CONTACTMIMIC: Humanoid Object Interaction via Contact Control

Xinyao Li* **Xialin He*** **Runpei Dong** **Saurabh Gupta**
University of Illinois Urbana-Champaign

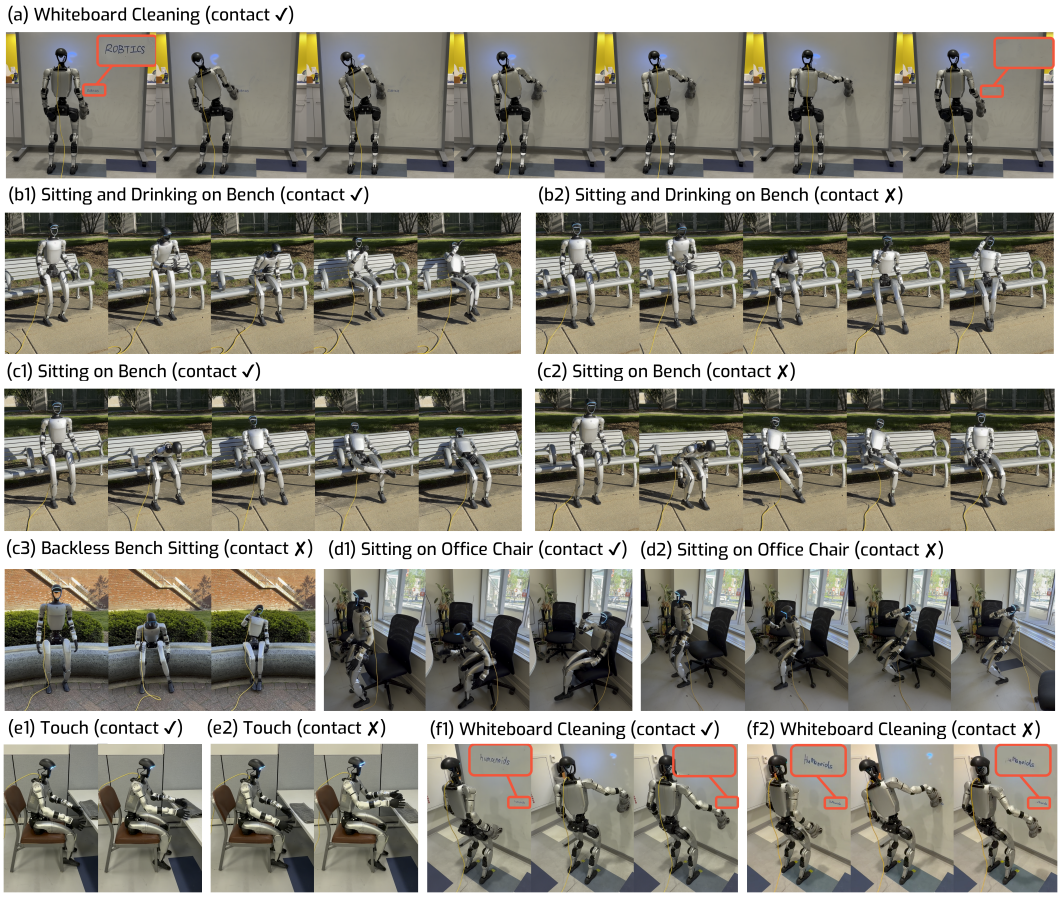

Figure 1: CONTACTMIMIC enables explicit contact control on a real humanoid across diverse interaction tasks. For each task, a same policy is commanded to either make the task-relevant contact (*contact* ✔) or to suppress it (*contact* ✘), by toggling a per-part contact label.

Abstract: Keypoint tracking alone is insufficient for object interaction tasks such as sitting on a chair, wiping a board, or pushing furniture, where the robot can reach the correct pose without making meaningful physical contact with the object. We present CONTACTMIMIC, a learning framework that tracks explicit part-level binary contact commands alongside keypoint trajectories. CONTACTMIMIC is made possible through the use of contact-following rewards and a trajectory augmentation scheme aimed at breaking the correlations between keypoint trajectories and contact labels. The resulting policy successfully decouples contact behavior from keypoint geometry, and achieves precise physical contact as well as contact-controllability (produce or suppress contact during deployment as desired). Simulation experiments across 10 diverse human-object interaction motions confirm that CONTACTMIMIC exhibits contact controllability that en-

*Equal contribution.

ables it to complete manipulation tasks without task-specific rewards, while also outperforming keypoint-only trackers on contact-relevant tasks. Ablations confirm the necessity of the proposed trajectory augmentation scheme and sim2real deployment validates contact controllability in the real world across 5 different motions. Video results are available on https://lixinyao11.github.io/contactmimic-page/.

Keywords: Humanoid Loco-manipulation, Motion Tracking, Contact Modeling

1 Introduction

Most useful whole-body loco-manipulation tasks with a humanoid require making contacts with the environment in important ways. Consider a humanoid wiping a whiteboard, or pushing chairs to tidy a room, or picking up boxes. Success is determined not by the robot's keypoint trajectory, but rather by what body part contacts what objects, and when. For wiping a whiteboard, it is precisely the contacts at the hands that differentiate between waving the hand very close to a whiteboard and actually wiping it. Thus, just keypoint trajectories is an incomplete specification and many useful loco-manipulation tasks can't be expressed just with keypoint trajectories. Yet, current humanoid trackers [1–6] aren't aware of the contacts they should make along the way, and are only trained to track keypoint trajectories. As a result, while they may reproduce the shape of a motion, they miss the contacts that make the motion useful for a task. Thus, even universal keypoint trackers aren't directly useful and require finetuning with task-specific rewards for success [1, 6, 7].

Consider instead *contact-conditioned keypoint trackers*, where rather than just conditioning trackers on keypoint trajectories, we also provide the desired per time-step body-link contact labels as additional input to the policy. The additional contact information disambiguates tasks (*e.g.* wiping *vs.* waving, sitting *vs.* squatting). Furthermore, it can provide more fine-grained control over the motion, *e.g.* sit in the chair but without leaning on the back. And finally, it may be possible to train task-specific policies *without requiring task-specific rewards*, but simply by training for the objective of matching keypoint and contact trajectories.

So, how do we train a *contact-conditioned keypoint tracker*? There are two key considerations: a) how do we source trajectory data with associated contact labels for training, and b) how to design and train contact conditioned policies. For the first question, we leverage human object interaction datasets. Rather than just extracting and retargeting keypoint trajectories, we additionally extract and retarget contact patterns. However, in raw human-object interaction data, keypoint trajectories and contact patterns are often strongly correlated: a motion usually appears with only one typical contact pattern. As a result, the policy may ignore the contact command and simply infer the expected contacts from the keypoints. We mitigate this issue by developing an augmentation scheme that generates motion pairs that have similar keypoints but differ in the contact patterns. The policy can't just go by keypoints but has to attend to the contact commands.

For the second question, we use contact data for policy training in two ways. First, we inject contact commands into the policy. Second, we include rewards that encourage the policy to match the commanded contact patterns: a *contact label matching* reward that measures the agreement between the reference and actual body-part object-part contacts, and a *contact distance* reward that guides relevant body parts toward (or away from) the object surface based on the reference label.

Across both sim and real world experiments with the G1 humanoid robot, we find that contact controllability works: under the same keypoint trajectory, changing the control command changes the robot's behavior. This allows us to execute subtle humanoid object interaction, such as sitting in a chair with or without using the backrest. Compared with a leading keypoint-only tracker, BeyondMimic [1], our method achieves more consistent task-relevant contacts and provides explicit contact control. Furthermore, contact conditioning lets us complete object manipulation tasks (*e.g.* lifting a box), without needing task specific rewards. Ablations suggest that the specific data design that breaks the correlations between keypoints and contacts is important. Finally, we observe that propri-

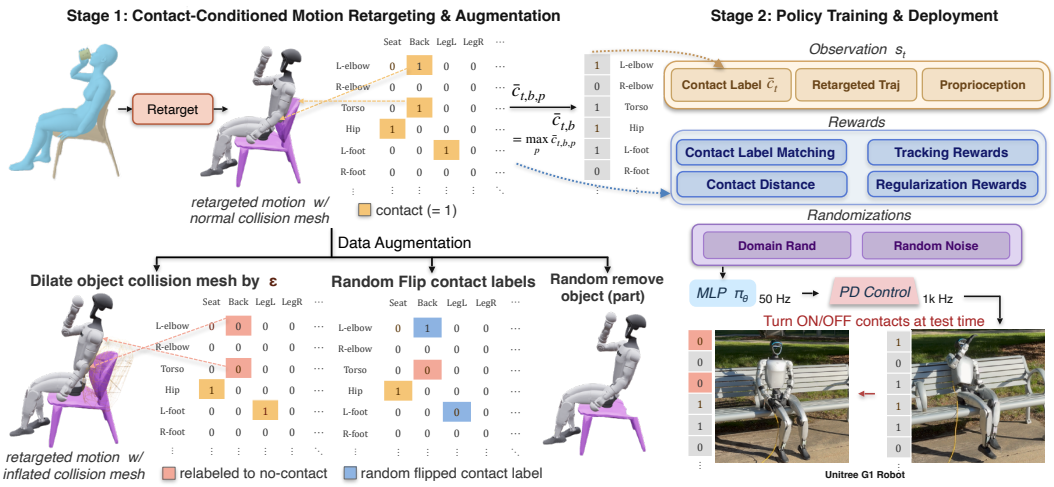

Figure 2: Overview of our pipeline. We retarget human-object interaction clips from HUMOTO, extracting both reference keypoint trajectories and per-body contact labels. We then synthesize augmented motion pairs by **inflated geometry**, **contact-label flipping** and **object removal**. A contact-conditioned policy is then trained on this data with contact-and-keypoint-tracking and contact-aware rewards, so that contact can be **turned on or off at test time** through the conditioning label.

oception itself provides cues for the runtime contact state, justifying our choice of not using contact sensors as policy inputs.

2 Related Work

Humanoid Whole-Body Tracking and Loco-Manipulation. A large body of work trains humanoid policies to imitate human motion through reinforcement learning, by tracking a purely *kinematic* trajectory of keypoints or joint targets [1–15]. Loco-manipulation systems build on such trackers, driving the upper body via teleoperation, vision, task-specific rewards, or force adaptation [16–22]. Across these works, contact with the environment is an incidental byproduct of matching the reference geometry rather than a controllable target, which breaks down on tasks whose meaning is defined by contact (*e.g.* wiping a board *vs.* waving just above it). ResMimic [23] adds a contact-tracking reward to a residual HOI tracker, but conditions only on motion and object trajectories and thus doesn't exhibit fine-grained control (*e.g.* sit without leaning).

Human-Object Interaction Data and Retargeting. Training contact-aware humanoid policies requires reference data that also contains contact patterns with objects. Unlike contact-free motion-capture corpora [24], HOI datasets [25–28] additionally capture object contact. Retargeting these motions onto a humanoid while preserving contact is itself non-trivial: naive joint-angle retargeting yields foot-skating, penetration, or hands floating above intended surfaces. PHC [8] addresses this for free-space motion through physics-based imitation, while OmniRetarget [29] (which we use) and GMR [30] explicitly preserve interaction structure and reduce retargeting artifacts. Existing pipelines, however, output a single canonical retargeted trajectory per clip, which is too narrow to teach a policy to *decouple* contact from keypoint geometry.

Contact-Aware Motion Synthesis. A separate line of work does model contact explicitly, but not as a runtime-controllable input to a closed-loop policy. Classical robotics plans through contact via trajectory optimization with complementarity constraints [31–33], and physics-based character animation conditions motion on scene contact [34–36]. Both require contact schedules to be fixed at planning time. Learning-based HOI synthesis methods, whether kinematic generators [27, 37–39] or physics-based goal/task-conditioned policies [40–42], treat contact as an *intermediate representation* or affordance that keeps motion physically plausible (*e.g.* hands attached to grasped objects), rather than as an explicit knob exposed to the user, which is the goal of our work.

3 Method

Our proposed contact and keypoint tracker, $\pi_\theta(\mathbf{a}_t|\mathbf{p}_t, \bar{\mathbf{k}}_t, \bar{\mathbf{c}}_t)$, takes as input proprioception $\mathbf{p}_t$, reference keypoint position targets $\bar{\mathbf{k}}_t$, and a binary reference contact label map $\bar{\mathbf{c}}_t \in \{0,1\}^{|\mathcal{B}|}$, where $\mathcal{B}$ is the set of contact-capable robot bodies (pelvis, torso, hips, knees, ankles, shoulders, and wrists). The policy outputs target joint angles that are converted into torques using a PD controller.

Overall, we follow standard practices in training the policy π (see Fig. 2 for an overview): we preprocess and retarget human-object reference data to generate reference robot trajectories that are used as tracking targets for training the policy. Our key technical contributions are in the design of the rewards that encourage the policy to make the intended contacts (§ 3.1) and in generating data that allows the policy to learn genuine contact-conditioned behavior (§ 3.2).

3.1 Reward Design

The total reward decomposes into standard regularization terms and keypoint-tracking terms inherited from prior work (described in supp.), and contact-aware rewards introduced in this paper:

$$r_t = \underbrace{r_t^{\text{track}}}_{\text{tracking}} + \underbrace{w_{\text{lm}}\, r_t^{\text{lm}} + w_{\text{cd}}\, r_t^{\text{cd}}}_{\text{contact-aware (ours)}} + \underbrace{r_t^{\text{reg}}}_{\text{regularization}} . \tag{1}$$

The contact-aware terms compare the reference contact label $\bar{c}_{t,b,p} \in \{0,1\}$ to the *actual* contact state $c_{t,b,p} \in \{0,1\}$, between robot body part b, object semantic part p at time t via the following:

Contact label assignment, r_t^{lm}: This term rewards the policy for matching $c_{t,b,p}$ to the reference label $\bar{c}_{t,b,p}$ across all contact pairs. Let $\mathcal{S}_+ = \{(b,p) : \bar{c}_{t,b,p} = 1\}$ and $\mathcal{S}_- = \{(b,p) : \bar{c}_{t,b,p} = 0\}$ denote the set of object and robot parts that are / aren't intended to be in contact. Define the True Positive Rate (TPR), True Negative Rate (TNR), and the False Positive Rate (FPR) as:

$$\text{TPR} = \frac{1}{|\mathcal{S}_+|} \sum_{(b,p)\in\mathcal{S}_+} c_{t,b,p}, \quad \text{TNR} = \frac{1}{|\mathcal{S}_-|} \sum_{(b,p)\in\mathcal{S}_-} (1 - c_{t,b,p}), \quad \text{FPR} = 1 - \text{TNR}. \tag{2}$$

r_t^{lm} takes either of the following two forms:

- **Balanced accuracy** (default): $r_t^{\text{lm}} = \frac{1}{2}(\text{TPR} + \text{TNR})$.
- **TP−FP** (for sparse-contact motions): $r_t^{\text{lm}} = \text{TPR} - \lambda\,\text{FPR}$, which provides a stronger gradient when most pairs have $\bar{c}_{t,b,p} = 0$ and TNR saturates.

Contact distance, r_t^{cd} is computed as $r_t^{\text{cd,+}} + r_t^{\text{cd,-}}$, where $r_t^{\text{cd,+}}$ encourages part-pairs that are supposed to be in contact to be close, and $r_t^{\text{cd,-}}$ encourages part-pairs that are not supposed to be in contact to be far. Specifically, let $d(b,p)$ denote the distance of the origin of robot body b from the surface of the object part p, and $\mathbf{1}[\cdot]$ the indicator function. $r_t^{\text{cd,+}}$ and $r_t^{\text{cd,-}}$ are computed as:

$$r_t^{\text{cd,+}} = \frac{1}{|\mathcal{S}_+|} \sum_{(b,p)\in\mathcal{S}_+} \exp\Big(\frac{-d(b,p)^2}{2\sigma^2}\Big), \quad r_t^{\text{cd,-}} = \frac{-1}{|\mathcal{S}_-|} \sum_{(b,p)\in\mathcal{S}_-} \mathbf{1}[d(b,p) < \delta]\,. \tag{3}$$

3.2 Generating Motion Pairs to Break Correlations between Keypoints and Contacts

Our proposed contact and keypoint tracker needs reference trajectories with per time step keypoint and contact labels for training. We extract such contact label-paired keypoint trajectories from human MoCap data. Existing works only extract keypoints, we extend them to also extract contact information. As our experiments show, just training on this data alone doesn't lead to good contact control due to the correlations between keypoint patterns and contacts. Thus, to achieve genuine contact control, we propose augmentation strategies that break this correlation.

Retargeting Human Motions to G1 and Extracting Contact Information. We use OmniRetarget [29] to retarget a given human-object interaction clip to the G1 humanoid, producing a reference robot configuration trajectory $\bar{\mathbf{q}}_{1:T}$ (joint positions and root pose). We extract *reference contact labels* $\bar{c}_{t,b,p}$ directly from the retargeted trajectory: at each frame t, we mark a robot body part b as in

contact ($\bar{c}_{t,b,p} = 1$) when its distance to the object surface is below a 1 cm threshold, and assign the semantic object part p (e.g., chair seat and board surface) as the nearest object part. $\bar{c}_{t,b,p}$ is directly used to compute rewards, and also gives the conditioning vector $\bar{c}_{t,b}$ via $\max_p \bar{c}_{t,b,p}$.

Augmented trajectory. For each retargeted clip, we synthesize several augmentations that preserve the overall motion structure but break the correlations between the keypoint positions and the contact labels for the task relevant contacts (*e.g.* wrist-box contact for the task of lifting a box).

Specifically, we generate three different augmentations: ❶ **Contact-label flipping**: we retain the original trajectory but flip the task-relevant contact labels, the object remains in simulation but the robot is penalized for making contact with it; ❷ **Object removal**: we remove the interaction object and force the target contact labels to zero, while fully retaining the original keypoint trajectory; and ❸ **Inflated geometry**: we inflate the collision geometry of target object parts during retargeting to route the robot around them. This produces a perturbed keypoint trajectory where target contacts no longer occur, and those contact labels are zeroed out accordingly. These augmentations are composed together. For instance, combining *Inflated Geometry* with *Contact-label flipping* pairs a distant trajectory with a *true* contact label, simulating the situation where we want contact to happen but reference keypoints are slightly misspecified.

4 Experiments

Our experiments across diverse human-object interaction motions seek to answer 5 key questions: Do our policies follow the commanded contact conditioning? Does contact control transfer to the real world? Is keypoint control alone sufficient, or does a state-of-the-art keypoint-tracking baseline already produce accurate physical contact? Is our data engineering strategy necessary for learning contact control? To what extent does the policy's internal representation already encode the robot's actual runtime contact state, given only proprioceptive observations?

4.1 Experimental Setup

Robot and Simulator. We use the Unitree G1 humanoid robot (29 DoF) in Isaac Lab [43] with PhysX rigid-body simulation. Each environment contains one interaction object, which is either fixed or free depending on the motion.

Motion Dataset. We train on 10 motion clips from the HUMOTO [28] dataset, spanning a range of human-object interaction categories, as summarized in Table 1.

Table 1: Motions used in our experiments. "Contact pair" specifies the task-relevant robot-body and object part involved in the contact. All objects, except for the last two motions, are fixed in the environment.

Motion	Contact pair	Object (type)
Wipe whiteboard	hand & whiteboard surface	whiteboard
Sit in front of table	hands & table top	chair + table
Lean on backrest I	torso & chair backrest	chair
Lean on backrest II	torso & chair backrest	chair
Step foot on chair	foot & chair seat	chair
Sit on table	pelvis / hands & table top	table
Lean against table	hands & table top	table
Sit and squat	pelvis & chair seat (sit only)	chair
Kick chair	foot & chair leg/base	chair (free)
Pick up box	hands & box surface	box (free)

Metrics. Across all motions we report **contact bodies**, the number of robot bodies in contact with the *target object part* (e.g., chair seat) averaged over the episode (with the reference count *Ref* for comparison); **contact impulse** (N·s) accumulated over the episode; mean **key-joint torque** (N·m) at a motion-specific joint; and **MPJPE** (cm) against the reference. For free-object motions we also report the **object displacement** (m), how much the object travels from its initial pose; larger values indicate that the policy successfully manipulates the object.

4.2 Contact Control Works in Simulation

We measure the extent to which the policy follows the commanded contact. For the same keypoint trajectory τ, we command the policy to either make task-relevant contact (denoted by $\tau^{✔}$) or to not make task-relevant contact (denoted by $\tau^{✘}$). For a sitting motion, $\tau^{✔}$ would be resting the torso against the chair back, while $\tau^{✘}$ would be leaning back but without making contact with the

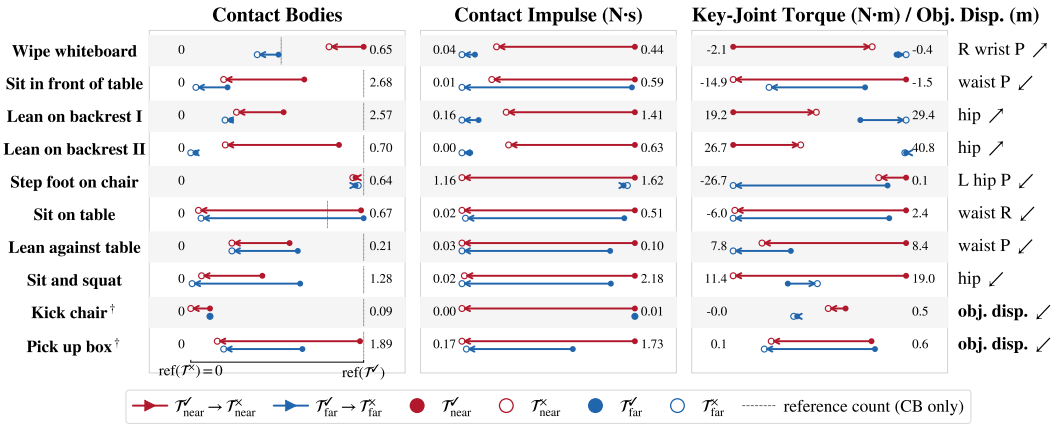

Figure 4: Per-motion visualization of contact controllability. Red arrows (near keypoints, contact ✔→✘, i.e. $\mathcal{T}^{✔}_{\text{near}} \rightarrow \mathcal{T}^{✘}_{\text{near}}$) and blue arrows (far keypoints, contact ✔→✘, i.e. $\mathcal{T}^{✔}_{\text{far}} \rightarrow \mathcal{T}^{✘}_{\text{far}}$) show contact metrics dropping with the contact command turned off. Both arrows confirm that the policy corresponds to the contact command. In the rightmost panel, the arrow beside each key joint marks whether its torque should increase (↗) or decrease (↙) when the contact command is turned off.

chair back. We assess whether the policy makes task-relevant contacts when asked to track $\tau^{✔}$ and whether the task-relevant contacts drop when asked to track $\tau^{✘}$ instead. We conduct this test on two sets of keypoint trajectories: a) $T_{\text{kp=near}}$, the original object-interaction trajectories in HUMOTO, and b) $T_{\text{kp=far}}$, augmented trajectory where the keypoints have been moved far from the object surface using the inflation process from § 3.2. Altogether we test on 4 trajectory sets: $\mathcal{T}^{✔}_{\text{near}}$, $\mathcal{T}^{✘}_{\text{far}}$, $\mathcal{T}^{✘}_{\text{near}}$, $\mathcal{T}^{✔}_{\text{far}}$.

Results. We plot the number of contacts and contact impulses in Fig. 4. As expected, we observe that the number of contacts (and contact impulse) is higher for trajectories $\mathcal{T}^{✔}_{\text{near}}$and they decrease as we turn off the contacts in trajectories $\mathcal{T}^{✘}_{\text{near}}$(the red arrow points left). The same trend holds when we go from $\mathcal{T}^{✔}_{\text{far}}$to $\mathcal{T}^{✘}_{\text{far}}$(the blue arrow points left). The right pane in Fig. 4 shows that the torque at the contact-relevant changes as expected, *e.g.* the hip torque increases as the robot supports its back when it is not leaning on the backrest.

For motions that lead to object motion when contact is being made ('Kick chair with foot' and 'Pick up box'), we see that the object displacement is indeed larger when the policy is asked to make contacts. Fig. 3 visualizes the box lifting motion in simulation. Just with keypoint+contact tracking and without any task-specific rewards, CONTACTMIMIC is able to loco-manipulate objects.

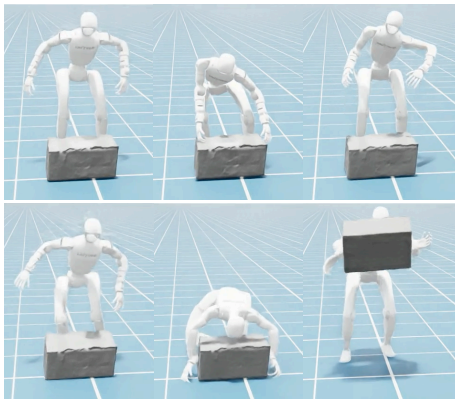

Figure 3: The box-lifting motion in simulation. Top row: contact ✘, the same keypoints are tracked but the box is left untouched; bottom row: contact ✔, the robot grasps and lifts the box.

4.3 Contact Control Transfers to the Real World

We replicate the contact controllability study on hardware. We evaluate five motions on the real robot: *wipe whiteboard*, *sit in front of table*, *lean on backrest* (I & II), and *sit and squat*. For each motion, we deploy the policy with the same keypoints but toggle the contact command from ✔to ✘. A trial is successful if the robot qualitatively follows the intended trajectory and its contact behavior matches the commanded contact, *i.e.* contact between target pair occurs if and only if the contact label is *true* (see Table 2 for per-motion criteria).

Results. Table 2 reports real-world success rates under both contact commands: ✔and ✘. Similar to simulation results, we observe that the policy follows the commanded contact.

Table 2: Real-world success rates under different controllability conditions. A trial is successful if the robot's physical contact behavior matches contact command (see criteria column).

Motion	**Success criterion**	contact ✔	contact ✘
Wipe whiteboard	Hand contact leaves visible trace ⇔ contact=✔	5/5	5/5
Sit in front of table	Full seat contact + hands on table ⇔ contact=✔	4/5	5/5
Lean on backrest I	Sustained torso-backrest contact ⇔ contact=✔	9/10	10/10
Lean on backrest II	Sustained torso-backrest contact ⇔ contact=✔	10/10	9/10
Sit and squat	Seated posture ⇔ contact=✔, else squat posture	5/5	5/5

Table 3: Despite similar MPJPE, the keypoint-only method, BeyondMimic [1], fails to establish physical contact. † For object-manipulation motions we also report the object displacement (higher means the object is moved more).

Motion	Contact bodies↑		Impulse (N·s)↑		Obj. disp. (m)↑		MPJPE (cm)↓	
	BM	Ours	BM	Ours	BM	Ours	BM	Ours
Wipe whiteboard	$0.01_{\pm 0.09}$	$\mathbf{0.65}_{\pm 0.45}$	$0.00_{\pm 0.03}$	$\mathbf{0.44}_{\pm 0.37}$	—	—	$3.9_{\pm 1.0}$	$3.6_{\pm 0.8}$
Sit in front of table	$1.51_{\pm 0.51}$	$\mathbf{1.76}_{\pm 0.33}$	$\mathbf{4.14}_{\pm 8.20}$	$0.59_{\pm 0.22}$	—	—	$5.8_{\pm 1.9}$	$5.2_{\pm 2.5}$
Lean on backrest I	$0.12_{\pm 0.24}$	$\mathbf{1.38}_{\pm 1.27}$	$0.04_{\pm 0.15}$	$\mathbf{1.41}_{\pm 1.46}$	—	—	$7.1_{\pm 2.2}$	$4.8_{\pm 1.3}$
Lean on backrest II	$0.39_{\pm 0.35}$	$\mathbf{0.60}_{\pm 0.39}$	$0.59_{\pm 0.58}$	$\mathbf{0.63}_{\pm 0.55}$	—	—	$4.9_{\pm 1.9}$	$5.9_{\pm 3.0}$
Step foot on chair	$\mathbf{0.62}_{\pm 0.48}$	$0.61_{\pm 0.48}$	$1.29_{\pm 1.06}$	$\mathbf{1.62}_{\pm 1.33}$	—	—	$3.9_{\pm 1.5}$	$4.3_{\pm 1.3}$
Sit on table	$0.22_{\pm 0.34}$	$\mathbf{0.66}_{\pm 0.47}$	$0.44_{\pm 0.85}$	$\mathbf{0.51}_{\pm 0.60}$	—	—	$9.5_{\pm 5.0}$	$9.6_{\pm 3.2}$
Lean against table	$0.07_{\pm 0.25}$	$\mathbf{0.12}_{\pm 0.30}$	$0.03_{\pm 0.15}$	$\mathbf{0.10}_{\pm 0.51}$	—	—	$5.3_{\pm 4.3}$	$5.9_{\pm 4.0}$
Sit and squat	$0.12_{\pm 0.24}$	$\mathbf{0.53}_{\pm 0.51}$	$0.04_{\pm 0.15}$	$\mathbf{2.18}_{\pm 2.22}$	—	—	$7.1_{\pm 2.2}$	$5.6_{\pm 1.6}$
Kick chair†	$\mathbf{0.02}_{\pm 0.09}$	$0.01_{\pm 0.08}$	$\mathbf{0.02}_{\pm 0.23}$	$0.01_{\pm 0.12}$	$0.15_{\pm 0.13}$	$\mathbf{0.31}_{\pm 0.15}$	$4.0_{\pm 1.4}$	$3.8_{\pm 1.2}$
Pick up box†	$0.29_{\pm 0.59}$	$\mathbf{1.85}_{\pm 0.94}$	$0.09_{\pm 0.22}$	$\mathbf{1.73}_{\pm 1.18}$	$0.03_{\pm 0.01}$	$\mathbf{0.49}_{\pm 0.47}$	$3.7_{\pm 5.2}$	$7.5_{\pm 3.5}$

Qualitative observations. When commanded to make contact, the policy commits to the physical interaction, whereas when commanded to suppress it the policy follows a similar pose while staying just off the surface. For *wipe whiteboard*, the hand applies sustained pressure, and the eraser erases, *vs.* hovering just off the board with no effect. For *sit in front of table*, the robot transfers weight onto the seat and rests its hands on the table top, *vs.* lowering to the seated pose without bearing weight. For *lean on backrest*, the torso establishes sustained backrest contact (with distinct postural styles across the two variants), *vs.* remaining upright just shy of the backrest. For *sit*, the robot transfers weight to the seat, *vs.* for *squat* it performs a dynamically balanced squat at the matched height.

4.4 Keypoint Control Alone Is Insufficient

Next, we assess if just tracking keypoints (the current state-of-the-art) is sufficient when the underlying task requires making contact. Here as a comparison point, we use BeyondMimic [1], a state-of-the-art whole-body motion tracking framework that doesn't take any contact conditioning as input and is also not trained with any contact rewards. Because BeyondMimic doesn't take any contact conditioning as input, we only train it on sequences that have contact and only compare to it in situations where the task is to make contact.

Results. Table 3 shows that BeyondMimic [1] has much lower contact metrics (both contact counts and contact impulses) than our method. This suggests that keypoint tracking alone is insufficient for tasks that require making contacts with the environment. At the same time, both methods have comparable MPJPE, indicating that our method is not trading off keypoint tracking accuracy while following contact conditioning. Lastly, for object manipulation tasks, keypoint-only control fails to manipulate the object, while our proposed keypoint+contact control succeeds in manipulating the object, even when the keypoint tracking errors are similar.

4.5 Our Proposed Data Augmentation Scheme Is Important

We measure the importance of our data augmentation techniques for successful contact control. Fig. 5 compares our method with and without the data augmentation strategies from § 3.2 when tested on trajectories $\mathcal{T}^{\checkmark}_{\text{near}}$and $\mathcal{T}^{\times}_{\text{near}}$. As before, left-pointing arrows will suggest good contact control. Our method with data augmentation has much better contact control than the version without the data for all but one motion. This suggests that just the policy input and reward functions aren't

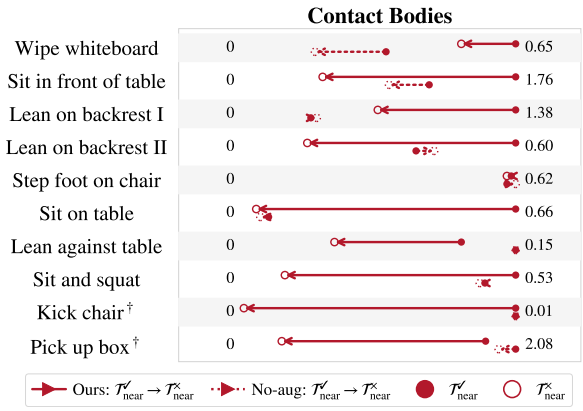

Figure 5: Removing proposed data augmentation hurts contact controllability. Solid arrows for our full method, are more consistently long and facing left (*i.e.* contacts reduce when commanded not to make contact) than the dotted arrows for the version trained without the augmentations from § 3.2.

Motion	Cha.	Ref label	Obs	Layer 2
Wipe whiteboard	61	0.843	0.956	**0.964**
Sit in front of table	99	0.762	**0.997**	**0.997**
Lean on backrest I	70	0.906	0.943	**0.964**
Lean on backrest II	75	0.894	0.944	**0.968**
Step foot on chair	67	0.963	0.987	**0.994**
Sit on table	69	0.922	**0.958**	**0.958**
Lean against table	13	0.885	**0.926**	0.915
Sit and squat	62	0.906	0.970	**0.974**
Kick chair†	2	0.000	0.267	**0.476**
Pick up box†	81	0.846	0.930	**0.931**

Table 4: Runtime contact state can be inferred using proprioception. Linear probing F1 for predicting the runtime contact state (task-relevant contact) from the policy input (*Obs*) and layer-2 representations (*Layer 2*). Both are well above the F1 from the reference contact label (*Ref label*); *Cha.*(*Chance* (%)) is the positive fraction of frames.

sufficient to successfully learn policies with contact-control, training data needs to be properly engineered to break the correlation between keypoint positions and contact labels.

4.6 Policy Representations Encode Runtime Contact State

As most robots do not have whole-body contact sensors, we made the deliberate decision to not require the actual runtime contact state for the different robot body parts as a policy input. Partly our rationale was that the contact state could be inferred using proprioception. We assess this quantitatively by computing the contact state prediction accuracy of a linear probe on the policy inputs and the policy's intermediate representations.

Results. Table 4 reports the test-set F1 scores across all motions. For both the raw observation and Layer 2 representations, the F1 scores are very high, and much higher than the chance performance (*i.e.* the number of time steps with contact) as well as the reference label itself. Thus, despite receiving no explicit contact sensing at inference time, the policy has a sense of the robot's runtime contact state and can adapt its behavior to fulfill the commanded contact state.

5 Discussion and Limitations

We presented CONTACTMIMIC, a framework that augments humanoid keypoint tracking with an explicit, test-time controllable contact label. By training a single policy on paired motions that share keypoint structure but differ in contact commands, and supervising it with contact-aware rewards, the policy learns to make or suppress physical contact on command. Across diverse human-object interaction motions in simulation and on the real Unitree G1, the policy follows the commanded contact, outperforms a keypoint-only baseline on contact metrics at comparable tracking accuracy, and even encodes its runtime contact state from proprioception alone. We believe explicit contact conditioning is a simple and general interface for expressing contact-rich tasks, and a step toward humanoids that interact with the world purposefully rather than incidentally.

Our approach has several limitations. First, we train a separate policy per motion rather than a single universal contact-conditioned tracker. Jointly training a contact-conditioned policy across motions would be a natural next step. Second, we built our approach on high-quality human-object interaction data from the HUMOTO dataset which limits the diversity of interactions we can currently cover. Developing methods that can use in-the-wild video data instead can help. Finally, our real-world evaluation spans five motions on a single robot; broader hardware validation is future work.

Acknowledgments

We are grateful to the Coordinated Science Laboratory and the Center for Autonomy for access to experimental space and the G1 humanoid robot used in this work. We thank John Hart for his invaluable help with the humanoid robots. This material is based upon work supported by an NSF CAREER Award (IIS-2143873).

References

[1] 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] 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.

[3] 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. In *Conference on Robot Learning (CoRL)*, 2024.

[4] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. In *Conference on Robot Learning (CoRL)*, 2024.

[5] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbabu, C. Pan, Z. Yi, G. Qu, K. Kitani, J. Hodgins, L. Fan, Y. Zhu, C. Liu, and G. Shi. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. In *Robotics: Science and Systems (RSS)*, 2025.

[6] 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 (SIGGRAPH Asia)*, 2024.

[7] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang, L. Fan, and Y. Zhu. Hover: Versatile neural whole-body controller for humanoid robots. *arXiv preprint arXiv:2410.21229*, 2024.

[8] Z. Luo, J. Cao, A. Winkler, K. Kitani, and W. Xu. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[9] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang. Expressive whole-body control for humanoid robots. In *Robotics: Science and Systems (RSS)*, 2024.

[10] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[11] 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.

[12] W. Xie, J. Han, J. Zheng, H. Li, X. Liu, J. Shi, W. Zhang, C. Bai, and X. Li. Kungfubot: Physics-based humanoid whole-body control for learning highly-dynamic skills. *arXiv preprint arXiv:2506.12851*, 2025.

[13] 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.

[14] J. Li, X. Cheng, T. Huang, S. Yang, R.-Z. Qiu, and X. Wang. Amo: Adaptive motion optimization for hyper-dexterous humanoid whole-body control. *arXiv preprint arXiv:2505.03738*, 2025.

[15] A. Allshire, H. Choi, J. Zhang, D. McAllister, A. Zhang, C. M. Kim, T. Darrell, P. Abbeel, J. Malik, and A. Kanazawa. Visual imitation enables contextual humanoid control. *arXiv preprint arXiv:2505.03729*, 2025.

[16] 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.

[17] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu. Generalizable humanoid manipulation with 3d diffusion policies. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2873–2880, 2025. doi: 10.1109/IROS60139.2025.11246340.

[18] X. He, S. Xu, X. Li, R. Dong, L. Bian, Y.-X. Wang, and L.-Y. Gui. Ultra: Unified multimodal control for autonomous humanoid whole-body loco-manipulation. *arXiv preprint arXiv:2603.03279*, 2026.

[19] R. Dong, Z. Li, X. He, and S. Gupta. Learning humanoid end-effector control for open-vocabulary visual loco-manipulation. *arXiv preprint arXiv:2602.16705*, 2026.

[20] Y. Zhang, Y. Yuan, P. Gurunath, I. Gupta, S. Omidshafiei, A.-a. Agha-mohammadi, M. Vazquez-Chanlatte, L. Pedersen, T. He, and G. Shi. Falcon: Learning force-adaptive humanoid loco-manipulation. *arXiv preprint arXiv:2505.06776*, 2025.

[21] 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.

[22] F. Liu, Z. Gu, Y. Cai, Z. Zhou, H. Jung, J. Jang, S. Zhao, S. Ha, Y. Chen, D. Xu, and Y. Zhao. Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation. *arXiv preprint arXiv:2409.20514*, 2024.

[23] 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.

[24] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[25] 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 (ECCV)*, 2020.

[26] B. L. Bhatnagar, X. Xie, I. Petrov, C. Sminchisescu, C. Theobalt, and G. Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[27] J. Li, J. Wu, and C. K. Liu. Object motion guided human motion synthesis. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2023.

[28] J. Lu, C.-H. P. Huang, U. Bhattacharya, Q. Huang, and Y. Zhou. Humoto: A 4d dataset of mocap human object interactions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[29] 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.

[30] J. P. Araujo, Y. Ze, P. Xu, J. Wu, and C. K. Liu. Retargeting matters: General motion retargeting for humanoid motion tracking. *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

[31] I. Mordatch, E. Todorov, and Z. Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (SIGGRAPH)*, 31(4), 2012.

[32] M. Posa, C. Cantu, and R. Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research (IJRR)*, 33(1):69–81, 2014.

[33] Y. Tassa, T. Erez, and E. Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4906–4913, 2012.

[34] 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 (SIGGRAPH)*, 2018.

[35] S. Starke, H. Zhang, T. Komura, and J. Saito. Neural state machine for character-scene interactions. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2019.

[36] M. Hassan, D. Ceylan, R. Villegas, J. Saito, J. Yang, Y. Zhou, and M. J. Black. Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[37] 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.

[38] 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.

[39] J. Li, A. Clegg, R. Mottaghi, J. Wu, X. Puig, and C. K. Liu. Controllable human-object interaction synthesis. In *European Conference on Computer Vision (ECCV)*, 2024.

[40] X. B. Peng, A. Kanazawa, J. Malik, P. Abbeel, and S. Levine. Sfv: Reinforcement learning of physical skills from videos. *ACM Transactions on Graphics (TOG)*, 37(6), 2018.

[41] S. Xu, S. Schulter, M. Ziyadi, X. He, X. Fei, Y.-X. Wang, and L.-Y. Gui. Interprior: Scaling generative control for physics-based human-object interactions. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[42] L. Pan, Z. Yang, Z. Dou, W. Wang, B. Huang, B. Dai, T. Komura, and J. Wang. Tokenhsi: Unified synthesis of physical human-scene interactions through task tokenization. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[43] M. Mittal, P. Roth, J. Tigue, A. Richard, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

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

APPENDIX

In this supplementary, we provide additional details and results that supplement CONTACTMIMIC:

1. § A describes the organization of the supplementary video and the anonymous project page.
2. § B provides additional implementation details, including PPO hyperparameters, policy observations and architecture, and domain randomization.
3. § C lists the full set of reward terms with their weights and shaping parameters.
4. § D details our paired-motion generation pipeline: contact-label extraction, augmentation parameters, and the per-motion training configuration.
5. § E describes our real-world deployment on the Unitree G1.
6. § F reports additional quantitative results, including the full per-motion contact controllability table and the per-motion data augmentation ablation.

A Supplementary Videos and Project Page

We provide an overview video and the raw recordings of every real-world trial reported in Table 2 on our project page. The videos and page include:

- **Method overview**: the full pipeline, from human-motion retargeting and paired-motion augmentation that breaks keypoint–contact correlations, to contact-conditioned policy training with our contact-following rewards and real-world deployment.
- **Real-world results**: for each of the five tested motions (*wipe whiteboard*, *sit in front of table*, *lean on backrest* I/II, *sit and squat*), side-by-side clips of the same policy on the real Unitree G1 commanded with contact ✔and contact ✘, demonstrating on-command toggling of physical contact.
- **Simulation results**: per-motion clips across all 10 motions under the four eval-time trajectories $\mathcal{T}^{\checkmark}_{\text{near}}$–$\mathcal{T}^{\checkmark}_{\text{far}}$, including the flipped conditions $\mathcal{T}^{\times}_{\text{near}}$and $\mathcal{T}^{\checkmark}_{\text{far}}$that test pure contact-label following.
- **Baseline comparison**: side-by-side comparisons against the state-of-the-art keypoint-only tracker BeyondMimic [1] under the same keypoint trajectory, showing that without contact conditioning it can only follow the single mode baked into the trajectory and cannot be steered at test time.
- **Ablation**: comparisons against a no-augmentation variant, showing that without our paired-motion augmentation the policy ignores the contact command and fails contact-controlled tasks (e.g., *sit on table*, *lean on backrest II*).

B Implementation Details

We train one policy per motion with PPO [44] in Isaac Lab [43] across 4096 parallel environments at 50 Hz. At each environment reset, the episode is randomly assigned either the default trajectory or one of its augmented counterparts (§ 3.2), so the same policy sees both contact and no-contact target labels during training. PPO hyperparameters are listed in Table 5.

Policy Observations. The actor observation $\mathbf{o}_t$ concatenates: (i) *proprioception* $\mathbf{p}_t$: joint positions, joint velocities, base angular velocity, and projected gravity; (ii) *reference keypoint targets* $\bar{\mathbf{k}}_t$, expressed in the robot's local frame and derived from the reference configuration $\bar{\mathbf{q}}_t$ via forward kinematics; (iii) the *binary reference contact label* $\bar{\mathbf{c}}_t \in \{0,1\}^{|\mathcal{B}|}$ (with $\bar{c}_{t,b} = \max_p \bar{c}_{t,b,p}$) for the current frame. Uniform observation noise is added during training to all actor observations.

Policy Architecture. Both actor and critic are MLPs with hidden dimensions $[512, 256, 128]$ and ELU activations. The critic receives *noise-free* versions of the actor's observations, plus the base linear velocity, which is not available to the actor; this acts as the critic's privileged information.

Table 5: PPO hyperparameters used to train each contact-conditioned policy.

Parameter	Value
Parallel environments	4096
Policy step rate	50 Hz
Steps per env per update	24
Max iterations	30,000
Learning rate	1×10^{-3} (adaptive, KL target 0.01)
Discount factor γ	0.99
GAE λ	0.95
PPO clip parameter	0.2
Mini-batches per epoch	4
Learning epochs per update	5
Entropy coefficient	5×10^{-3}
Value-loss coefficient	1.0
Max gradient norm	1.0
Initial action noise std	1.0

Domain Randomization. We randomize, at each episode start, robot link masses, joint friction, and object friction and mass within standard ranges, to facilitate sim-to-real transfer.

C Reward Details

Following § 3.1, the per-step reward decomposes into tracking, contact-aware, and regularization terms. Table 6 lists every reward term, its functional form, weight, and any tunable shaping parameter. Tracking terms follow BeyondMimic [1] and use Gaussian kernels $\exp(-e^2/\sigma^2)$ on the corresponding error signal e.

Table 6: Full list of reward terms used to train each contact-conditioned policy. Positive weights are rewards; negative weights are penalties. Gaussian terms are shaped as $\exp(-e^2/\sigma^2)$ on the underlying error e.

Group	Term	Error signal / form	Weight
Tracking	Global anchor pos.	Root position error; $\sigma=0.3$ m	0.5
	Global anchor ori.	Root orientation error; $\sigma=0.4$ rad	0.5
	Body position	Per-body position error; $\sigma=0.3$ m	1.0
	Body orientation	Per-body orientation error; $\sigma=0.4$ rad	1.0
	Body lin. vel.	Per-body linear velocity error; $\sigma=1.0$ m/s	1.0
	Body ang. vel.	Per-body angular velocity error; $\sigma=\pi$ rad/s	1.0
Contact	Label matching	Balanced accuracy $\frac{1}{2}(\mathrm{TPR}+\mathrm{TNR})$, force threshold 1 N	4.0
	Distance	Gaussian on body-to-part dist., $\sigma = 0.2$ m; unwanted-contact penalty at $\delta=0.05$ m	3.0
Reg.	Action rate	$\|\mathbf{a}_t - \mathbf{a}_{t-1}\|_2^2$	-0.15
	Joint limits	Quadratic, on joint-limit violation	-10.0
	Undesired contacts	Non-foot/wrist body in contact with terrain	-0.1

The contact label-matching reward defaults to the balanced-accuracy form $r_t^{\mathrm{lm}} = \frac{1}{2}(\mathrm{TPR}+\mathrm{TNR})$; for sparse-contact motions we instead use the TP$-$FP form $r_t^{\mathrm{lm}} = \mathrm{TPR} - \lambda\,\mathrm{FPR}$ with $\lambda = 1.0$, which provides a stronger gradient when most $\bar{c}_{t,b,p} = 0$ and TNR saturates.

D Paired Motion Generation Details

Contact label extraction. Given a retargeted trajectory from OmniRetarget [29], we replay the trajectory in MuJoCo and at each frame t mark a robot body part b as in contact ($\bar{c}_{t,b,p} = 1$) when its distance to the object surface is below 1 cm; the semantic part p is assigned by looking up the nearest object surface point.

Augmentation parameters. We synthesize three augmentations (§ 3.2). For inflated-geometry augmentation, the collision geometry of the target object part(s) is expanded outward by an isotropic offset of $\delta_{\text{infl}} \in [5, 10]$ cm in all directions during a second retargeting pass; this forces the retargeter to route the relevant robot bodies further from the original surface, and the contact labels of the target pairs are zeroed out accordingly. At training time we further apply random contact-label flipping and random interaction-object removal; the per-motion probabilities vary slightly and are listed in Table 7.

Per-motion settings. Table 7 summarizes the per-motion training configuration: which augmentations are used, the contact label-matching reward mode (balanced accuracy or TP−FP), the augmentation probabilities, and the number of PPO iterations. Sparse-contact motions (e.g., *wipe whiteboard*, free-object kicks and lifts) use the TP−FP mode to provide a stronger gradient when most reference labels are zero.

Table 7: Per-motion training settings. *Aug.* columns indicate which paired-motion augmentations are enabled: *Infl.* (inflated geometry), *Rem.* (interaction-object removal at train time), and *Flip* (random contact-label flipping). *Mode* is the label-matching reward form (Bal. = balanced accuracy, TP−FP = $\mathrm{TPR} - \lambda\mathrm{FPR}$, λ=1). $p_{\text{rem.}}$ and p_{flip} are the train-time augmentation probabilities (per episode).

Motion	Infl. aug.	Rem. aug.	Flip aug.	Mode	$p_{\text{rem.}}/p_{\text{flip}}$
Wipe whiteboard	✔	✔	✔	TP−FP	0.3 / 0.2
Sit in front of table	✔	✔	✔	Bal.	0.3 / 0.2
Lean on backrest I	✔	✔	✔	Bal.	0.3 / 0.2
Lean on backrest II	✔	✔	✔	Bal.	0.3 / 0.2
Step foot on chair	✔	✔	✔	Bal.	0.3 / 0.2
Sit on table	✔	✘	✔	Bal.	0.0 / 0.2
Lean against table	✔	✔	✔	Bal.	0.3 / 0.2
Sit and squat	✘	✔	✔	Bal.	0.3 / 0.2
Kick chair†	✔	✔	✔	TP−FP	0.3 / 0.2
Pick up box†	✔	✔	✔	TP−FP	0.3 / 0.2

Sit and squat is the only motion that does *not* use inflated geometry: because the augmented "no-contact" trajectory for this motion is simply a squat at the same height, we obtain it directly by removing the chair from the scene rather than by re-retargeting against an inflated chair. *Sit on table* skips the object-removal augmentation since the robot needs to sit on the table surface, and removing the table would directly lead to failure.

E Real-World Deployment

We deploy each policy on a Unitree G1 (29 actuated DoFs). The policy runs onboard at 50 Hz, producing target joint positions that are tracked by the robot's joint-level PD controller at 1000 Hz, identical to the simulation control stack. Object placement (chair, table, board, bench) is measured by hand before each trial and reproduced on a fixed mat to align the robot's start pose with the reference. No external motion capture or vision is used at deployment time; the policy receives only its onboard proprioception, the pre-recorded reference, and the contact-label command.

F Additional Quantitative Results

Per-motion contact controllability. Table 8 reports the full per-motion contact controllability results across all four trajectories ($\mathcal{T}^{✔}_{\text{near}}$–$\mathcal{T}^{✔}_{\text{far}}$). Contact bodies and impulse track the contact label rather than the keypoint variant, confirming the policy follows the commanded contact.

Per-motion takeaways from Table 8. While the overall trend—contact bodies and impulse track the contact label—holds across motions, several per-motion patterns are worth highlighting:

- **Sit on table** shows the cleanest separation: when commanded contact ✘, both impulse and bodies collapse to near zero ($\mathcal{T}^{✘}_{\text{far}}$, $\mathcal{T}^{✘}_{\text{near}}$), while contact ✔recovers them to the reference level ($\mathcal{T}^{✔}_{\text{near}}$, $\mathcal{T}^{✔}_{\text{far}}$). This is the prototypical case where the contact label fully drives the policy's behavior.

- **Step foot on chair** is an interesting partial-control case: contact bodies are essentially identical across all four trajectories (~ 0.6), because the robot cannot balance on a single leg with the other foot raised that high, so it *must* keep contact with the chair. However, the contact impulse still tracks the label ($\mathcal{T}^{\text{✘}}_{\text{near}}$ $1.16\,\text{N}\cdot\text{s}$ vs. $\mathcal{T}^{\text{✔}}_{\text{near}}$ $1.62\,\text{N}\cdot\text{s}$), showing that the policy modulates how *firmly* the foot pushes down even when contact is unavailable.
- **Lean on backrest I** and **Wipe whiteboard** show some residual contact under contact ✘with near keypoints ($\mathcal{T}^{\text{✘}}_{\text{near}}$bodies 0.68 and 0.52): the reference keypoints place the torso/hand sufficiently close to the surface that fully suppressing contact would incur a keypoint tracking error whose cost exceeds the contact-aware penalty under our reward weighting, leading the policy to settle at a trade-off where residual contact persists. The impulse, however, still drops by a factor of 3–$4\times$.
- For the free-object motions, the most informative signal is the object displacement. **Kick chair** only produces a brief foot–chair contact (two kicks), so contact bodies and impulse are tiny (~ 0.01) and barely vary across conditions. Yet the displacement shows that with contact ✔and near keypoints ($\mathcal{T}^{\text{✔}}_{\text{near}}$) the two kicks land cleanly ($0.31\,\text{m}$), while contact ✘($\mathcal{T}^{\text{✘}}_{\text{near}}$, $0.26\,\text{m}$) or far keypoints with contact ✔($\mathcal{T}^{\text{✔}}_{\text{far}}$, $0.17\,\text{m}$) lead to weaker or partial contact.
- **Pick up box** has the largest displacement under contact ✔($\mathcal{T}^{\text{✔}}_{\text{near}}0.49$, $\mathcal{T}^{\text{✔}}_{\text{far}}0.50$): the policy commits to grasping and lifting the box. Under contact ✘the box is barely displaced ($\mathcal{T}^{\text{✘}}_{\text{far}}0.18$, $\mathcal{T}^{\text{✘}}_{\text{near}}0.20$), even though the keypoint trajectory for $\mathcal{T}^{\text{✘}}_{\text{near}}$is the original pickup motion—this is the clearest example of contact conditioning enabling task-relevant manipulation that keypoints alone cannot produce.

Data augmentation ablation (per-motion). Fig. 6 visualizes the per-motion arrow plot for both contact bodies and impulse, and Table 9 reports the underlying numbers for our method with and without the data augmentation strategies from § 3.2, evaluated on $\mathcal{T}^{\text{✔}}_{\text{near}}$and $\mathcal{T}^{\text{✘}}_{\text{near}}$. Without augmentation, contact metrics are largely insensitive to the contact label ($\mathcal{T}^{\text{✔}}_{\text{near}}$and $\mathcal{T}^{\text{✘}}_{\text{near}}$produce similar values), showing the policy ignores the conditioning signal.

Per-motion takeaways from Fig. 6 and Table 9. The augmentation gap is largest on motions where contact is geometrically *not forced* by the keypoint reference:

- **Sit and squat** shows the most dramatic effect: under contact ✘($\mathcal{T}^{\text{✘}}_{\text{near}}$), No-aug still produces 0.46 contact bodies and $1.17\,\text{N}\cdot\text{s}$ impulse (nearly identical to its $\mathcal{T}^{\text{✔}}_{\text{near}}$value of $0.47/1.57$), meaning the policy ignores the contact label entirely; with augmentation, the same $\mathcal{T}^{\text{✘}}_{\text{near}}$drops to $0.08/0.05$, roughly a $6-23\times$ reduction.
- **Sit in front of table**, **Sit on table**, and **Lean on backrest II** follow the same pattern: No-aug barely modulates contact between $\mathcal{T}^{\text{✔}}_{\text{near}}$and $\mathcal{T}^{\text{✘}}_{\text{near}}$, while Ours produces a clear ON/OFF separation.
- **Step foot on chair**, by contrast, has nearly identical numbers for No-aug and Ours, because the keypoint reference itself forces foot contact regardless of the label—there is little room for augmentation to teach contact decoupling here.
- **Pick up box** shows that without augmentation the policy keeps grabbing the box even under contact ✘(No-aug $\mathcal{T}^{\text{✘}}_{\text{near}}$: 1.95 bodies, $1.85\,\text{N}\cdot\text{s}$); augmentation cuts these to $0.29/0.17$, allowing the same policy to follow a "don't touch" command on a manipulation motion.
- **Kick chair** is unaffected by the ablation (both variants ~ 0.01): the contact is so brief and the keypoint trajectory so determined by the kick swing that there is essentially no contact-vs-no-contact decision for the augmentation to teach.

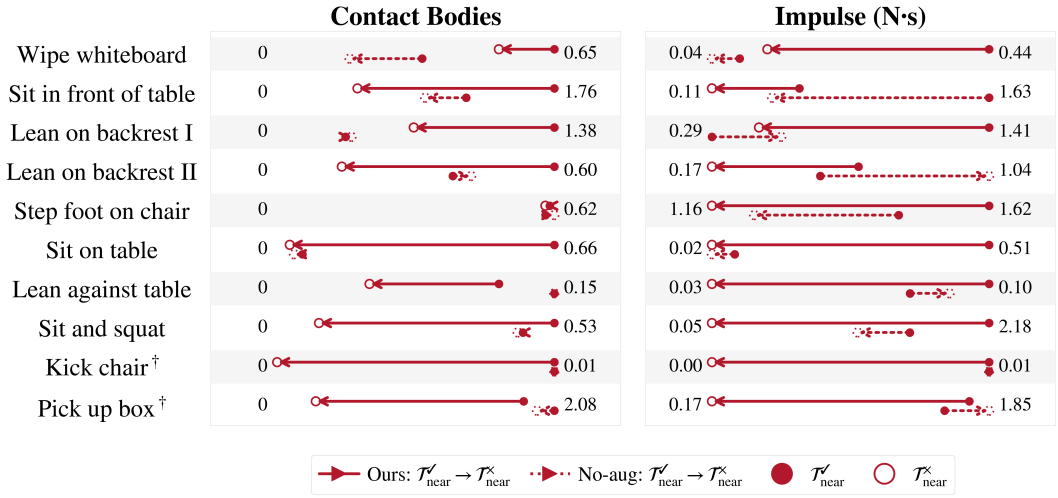

Figure 6: Per-motion ablation on paired-motion data augmentation, shown for both contact bodies and impulse. Solid arrows show our full method (Ours); dotted arrows show the no-augmentation baseline (No-aug). Both arrows go from $\mathcal{T}^{\checkmark}_{\text{near}}$ to $\mathcal{T}^{\times}_{\text{near}}$. Without augmentation, the arrows are short or even reverse direction, indicating the policy fails to follow the contact label.

Table 9: Ablation on paired-motion data augmentation across all motions. ($\mathcal{T}^{\checkmark}_{\text{near}}$= original keypoint with contact; $\mathcal{T}^{\times}_{\text{near}}$= original keypoint without contact (flipped)). Without augmentation, contact metrics are largely insensitive to the contact label (both $\mathcal{T}^{\checkmark}_{\text{near}}$and $\mathcal{T}^{\times}_{\text{near}}$produce similar values, whether high or low), showing the policy ignores the contact conditioning.

Motion	Variant	Contact bodies		Impulse (N·s)	
		$\mathcal{T}^{\checkmark}_{\text{near}}$	$\mathcal{T}^{\times}_{\text{near}}$	$\mathcal{T}^{\checkmark}_{\text{near}}$	$\mathcal{T}^{\times}_{\text{near}}$
Wipe whiteboard	No-aug	$0.34_{\pm 0.40}$	$0.17_{\pm 0.24}$	$0.08_{\pm 0.12}$	$0.04_{\pm 0.08}$
	Ours	$0.65_{\pm 0.45}$	$0.52_{\pm 0.44}$	$0.44_{\pm 0.37}$	$0.12_{\pm 0.13}$
Sit in front of table	No-aug	$1.20_{\pm 0.44}$	$0.94_{\pm 0.49}$	$1.63_{\pm 1.50}$	$0.44_{\pm 0.42}$
	Ours	$1.76_{\pm 0.33}$	$0.51_{\pm 0.64}$	$0.59_{\pm 0.22}$	$0.11_{\pm 0.21}$
Lean on backrest I	No-aug	$0.34_{\pm 0.51}$	$0.37_{\pm 0.40}$	$0.29_{\pm 0.66}$	$0.57_{\pm 1.19}$
	Ours	$1.38_{\pm 1.27}$	$0.68_{\pm 0.65}$	$1.41_{\pm 1.46}$	$0.48_{\pm 0.64}$
Lean on backrest II	No-aug	$0.38_{\pm 0.39}$	$0.42_{\pm 0.45}$	$0.51_{\pm 0.69}$	$1.04_{\pm 1.42}$
	Ours	$0.60_{\pm 0.39}$	$0.14_{\pm 0.24}$	$0.63_{\pm 0.55}$	$0.17_{\pm 0.37}$
Step foot on chair	No-aug	$0.60_{\pm 0.45}$	$0.62_{\pm 0.48}$	$1.47_{\pm 1.19}$	$1.23_{\pm 1.02}$
	Ours	$0.61_{\pm 0.48}$	$0.60_{\pm 0.48}$	$1.62_{\pm 1.33}$	$1.16_{\pm 1.01}$
Sit on table	No-aug	$0.06_{\pm 0.17}$	$0.04_{\pm 0.14}$	$0.06_{\pm 0.48}$	$0.02_{\pm 0.16}$
	Ours	$0.66_{\pm 0.47}$	$0.03_{\pm 0.16}$	$0.51_{\pm 0.60}$	$0.02_{\pm 0.20}$
Lean against table	No-aug	$0.15_{\pm 0.43}$	$0.15_{\pm 0.43}$	$0.08_{\pm 0.23}$	$0.09_{\pm 0.25}$
	Ours	$0.12_{\pm 0.30}$	$0.05_{\pm 0.22}$	$0.10_{\pm 0.51}$	$0.03_{\pm 0.16}$
Sit and squat	No-aug	$0.47_{\pm 0.50}$	$0.46_{\pm 0.46}$	$1.57_{\pm 1.66}$	$1.17_{\pm 1.72}$
	Ours	$0.53_{\pm 0.51}$	$0.08_{\pm 0.22}$	$2.18_{\pm 2.22}$	$0.05_{\pm 0.18}$
Kick chair†	No-aug	$0.01_{\pm 0.06}$	$0.01_{\pm 0.07}$	$0.01_{\pm 0.27}$	$0.01_{\pm 0.12}$
	Ours	$0.01_{\pm 0.08}$	$0.00_{\pm 0.05}$	$0.01_{\pm 0.12}$	$0.00_{\pm 0.06}$
Pick up box†	No-aug	$2.08_{\pm 0.92}$	$1.95_{\pm 0.93}$	$1.58_{\pm 1.13}$	$1.85_{\pm 1.26}$
	Ours	$1.85_{\pm 0.94}$	$0.29_{\pm 0.63}$	$1.73_{\pm 1.18}$	$0.17_{\pm 0.59}$

Table 8: Contact label controllability across all interaction motions. A single contact-conditioned policy is evaluated on four input trajectories per motion: $\mathcal{T}^{✔}_{\text{near}}$and $\mathcal{T}^{✘}_{\text{far}}$are matched (keypoint and label agree), while $\mathcal{T}^{✘}_{\text{near}}$and $\mathcal{T}^{✔}_{\text{far}}$are flipped (keypoint and label disagree). Object displacement is only reported for free-object motions (†). Bodies/impulse track the contact label (not the keypoint variant), indicating the policy follows the label rather than implicitly inferring contact from keypoints. The arrow beside each key joint marks whether its torque is intuitively expected to increase (↑) or decrease (↓) when the commanded contact is turned off.

Motion	Trajectory	Contact bodies	Ref	Impulse (N·s)	Key torque (N·m)
Wipe whiteboard	$\mathcal{T}^{✔}_{\text{near}}$	$0.65_{\pm 0.45}$	0.34	$0.44_{\pm 0.37}$	$-2.07_{\pm 1.61}$ (R wrist pitch ↑)
	$\mathcal{T}^{✘}_{\text{far}}$	$0.25_{\pm 0.25}$	0.00	$0.04_{\pm 0.06}$	$-0.44_{\pm 0.30}$ (R wrist pitch ↑)
	$\mathcal{T}^{✘}_{\text{near}}$	$0.52_{\pm 0.44}$	0.34	$0.12_{\pm 0.13}$	$-0.76_{\pm 0.60}$ (R wrist pitch ↑)
	$\mathcal{T}^{✔}_{\text{far}}$	$0.33_{\pm 0.39}$	0.00	$0.07_{\pm 0.11}$	$-0.53_{\pm 0.46}$ (R wrist pitch ↑)
Sit in front of table	$\mathcal{T}^{✔}_{\text{near}}$	$1.76_{\pm 0.33}$	2.68	$0.59_{\pm 0.22}$	$-1.46_{\pm 6.26}$ (waist pitch ↓)
	$\mathcal{T}^{✘}_{\text{far}}$	$0.08_{\pm 0.19}$	0.00	$0.01_{\pm 0.09}$	$-12.12_{\pm 5.08}$ (waist pitch ↓)
	$\mathcal{T}^{✘}_{\text{near}}$	$0.51_{\pm 0.64}$	2.68	$0.11_{\pm 0.21}$	$-14.91_{\pm 7.54}$ (waist pitch ↓)
	$\mathcal{T}^{✔}_{\text{far}}$	$0.57_{\pm 0.46}$	0.00	$0.58_{\pm 2.70}$	$-4.65_{\pm 8.78}$ (waist pitch ↓)
Lean on backrest I	$\mathcal{T}^{✔}_{\text{near}}$	$1.38_{\pm 1.27}$	2.57	$1.41_{\pm 1.46}$	$19.2_{\pm 8.4}$ (hip ↑)
	$\mathcal{T}^{✘}_{\text{far}}$	$0.51_{\pm 0.49}$	1.36	$0.16_{\pm 0.27}$	$29.4_{\pm 17.3}$ (hip ↑)
	$\mathcal{T}^{✘}_{\text{near}}$	$0.68_{\pm 0.65}$	2.57	$0.48_{\pm 0.64}$	$24.1_{\pm 12.5}$ (hip ↑)
	$\mathcal{T}^{✔}_{\text{far}}$	$0.60_{\pm 0.48}$	1.36	$0.28_{\pm 0.26}$	$26.7_{\pm 13.3}$ (hip ↑)
Lean on backrest II	$\mathcal{T}^{✔}_{\text{near}}$	$0.60_{\pm 0.39}$	0.70	$0.63_{\pm 0.55}$	$26.7_{\pm 12.0}$ (hip ↑)
	$\mathcal{T}^{✘}_{\text{far}}$	$0.00_{\pm 0.02}$	0.00	$0.00_{\pm 0.06}$	$40.7_{\pm 18.9}$ (hip ↑)
	$\mathcal{T}^{✘}_{\text{near}}$	$0.14_{\pm 0.24}$	0.70	$0.17_{\pm 0.37}$	$32.2_{\pm 16.6}$ (hip ↑)
	$\mathcal{T}^{✔}_{\text{far}}$	$0.02_{\pm 0.11}$	0.00	$0.03_{\pm 0.27}$	$40.8_{\pm 21.1}$ (hip ↑)
Step foot on chair	$\mathcal{T}^{✔}_{\text{near}}$	$0.61_{\pm 0.48}$	0.64	$1.62_{\pm 1.33}$	$+0.10_{\pm 11.62}$ (L hip pitch ↓)
	$\mathcal{T}^{✘}_{\text{far}}$	$0.62_{\pm 0.48}$	0.00	$1.60_{\pm 1.31}$	$-26.66_{\pm 19.59}$ (L hip pitch ↓)
	$\mathcal{T}^{✘}_{\text{near}}$	$0.60_{\pm 0.48}$	0.64	$1.16_{\pm 1.01}$	$-4.13_{\pm 14.82}$ (L hip pitch ↓)
	$\mathcal{T}^{✔}_{\text{far}}$	$0.61_{\pm 0.48}$	0.00	$1.59_{\pm 1.32}$	$-2.77_{\pm 11.17}$ (L hip pitch ↓)
Sit on table	$\mathcal{T}^{✔}_{\text{near}}$	$0.66_{\pm 0.47}$	0.53	$0.51_{\pm 0.60}$	$+2.43_{\pm 5.29}$ (waist roll ↓)
	$\mathcal{T}^{✘}_{\text{far}}$	$0.04_{\pm 0.18}$	0.00	$0.03_{\pm 0.25}$	$-5.99_{\pm 4.97}$ (waist roll ↓)
	$\mathcal{T}^{✘}_{\text{near}}$	$0.03_{\pm 0.16}$	0.53	$0.02_{\pm 0.20}$	$-5.91_{\pm 5.01}$ (waist roll ↓)
	$\mathcal{T}^{✔}_{\text{far}}$	$0.67_{\pm 0.46}$	0.00	$0.48_{\pm 0.43}$	$+1.61_{\pm 5.29}$ (waist roll ↓)
Lean against table	$\mathcal{T}^{✔}_{\text{near}}$	$0.12_{\pm 0.30}$	0.21	$0.10_{\pm 0.51}$	$8.4_{\pm 6.4}$ (waist pitch ↓)
	$\mathcal{T}^{✘}_{\text{far}}$	$0.05_{\pm 0.21}$	0.02	$0.03_{\pm 0.16}$	$7.8_{\pm 8.7}$ (waist pitch ↓)
	$\mathcal{T}^{✘}_{\text{near}}$	$0.05_{\pm 0.22}$	0.21	$0.03_{\pm 0.16}$	$7.9_{\pm 8.9}$ (waist pitch ↓)
	$\mathcal{T}^{✔}_{\text{far}}$	$0.13_{\pm 0.43}$	0.02	$0.09_{\pm 0.32}$	$8.0_{\pm 9.0}$ (waist pitch ↓)
Sit and squat	$\mathcal{T}^{✔}_{\text{near}}$	$0.53_{\pm 0.51}$	1.28	$2.18_{\pm 2.22}$	$19.0_{\pm 7.5}$ (hip ↓)
	$\mathcal{T}^{✘}_{\text{far}}$	$0.01_{\pm 0.05}$	1.28	$0.02_{\pm 0.24}$	$15.1_{\pm 6.9}$ (hip ↓)
	$\mathcal{T}^{✘}_{\text{near}}$	$0.08_{\pm 0.22}$	1.28	$0.05_{\pm 0.18}$	$11.4_{\pm 4.7}$ (hip ↓)
	$\mathcal{T}^{✔}_{\text{far}}$	$0.81_{\pm 0.80}$	1.28	$1.88_{\pm 2.19}$	$13.8_{\pm 5.3}$ (hip ↓)
Kick chair†	$\mathcal{T}^{✔}_{\text{near}}$	$0.01_{\pm 0.08}$	0.09	$0.01_{\pm 0.12}$	$0.31_{\pm 0.15}$ (disp. ↓)
	$\mathcal{T}^{✘}_{\text{far}}$	$0.01_{\pm 0.08}$	0.04	$0.01_{\pm 0.18}$	$0.16_{\pm 0.13}$ (disp. ↓)
	$\mathcal{T}^{✘}_{\text{near}}$	$0.00_{\pm 0.05}$	0.09	$0.00_{\pm 0.06}$	$0.26_{\pm 0.10}$ (disp. ↓)
	$\mathcal{T}^{✔}_{\text{far}}$	$0.01_{\pm 0.06}$	0.04	$0.01_{\pm 0.09}$	$0.17_{\pm 0.10}$ (disp. ↓)
Pick up box†	$\mathcal{T}^{✔}_{\text{near}}$	$1.85_{\pm 0.94}$	1.89	$1.73_{\pm 1.18}$	$0.49_{\pm 0.47}$ (disp. ↓)
	$\mathcal{T}^{✘}_{\text{far}}$	$0.36_{\pm 0.68}$	0.00	$0.21_{\pm 0.72}$	$0.18_{\pm 0.26}$ (disp. ↓)
	$\mathcal{T}^{✘}_{\text{near}}$	$0.29_{\pm 0.63}$	1.89	$0.17_{\pm 0.59}$	$0.20_{\pm 0.24}$ (disp. ↓)
	$\mathcal{T}^{✔}_{\text{far}}$	$1.22_{\pm 1.20}$	0.00	$1.17_{\pm 1.40}$	$0.50_{\pm 0.46}$ (disp. ↓)

† Free-object motion; the torque column reports object displacement (disp., m) instead.