] D. Makoviichuk and V. Makoviyuchuk, “rl-games: A high-performance framework for reinforcement learning,” May 2021.
- [46] G. Authors, “Genesis: A universal and generative physics engine for robotics and beyond,” December 2024.

APPENDIX I IMPLEMENTATION DETAILS

A. Flow Matching and Energy Guidance

This section expands the sampling formulation summarized in Sec. III-B, so that the guidance derivation is self-contained for readers unfamiliar with flow matching.

Flow-matching sampler. MAD-HOI [12] generates continuous motion latents with a flow-matching objective. Rather than learning a sequence of discrete denoising transitions, it learns a time-dependent velocity field $\mathbf{v}_\theta(\mathbf{x}, t, \mathbf{z})$ that transports samples from a simple noise distribution toward the data distribution, conditioned on context $\mathbf{z}$. Under a linear interpolation (coupling) path between clean latents and Gaussian noise, the learned field defines the probability-flow ODE

$$\dot{\mathbf{x}} = \mathbf{v}_\theta(\mathbf{x}, t, \mathbf{z}), \quad (4)$$

whose integration from $t=0$ to $t=1$ transforms noise into a motion sample. The context $\mathbf{z}$ is produced by the transformer from the currently unmasked tokens and the text prompt.

Energy guidance. Following classifier-style guidance for diffusion and flow models [42], we bias sampling with an auxiliary energy $E(\mathbf{x})$,

$$\dot{\mathbf{x}} = \mathbf{v}_\theta(\mathbf{x}, t, \mathbf{z}) - w \nabla_{\mathbf{x}} E(\mathbf{x}), \quad (5)$$

where $w \geq 0$ controls guidance strength. The gradient term steers trajectories toward lower-energy regions while preserving the generative prior encoded by the learned flow. In our method $E(\mathbf{x})$ is the inverse-kinematics residual of Eq. (3), evaluated on the target humanoid, and guidance is applied only at inference: the generator’s weights are never updated.

We use a constant guidance weight $w=0.5$ rather than a schedule $w(t)$. We integrate Eq. (5) with a fourth-order Runge–Kutta solver using 20 steps per unmasking round. Unguided baselines are drawn with MAD-HOI’s released sampler, so the comparison in Table II evaluates the guided sampler as a whole (RK4 integrator plus energy term) against the released one rather than isolating the energy term.

Masking. The gradient is applied only to hand tokens, through the elementwise mask $\mathbf{m}_{\text{hand}}$ of Alg. 1. Object tokens receive no guidance gradient, so the generated object trajectory and articulation are bit-identical between a guided and its seed-matched unguided sample. This is what makes the object-side reference, and therefore the task reward, directly comparable across the two conditions.

B. IK Optimization Details

Objective. The guidance energy is computed from a differentiable IK optimization on the target humanoid. The optimization combines a primary wrist-pose tracking objective with regularization terms that improve kinematic feasibility and base consistency. Table IV lists the components and their weights. All weights are fixed once and shared across both embodiments, both generators, and every task; Sec. IV-B reports a sensitivity sweep over four of them.

TABLE IV: Components of the IK guidance objective. Wrist tracking is the primary task; the remaining terms regularize toward kinematically feasible arm and base configurations. The null-space term is a secondary optimization applied after the primary solve.

Component	Weight	Description
Wrist position	$w_{\text{pos}}=250$	Tracks the target wrist position.
Wrist orientation	$w_{\text{ori}}=5$	Tracks the target wrist orientation.
Rest-pose prior	$\lambda_{\text{rest}}=10$	Encourages solutions near a nominal arm configuration.
Joint-limit penalty	$w_{\text{joint}}=100$	Penalizes configurations outside joint limits.
Smoothness	$w_{\text{smooth}}=1$	Penalizes deviation from the previous timestep’s solution.
Root prior	$w_{\text{root}}=100$	Keeps the base near the current preferred placement $\mathbf{b}_0$.
Object clearance	$w_{\text{clear}}=100$	Penalizes arm configurations violating a minimum clearance radius around the object.
Null-space posture	$\lambda_{\text{manip}}=0.03$	Secondary objective improving manipulability without moving the wrist.

TABLE V: Optimization parameters of the IK guidance module.

Parameter	Value
LM iterations per timestep	10
LM damping coefficient	10^{-6}
Gradient-norm termination threshold	10^{-2}
Implicit differentiation	enabled
Solve variables $\mathbf{u}^t$	arm joints $\mathbf{q}^t$ + base translation $\mathbf{b}^t$
Base orientation	fixed
Base-prior refresh start s_{start}	0 (first unmasking step)
Base-prior updates per generation	18 (one per unmasking step)

Null-space optimization. Wrist-pose tracking defines the primary task; the remaining terms regularize toward stable, feasible arm and base configurations, and the clearance term discourages arm configurations that pass close to the manipulated object. After the primary Levenberg–Marquardt solve we apply the null-space posture objective, which adjusts the arm configuration without altering the achieved wrist pose. This yields postures better conditioned for downstream retargeting and policy learning, and is the component ablated as “arm conditioning” in Table III.

Solver settings. Table V lists the optimizer parameters, which are identical across all embodiments and tasks. A small iteration budget is sufficient for stable wrist solutions and keeps guidance runtimes practical: the solve runs once per RK4 step, so its cost is multiplied by 20 per unmasking round (Appendix III).

Gradients through the solver. We differentiate implicitly through the LM solve in JAX, so $\partial E / \partial(\tau, \phi)$ is defined without unrolling the iterations, and a JAX–PyTorch bridge propagates the gradient through the decoder $\mathcal{D}$ to the latent $\mathbf{x}$. Implicit differentiation keeps memory constant in the iteration count, which is what makes running the solve inside every flow step tractable.

C. Retargeting Details

Generated sequences are expressed in MANO hand parameters and object trajectories; policy learning requires robot joint trajectories. Wrist trajectories are converted to upper-body joint configurations with PyRoki [29], which optimizes joint states to match target wrist positions and orientations subject to the embodiment’s kinematic limits, solved from the base placement produced by guidance rather than from a fixed origin. MANO finger articulations are mapped to the XHand embodiment with dex-retargeting [32]. We additionally apply contact retargeting following Dex-Machina [10] to obtain reference contact locations $\hat{\mathbf{c}}_t$ on the hand links, used as auxiliary supervision during policy learning. The generated object trajectory and articulation are used unchanged as the object-side reference in observations, rewards, and terminations.

Because guidance never edits object tokens (Appendix I-A), the only difference between a guided and a seed-matched unguided reference after retargeting is the arm and finger trajectory, and the base placement they are solved from.

D. RL Environment and Policy Setup

Setup. Given retargeted trajectories, we train a policy π to imitate the reference motion while completing the manipulation task. We use PPO [43] implemented in rl-games [45], training in Genesis [46] on a single NVIDIA RTX 4090 (24 GB) with 8192 parallel environments for 2000 PPO iterations (≈ 4 h per policy).

Observations. The policy observes short histories of robot joint positions and velocities ($\mathbf{q}_t^{\text{robot}}, \dot{\mathbf{q}}_t^{\text{robot}}$) and previous actions $\mathbf{a}_t$ over the past nine steps, the normalized episode phase ϕ_t , and sparse future reference offsets $\hat{\mathbf{q}}_{t+\Delta}^{\text{robot}} - \mathbf{q}_t^{\text{robot}}$ for $\Delta \in \{1, 2, 4, 8, 16\}$, which encourage anticipatory tracking. For the articulated object it observes both part poses, the articulation state, and future tracking errors against the generated object trajectory. Table VI gives the full specification.

Actions. The policy outputs 39-D actions: 15-D *delta* targets for torso and arms and 24-D *absolute* finger targets. Delta control on the upper body yields smoother large-scale motion, while absolute finger commands improve learning given the limited joint ranges of the XHand. Action parameterization details — delta scale, PLAI interpolation coefficient, and action delay — are listed in Table IX.

Rewards. The total reward is

$$\begin{aligned} r_t = & w_{\text{task}} r_{\text{task}} + w_{\text{bc}} r_{\text{imi,upper}} \\ & + w_{\text{finger}} r_{\text{imi,dex}} + w_{\text{contact}} r_{\text{contact}} \\ & - \lambda_{\text{act}} r_{\text{act}} - \lambda_{\text{rate}} r_{\text{act-rate}} \\ & - \lambda_{\text{vel}} r_{\text{vel}} - \lambda_{\text{acc}} r_{\text{acc}} - \lambda_{\text{ctrl}} r_{\text{ctrl}}, \end{aligned} \quad (6)$$

where r_{task} rewards agreement with the reference object pose and articulation, $r_{\text{imi,upper}}$ and $r_{\text{imi,dex}}$ are the decoupled arm and hand imitation streams of Sec. III-C, and r_{contact} rewards agreement with the retargeted contact locations. The remaining terms regularize actions, control effort, and

H1-2 — unguided active $\text{sv}_{\text{min}} < 0.0001$

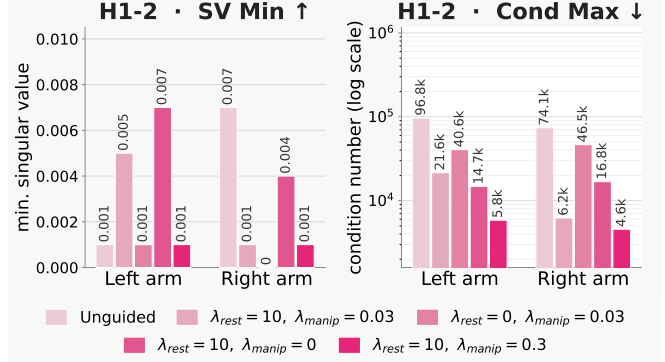


Fig. 8: Worst-case conditioning of the Unitree H1-2 per arm over the ARCTIC prompts: minimum Jacobian singular value ($\uparrow$) and maximum condition number ($\downarrow$, log scale). Compare Fig. 5 for the G1.

smoothness. Table VII defines each term and Table IX gives all weights and temperatures.

Terminations. Episodes end at the task-dependent horizon T_{max} , on NaN in the simulation state, when the object falls below the table, when object articulation departs the reference by more than ϵ_a , or when either imitation stream exceeds its threshold — the *maximum* over arm and torso joints for the upper body, the *mean* over finger joints for the hand. Table VIII states each condition; thresholds are applied to the half-squared joint error $\frac{1}{2}\Delta q^2$, so $\epsilon_{\text{upper}}=0.12$ corresponds to a 0.49 rad deviation on the worst joint.

Curricula. We never actuate the object, so unlike Dex-Machina [10] we use no virtual-object-controller curriculum and no reward curriculum. We have an additional entropy curriculum, which halves the entropy % coefficient and disables random state initialization once mean episode length exceeds 80% of the horizon). Without this curriculum, the policy can learn specific segments of the demonstration (lifting box, opening box), but fails to join the different segments into a single demonstration.

APPENDIX II

ADDITIONAL QUANTITATIVE RESULTS

A. Kinematic Analysis

Conditioning on the H1-2. Figure 8 reports the minimum Jacobian singular value and maximum condition number for the Unitree H1-2 under the protocol of Sec. IV-B. The trends are less pronounced than on the G1: because the H1-2’s 68.5 cm arms are closer to human proportions, both guided and unguided conditions remain far from singularity ($\sigma_{\text{min}} \approx 0.13$) and there are few near-singular configurations for base placement to remove. The benefit on this embodiment therefore comes from the residual reachability error rather than from singularity avoidance, which is consistent with the downstream gain reported in Sec. IV-D.

Observation	Interpretation
$\{\mathbf{q}_{t-k}^{\text{robot}}\}_{k=0}^8$	Robot joint positions over the past 9 steps (proprioception).
$\{\dot{\mathbf{q}}_{t-k}^{\text{robot}}\}_{k=0}^8$	Robot joint velocities over the past 9 steps.
$\{\mathbf{a}_{t-k}\}_{k=0}^8$	Previous actions over the past 9 steps ($n_a=39$ controlled DoFs).
ϕ_t	Normalized episode phase $2t/T_{\max} - 1$.
$\hat{\mathbf{q}}_{t+\Delta}^{\text{robot}} - \mathbf{q}_t^{\text{robot}}, \Delta \in \{1, 2, 4, 8, 16\}$	Future reference joint offsets (retargeted trajectory minus current joints).
$T_{t,k}^{\text{obj}}, R_{t,k}^{\text{obj}}, k \in \{1, 2\}$	3D position and 4D orientation of both articulated object parts.
α_t^{obj}	1D object articulation (joint opening).
$d_p, d_r, (\alpha_{t+h}^{\text{obj}} - \alpha_t^{\text{obj}}), h \in \{1, 2, 4, 8, 16\}$	Future tracking errors against the reference object pose ($d_p = \ T_t^{\text{obj}} - \hat{T}_{t+h}^{\text{obj}}\ _2, d_r = d_{\text{rot}}(R_t^{\text{obj}}, \hat{R}_{t+h}^{\text{obj}})$).
$\hat{\mathbf{p}}_{t+h}^{\text{obj}}, h \in \{1, 2, 4, 8, 16\}$	Reference object root positions at future timesteps.
$\hat{\mathbf{p}}_t^{\text{obj}} - \mathbf{p}_t^{\text{obj}}$	Current object root position error against the reference.

TABLE VI: Policy observations for bimanual manipulation on articulated objects. The reference quantities $\hat{\cdot}$ come from the generated (guided or unguided) trajectory, never from ground truth.

Reward term	Interpretation
$r_{\text{task}} = \left\ T_t^{\text{obj}} - \hat{T}_t^{\text{obj}} \right\ _2 \cdot d_{\text{rot}}(R_t^{\text{obj}}, \hat{R}_t^{\text{obj}}) \cdot \beta_a \left(\alpha_t^{\text{obj}} - \hat{\alpha}_t^{\text{obj}} \right)^2$	Rewards matching the reference object position, orientation, and articulation in the world frame.
$r_{\text{imi,upper}} = \left\ \mathbf{q}_t^{\text{upper}} - \hat{\mathbf{q}}_t^{\text{upper}} \right\ _2^2$	Upper-body imitation stream: tracking of the 15 torso and arm joints.
$r_{\text{imi,dex}} = \left\ \mathbf{q}_t^{\text{dex}} - \hat{\mathbf{q}}_t^{\text{dex}} \right\ _2^2$	Dexterous imitation stream: tracking of the 24 finger joints.
$r_{\text{contact}} = \left\ \mathbf{c}_t - \hat{\mathbf{c}}_t \right\ _2$	Rewards matching simulated contact locations to the retargeted reference contacts on hand links.
$r_{\text{act}} = \left\ \mathbf{a}_t \right\ _2$	Penalizes large actions.
$r_{\text{act-rate}} = \left\ \mathbf{a}_t - \mathbf{a}_{t-1} \right\ _2^2$	Penalizes large action rates.
$r_{\text{ctrl}} = \left\ \boldsymbol{\tau}_t \right\ _1$	Penalizes large joint control torques.
$r_{\text{vel}} = \left\ \dot{\mathbf{q}}_t \right\ _1$	Penalizes large joint velocities.
$r_{\text{acc}} = \left\ \ddot{\mathbf{q}}_t \right\ _1$	Penalizes large joint accelerations.

TABLE VII: Reward terms. The two imitation streams are kept separate rather than summed over all 39 DoFs; Table III shows this decoupling is worth 5.2 points of success rate.

Termination condition	Interpretation
$t \geq T_{\max}$	Ends the episode at the task-dependent horizon.
NaN in simulation state	Ends the episode on numerical failure in robot or object state.
$\max_i \frac{1}{2} \left(q_{t,i}^{\text{upper}} - \hat{q}_{t,i}^{\text{upper}} \right)^2 > \epsilon_{\text{upper}}$	Ends the episode when <i>any</i> arm or torso joint deviates too far: a large arm failure ends the rollout.
$\text{mean}_i \frac{1}{2} \left(q_{t,i}^{\text{dex}} - \hat{q}_{t,i}^{\text{dex}} \right)^2 > \epsilon_{\text{dex}}$	Ends the episode on <i>average</i> finger deviation: an isolated finger error is not catastrophic.
$p_{t,z}^{\text{obj}} < h_{\text{table}}$	Ends the episode when the object falls below the table.
$\frac{1}{2} \left(\alpha_t^{\text{obj}} - \hat{\alpha}_t^{\text{obj}} \right)^2 > \epsilon_a$	Ends the episode when object articulation departs the reference trajectory.

TABLE VIII: Episode termination conditions. Thresholds are applied to the half-squared joint error, so $\epsilon_{\text{upper}}=0.12$ corresponds to a 0.49rad deviation on the worst joint. The asymmetry between the maximum (arm) and the mean (hand) is the termination-side counterpart of the decoupled rewards.

Per-setting weight sweep. Sec. IV-B reports that perturbing the IK weights moves the median IK energy by at most 0.03 and $\sigma_{\min}$ by less than 1%, while every perturbation degrades generation quality (FID 0.615–0.869 vs. 0.545). Table X gives the per-setting numbers behind those ranges.

B. Additional Downstream Results

Stratified comparison of guidance and selection. Sec. IV-A states that best-of- N selection reaches lower IK energy than guidance on 88–94% of prompts regardless of difficulty. Table XI gives the per-bin numbers. Prompts

Category	Parameter	Value
<i>PPO / training</i>		
	Parallel environments B	8192
	PPO iterations	2000
	Rollout horizon (steps per env)	16
	Learning rate	3×10^{-4}
	Discount γ / GAE τ	0.99 / 0.95
	PPO clip ϵ	0.2
	Entropy coefficient	0.0055
	Bounds-loss coefficient	10^{-4}
	Minibatch size	$8B$
	PPO epochs per rollout	$\max(1, \lfloor 5B/4096 \rfloor)$
	Reward-shaper scale	0.1
	Policy MLP	512, 512, 256, 128 (ELU)
	Training hardware	$1 \times$ RTX 4090 (24 GB)
<i>Environment</i>		
	Episode length $T_{\max}$	task dependent
	Object demo horizons	$\{1, 2, 4, 8, 16\}$
	Robot history steps	$\{0, \dots, 8\}$
	Robot future steps	$\{1, 2, 4, 8, 16\}$
<i>Reward weights</i>		
	w_{task}	1.3
	w_{bc} (upper body)	1.0
	w_{finger}	0.15
	w_{contact}	0.7
<i>Reward shaping</i>		
	$\beta_p, \beta_r, \beta_a$	10, 1, 5
	β_{bc} (upper)	30
	β_{finger}	20
	β_{contact}	30
<i>Penalties</i>		
	λ_{act}	0.02
	λ_{rate}	0.003
	λ_{vel}	0.03
	λ_{acc}	0.025
	λ_{ctrl}	0.01
<i>Termination thresholds (on $\frac{1}{2} \Delta q^2$)</i>		
	ϵ_{upper}	0.12
	ϵ_{dex}	0.2
	ϵ_a (articulation)	0.4
<i>Action parameterization</i>		
	Upper body (15 DoF)	delta (PLAI)
	Fingers (24 DoF)	absolute
	Joint delta scale	0.05
	PLAI α	$[0.8, 1.0]$
	Action delay (steps)	$[0, 2]$

TABLE IX: RL hyperparameters. The same values are used for every condition in Tables I and III and for both embodiments; only the reference trajectory differs between conditions. Domain randomization and observation/action noise are omitted.

are split into terciles by the IK energy of the *unguided, optimized-base* sample, so the binning does not depend on the method under test. The lower block reports the two quantities that explain the pattern: sample-to-sample variance of E within a prompt, against the size of the shift guidance

applies. Selection exploits the former and guidance the latter, and the gap between them widens with difficulty — which is why selection’s energy advantage does not shrink on hard prompts even though its downstream advantage never appears at all.

TABLE X: Sensitivity to IK objective weights on the same 400 ARCTIC test prompts. The default setting is the one used in all other experiments and was fixed before this sweep was run.

Setting	Median $E \downarrow$	$\sigma_{\min} \uparrow$	FID $\downarrow$	Top-3 $\uparrow$
Default (ours)	1.81	0.082	0.545	0.872
$\lambda_{\text{rest}}=0$	2.05	0.054	0.615	0.869
$\lambda_{\text{manip}}=0$	1.94	0.082	0.650	0.861
$\lambda_{\text{manip}} \times 3$	1.78	0.086	0.869	0.835
$w_{\text{root}} \times 0.3$	1.88	0.083	0.810	0.830

TABLE XI: Selection’s energy advantage does not depend on difficulty. 51 prompts split into % terciles by unguided optimized-base E (bin edges $E < 1.80$, $1.80\text{--}3.00$, >3.00 ; % $n=17$ each).

Method	% Reaching lower E than EmbodyHOI			Median $E \downarrow$	SR % $\uparrow$
	Easy	Med.	Hard		
Best-of-8 by E	65%	88%	75%	1.36	73.1%
Best-of-32 by E	88%	94%	88%	1.21	74.0%
EmbodHOI (ref.)	–	–	–	1.81	79.7%

Where the downstream gain concentrates. Table XII splits the 50 randomly sampled tasks by the conditioning of their unguided reference. The subset boundary ($\sigma_{\min} < 0.1$) is fixed before any policy is trained. Most randomly drawn prompts are already reachable and guidance changes little on them; the aggregate gain is carried by the poorly conditioned minority, which is the population the method is designed for.

Reachability-free reference point. Sec. IV-C notes that DexMachina’s released floating-wrist system reaches 86.1% on the four-task suite. That system commands a free-floating 6-DoF wrist with no arm chain, so it never encounters the reachability problem we study; its number is an upper bound obtained by assuming away the constraint, not a baseline. We report it here so the gap to our 79.3% can be read as the cost of driving a real 15-DoF arm rather than as a deficiency of the RL recipe.

APPENDIX III COMPUTATIONAL COST

Table XIII summarizes the cost of each pipeline stage. Guidance is applied only at inference and adds no training cost to the generative model. The $32\times$ increase in sampling time is the method’s main practical cost; it is paid once per reference and amortized over the ≈ 4 h of policy training that reference supports, and it remains far cheaper than training an embodiment-specific motion generator, which additionally requires robot-feasible data for every new robot (Sec. II).

Restricting guidance to a subset of unmasking rounds, or to a subset of RK4 steps within each round, would reduce the sampling cost proportionally; we did not explore this tradeoff and note it as the most direct route to cheaper guidance (Sec. V).

TABLE XII: Success rate on the 50 randomly sampled G1 tasks, split by $\sigma_{\min}$ of the unguided reference; mean $\pm$ s.e.m. over 5 seeds.

Subset	n	Unguided SR %	Guided SR % $\uparrow$
All tasks	50	68.1	79.2
Well-conditioned ($\sigma_{\min} \geq 0.1$)	40	70.7	78.3
Hard ($\sigma_{\min} < 0.1$)	10	57.9	82.1

TABLE XIII: Approximate runtime per pipeline stage. Generation was measured on an NVIDIA A100 and PPO training on an RTX 4090. Runtimes are approximate and vary with sequence length and task.

Stage	Runtime	Hardware
MAD-HOI generation (unguided)	5 s	A100
IK-guided generation (ours)	160 s	A100
Post-hoc IK refinement	55 s	A100
Best-of-32 selection	160 s	A100
Retargeting per sequence	50 s	CPU
PPO training per policy	≈ 4 h	RTX 4090

APPENDIX IV ADDITIONAL QUALITATIVE RESULTS

Guided versus unguided generations. Figure 9 compares unguided MAD-HOI samples with our IK-guided samples for the same prompt and seed. Guidance reshapes wrist trajectories toward configurations the target humanoid can reach while preserving the semantic structure of the interaction — the grasp type, the contact locations, and the object motion are unchanged, consistent with the unchanged contact ratio and penetration in Table II. This is the static counterpart to the claim in Sec. IV-B that guidance does not collapse the semantic content of the interaction.

Additional executed tasks. See Figure 10 for more samples generated from our pipeline.

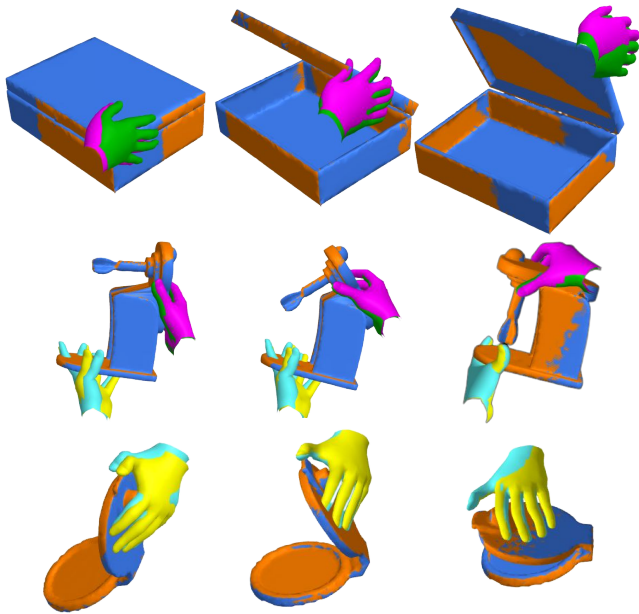


Fig. 9: Unguided MAD-HOI generations (left: cyan; right: green) against our IK-guided generations (left: yellow; right: magenta) for matched prompts and seeds. Guidance steers wrist trajectories toward kinematically feasible configurations for the target embodiment while preserving the structure of the hand-object interaction.

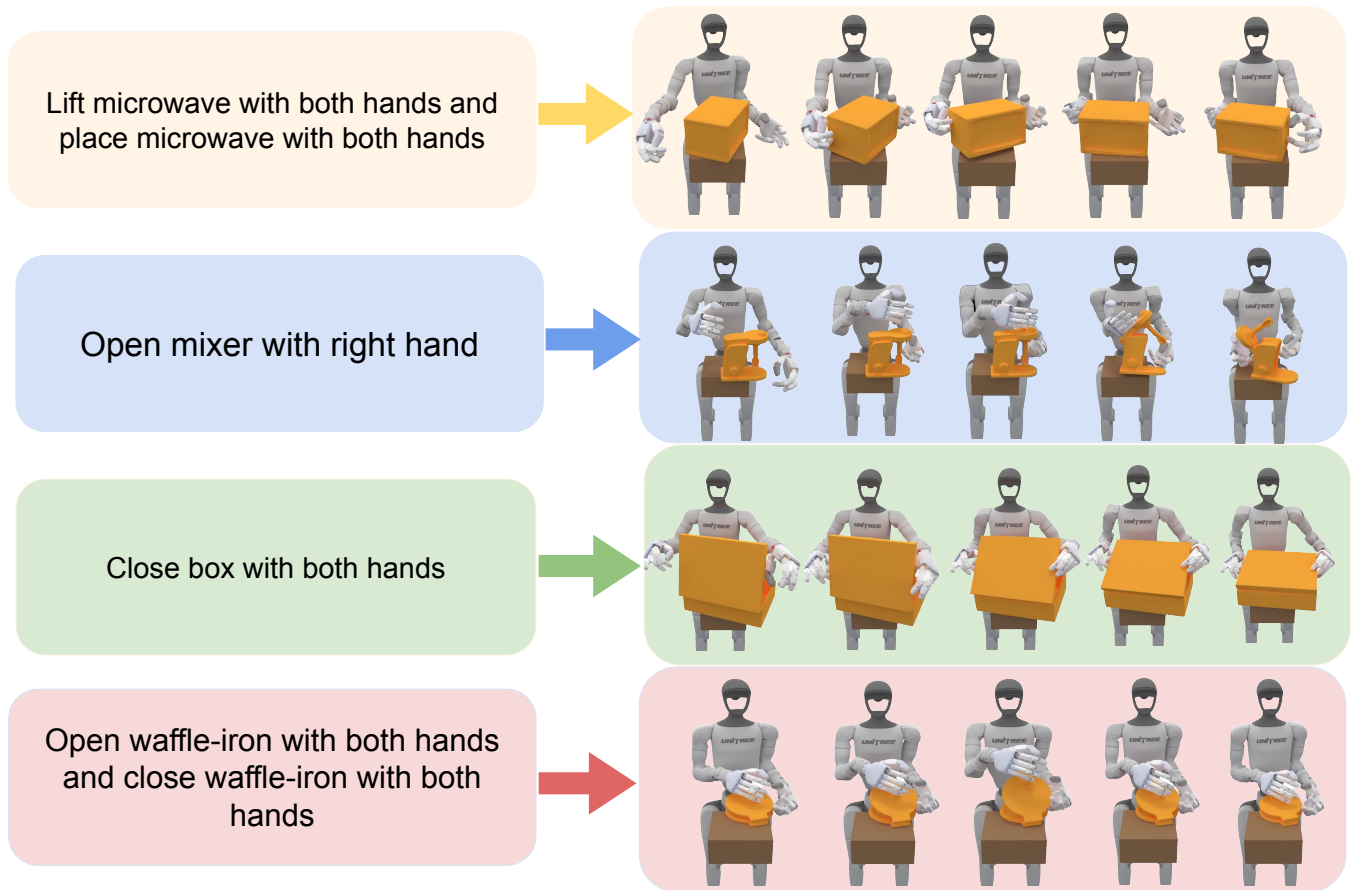


Fig. 10: Additional articulated-object manipulation tasks executed on the Unitree G1 by policies trained from EMBODYHOI references, spanning opening, closing, lifting, and bimanual behaviors across the ARCTIC object set.