PrioriTouch: Adapting to User Contact Preferences for Whole-Arm Physical Human-Robot Interaction

Rishabh Madan[1], Jiawei Lin[1], Mahika Goel[1], Amber Li[1], Angchen Xie[1], Xiaoyu Liang[1], Marcus Lee[1], Justin Guo[1], Pranav N. Thakkar[1], Rohan Banerjee[1], Jose Barreiros[2], Kate Tsui[2], Tom Silver[1], Tapomayukh Bhattacharjee[1]
[1]Cornell University, [2]Toyota Research Institute

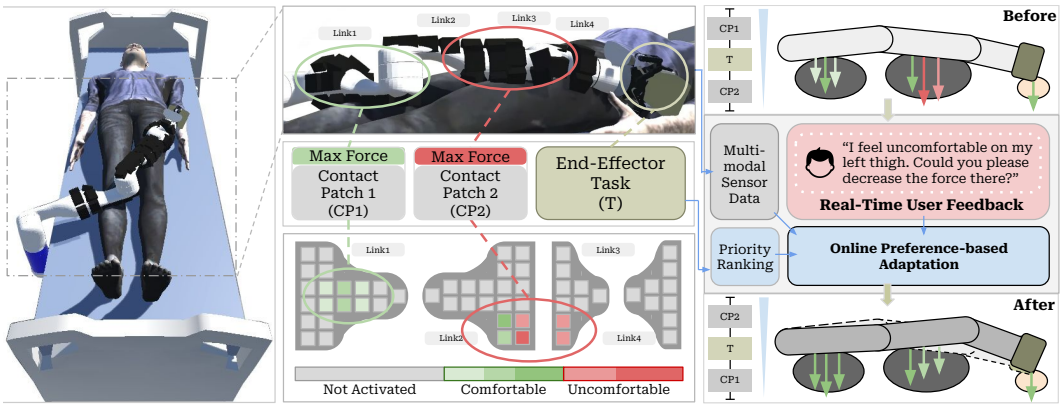

Figure 1: Overview of PrioriTouch: A hierarchical controller coupled with simulation-in-the-loop online contact preference learning aids in personalizing multi-contact physical human-robot interaction for safe and comfortable whole-arm manipulation.

Abstract: Physical human–robot interaction (pHRI) requires robots to adapt to individual contact preferences, such as where and how much force is applied. Identifying preferences is difficult for a single contact; with whole-arm interaction involving multiple simultaneous contacts between the robot and human, the challenge is greater because different body parts can impose incompatible force requirements. In caregiving tasks, where contact is frequent and varied, such conflicts are unavoidable. With multiple preferences across multiple contacts, no single solution can satisfy all objectives–trade-offs are inherent, making prioritization essential. We present PrioriTouch, a framework for ranking and executing control objectives across multiple contacts. PrioriTouch can prioritize from a general collection of controllers, making it applicable not only to caregiving scenarios such as bed bathing and dressing but also to broader multi-contact settings. Our method combines a novel learning-to-rank approach with hierarchical operational space control, leveraging simulation-in-the-loop rollouts for data-efficient and safe exploration. We conduct a user study on physical assistance preferences, derive personalized comfort thresholds, and incorporate them into PrioriTouch. We evaluate PrioriTouch through extensive simulation and real-world experiments, demonstrating its ability to adapt to user contact preferences, maintain task performance, and enhance safety and comfort. Website: https://emprise.cs.cornell.edu/prioritouch.

Keywords: Physical Human-Robot Interaction, Online Preference Learning, Assistive Robotics

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

1 Introduction

Physical human–robot interaction (pHRI) requires physical contact. Contact is not uniform: individuals have distinct preferences for acceptable forces and contact locations [1, 2, 3, 4, 5]. For pHRI to be safe and effective, robots must personalize their behavior, and a critical aspect of personalization is contact preferences. Even for a single contact, identifying and respecting these preferences while ensuring task success is challenging. Many physical robot caregiving tasks, such as bathing [6, 7], dressing [8, 9], and transferring [10], require whole-arm pHRI [11], where multiple segments of the robot arm simultaneously touch the human body. For example, during bed bathing (Fig. 1), the robot may need to reach over a user to wipe the upper arm while maintaining comfortable forces on the torso and shoulder. Although whole-arm manipulation expands workspace and improves maneuverability, it also exacerbates conflicts: different body parts can impose incompatible force requirements, and no single policy can satisfy all objectives.

To bootstrap personalization, we elicit population-level contact preferences offline and use them to seed a conservative base policy. However, a one-size-fits-all policy is insufficient: (i) stated preferences can diverge from realized comfort under true contact (pressure/shear, approach, speed, duration); and (ii) preferences are context-dependent and time-varying (posture, clothing, fatigue). Therefore, online interaction is necessary to accommodate individual preferences. Experimenting directly with the user is risky and inefficient because each update can involve repeated physical contact and multiple feedback exchanges. This increases the user's cognitive workload, prolongs the interaction, and may cause discomfort when forces are suboptimal or excessive.

We introduce PrioriTouch, a framework that casts contact preference learning as a learning-to-rank problem over control objectives. Given a reference trajectory produced by a high-level policy (e.g., a contact-aware planner generating end-effector or joint-space paths), PrioriTouch instantiates pose-tracking and force-regulation objectives from the current contact state. We develop LinUCB-Rank, a contextual bandit that learns a priority policy; H-OSC [12] then executes this ordering as a null space hierarchy, translating high-level preferences into low-level control. We initialize the policy with conservative priors derived from population-level user-study statistics. During interaction, LinUCB-Rank adapts the ordering online using sparse user feedback while safely refining the policy via simulation-in-the-loop learning before deploying it in real-world interactions. The framework is controller-agnostic: it can rank heterogeneous objectives, enabling principled trade-offs across simultaneous objectives.

We evaluate PrioriTouch across simulated and real-world environments, progressively increasing in complexity and realism. First, we design a simplified simulation scenario with predefined contacts and a static end-effector pose to isolate and specifically assess LinUCB-Rank's ability to learn user contact preferences. Second, we demonstrate PrioriTouch in a simulated caregiving scenario involving robot-assisted bed bathing, requiring whole-arm contact to safely wipe a user's limbs. Third, we showcase our approach's capability in intricate multi-contact scenarios through a real-world 3D goal-reaching maze with multiple vertical cylinders representing distinct body-part contact preferences. Finally, we validate PrioriTouch's practical feasibility by performing a realistic caregiving task in a user study with human subjects.

Our contributions are summarized as follows:

- We propose PrioriTouch, a framework that formulates contact preference learning as a ranking problem over control objectives and executes the learned priority ordering as a null space hierarchy via H-OSC for whole-arm pHRI.
- We introduce LinUCB-Rank, a contextual bandit that learns priority orderings from sparse user feedback while accounting for inter-objective coupling in hierarchical control.
- We enable safe and data-efficient learning through simulation-in-the-loop validation, where candidate priority updates are tested in a digital twin before real-world deployment.
- We conduct a user study to inform realistic models of contact preferences for robot-initiated touch, which we leverage to simulate authentic user feedback in our evaluation.

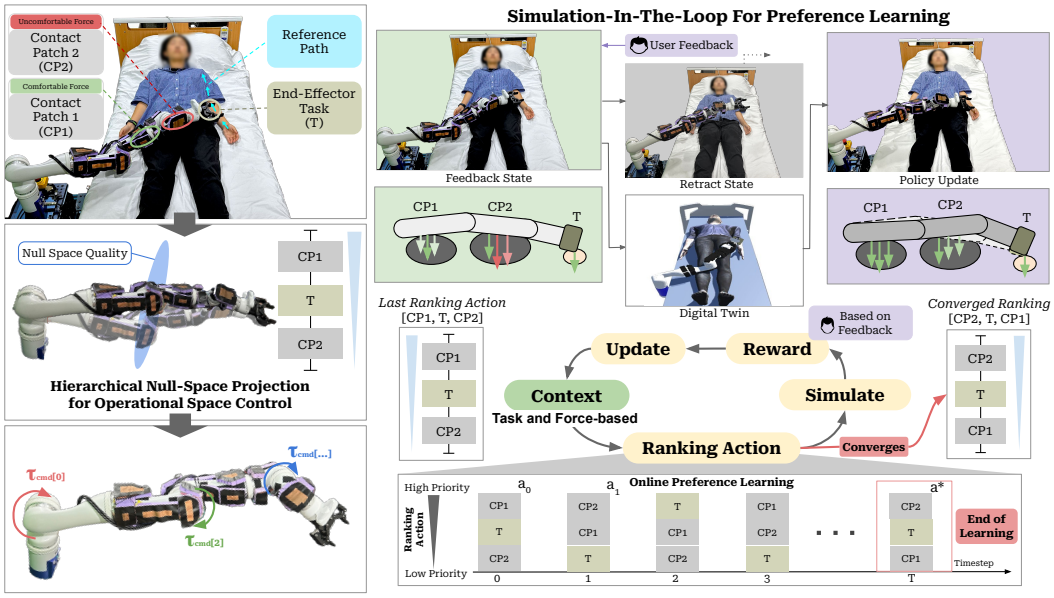

Figure 2: We implement H-OSC which uses a learned priority ranking to modulate multiple contacts during WAM. We propose LinUCB-Rank, a learning-to-rank contextual bandit algorithm, to update the prioritization policy using realistic user feedback and simulated interactions.

- We evaluate PrioriTouch through extensive simulation, real-world experiments, and a realistic caregiving user study, demonstrating effective adaptation to individual contact preferences without compromising task performance or comfort.

Our framework integrates user contact preference learning with low-level control by parameterizing operational space control using the outputs of a learned ranking policy. This structured integration ensures that high-level feedback is directly translated into low-level force regulation and pose tracking, effectively bridging the gap between user preferences and robot control.

2 Related Work

Whole-Arm Manipulation. Recent whole-arm manipulation (WAM) systems leverage distributed tactile skins to localize and regulate contact [13, 14, 15, 16]. These capabilities have been applied to manipulating heavy or bulky objects [17, 15], navigating clutter [18], and providing social touch [19]. In pHRI scenarios, however, explicitly managing multiple, potentially conflicting contact-force objectives remains challenging. Park et al. [20] proposed a hierarchical multi-contact controller that optimizes all contact forces with equal weighting, but their formulation does not account for preference-related ordering or any dynamic prioritization. In contrast, we build on hierarchical control but dynamically prioritize contact points using user feedback, enabling robots to accommodate individualized comfort and safety preferences across different users and body regions.

Online Preference Learning in HRI. While several works have explored online user preference learning in HRI, spanning high-level action sequences [21] to low-level parameters [22] and combinations thereof [23, 24], existing work has not explicitly addressed the challenge of learning user contact preferences. Most previous research either assumes human-initiated contact [25, 26, 27, 28] or focuses on user preferences that can be safely explored by adjusting non-contact parameters, such as trajectory speed [23]. However, in pHRI tasks requiring robot-initiated physical contact, direct trial-and-error exploration to determine acceptable forces is impractical and potentially unsafe. Exploring contact preferences in multi-contact settings is especially challenging, as mitigating discomfort at one body region may inadvertently increase forces at another, making conflict resolution substantially harder. We address this gap by casting contact preference learning as a learning-to-rank problem over control objectives, allowing us to reason about preferences pertaining to multiple si-

multaneous contacts in a structured manner. We also introduce simulation-in-the-loop learning, allowing robots to refine candidate prioritization policies virtually before resuming real interaction, enabling safe and personalized estimation of user contact preferences in pHRI.

3 Problem Statement

We ground the problem in whole-arm pHRI (e.g., bed bathing; Fig. 2). An n-DoF robot follows a reference path $\mathcal{P}_{\text{path}} = \{\mathbf{p}(1), \ldots, \mathbf{p}(N)\}$, where each $\mathbf{p}(i) \in \mathbb{R}^6$ is an end-effector pose. At time t, the joint configuration is $\mathbf{q}_t$ and current end-effector pose is $\mathbf{p}_t^{\text{curr}}$. The controller selects a waypoint index i_t and defines the desired pose as $\mathbf{p}_t^{\text{des}} \triangleq \mathbf{p}(i_t)$. The controller applies joint torques $\boldsymbol{\tau}_t \in \mathbb{R}^n$ to reduce the pose error and execute the task.

We model the human as a discrete set of body parts $\mathcal{B} = \{b_1, \ldots, b_m\}$ with pose configuration $\mathbf{h}_t$ at time t. The contact set at time t is $\mathcal{C}_t = \{ c_t(1), c_t(2), \ldots, c_t(K_t) \}$, where $K_t = |\mathcal{C}_t|$ is the number of contacts present at time t and $c_t(k)$ denotes the k-th contact. Each contact $c_t(k) \in \mathcal{C}_t$ is associated with a body part via $\psi : \mathcal{C}_t \to \mathcal{B}$ and produces a force vector $\mathbf{f}_t(c_t(k)) \in \mathbb{R}^3$. For a body part $b \in \mathcal{B}$, the set of contacts on b is given by $\mathcal{C}_t(b) \triangleq \{ c_t(k) \in \mathcal{C}_t \mid \psi(c_t(k)) = b \}$, and the aggregate force magnitude on b is $f_t(b) \triangleq \max_{c_t(k) \in \mathcal{C}_t(b)} \left\|\mathbf{f}_t(c_t(k))\right\|$, with $f_t(b) = 0$ if $\mathcal{C}_t(b) = \varnothing$. The set of body parts in contact at time t is $\mathcal{B}_t \triangleq \{ b \in \mathcal{B} \mid \mathcal{C}_t(b) \neq \varnothing \}$. For convenience, we jointly define the robot–human state as $\mathsf{s}_t = (\mathbf{q}_t, \mathbf{h}_t, \mathcal{C}_t)$.

User contact preferences are represented by an unknown model $\mathcal{H}$, which assigns a comfort threshold $f^{\max}(b)$ to each body part $b \in \mathcal{B}$ (Sec. 5.3). Although $\mathcal{H}$ is not directly observable, the robot receives sparse online feedback $\phi_t = (b, \delta)$, where $b \in \mathcal{B}$ identifies the body part of concern, and δ is a one-hot vector encoding the requested change. We fix the order as $\delta = [\texttt{dec-Large}, \texttt{dec-Small}, \texttt{inc-Small}, \texttt{inc-Large}]$, so a given δ selects decrease/increase and large/small. For example, "my stomach hurts a bit" implies $(\texttt{abdomen}, [0, 1, 0, 0])$, i.e., a small decrease on abdomen.

At each timestep t, the controller balances two objectives. The pose-tracking objective penalizes deviation from the desired waypoint, $J_{\mathbf{p}}(t) = \|\mathbf{p}_t^{\text{curr}} - \mathbf{p}_t^{\text{des}}\|$. The force-regulation objective penalize comfort-threshold exceedances for each contacted body part, $J_{f,b}(t) = \max\big(0,\ f_t(b) - f^{\max}(b)\big)$ for $b \in \mathcal{B}_t$. The active objective set is $\mathcal{J}_t = \{J_{\mathbf{p}}(t)\} \cup \{J_{f,b}(t) \mid b \in \mathcal{B}_t\}$.

Because objectives in $\mathcal{J}_t$ may conflict depending on s_t, they are executed according to a time-varying ordering $\sigma_t \in \mathfrak{S}(\mathcal{J}_t)$ (denotes all permutations of active objectives) using H-OSC (Sec. 4.1). A ranking policy $\pi_\sigma(t)$ predicts σ_t at run time with the goal of tracking $\mathcal{P}_{\text{path}}$ while respecting comfort thresholds with minimal user feedback. We formulate this as a contextual bandit (CB) problem. At each timestep t: the context $\mathbf{z}_t$ is derived from the observable state s_t; the action is the ranking σ_t drawn from the action space $\mathcal{A} = \mathfrak{S}(\mathcal{J}_t)$; and the reward $r_t \in \mathbb{R}$ is a scalar feedback signal returned after executing σ_t. The CB thus selects σ_t, and H-OSC executes it as a strict hierarchy. Details of how $\mathbf{z}_t$ and r_t are constructed are provided in Sec. 4.2.

4 PrioriTouch: Contact Preference Learning for Whole-arm Manipulation

PrioriTouch casts contact preference learning as a *learning-to-rank* problem over control objectives. LinUCB-Rank, a contextual bandit, learns a priority ordering that H-OSC [12] executes as a null space hierarchy, translating high-level contact preferences into low-level control. For safety and data efficiency, candidate re-orderings are validated in a digital twin before deployment. We now detail the components that enable this framework.

4.1 Hierarchical Operational Space Control for WAM

H-OSC [12] enables robots to manage multiple competing objectives by enforcing a strict hierarchy, making it particularly suitable for WAM scenarios with simultaneous pose tracking and force regulation. Given a ranking σ_t, H-OSC recursively projects lower-priority objectives into the null space of higher-priority ones, ensuring that higher-priority tasks are executed without interference.

Formally, the robot dynamics can be written as $\mathbf{M}(\mathbf{q}_t)\ddot{\mathbf{q}}_t + \mathbf{C}(\mathbf{q}_t, \dot{\mathbf{q}}_t) + \mathbf{g}(\mathbf{q}_t) = \boldsymbol{\tau}_t$. H-OSC generates torques as $\boldsymbol{\tau}_t = \sum_{j \in \mathcal{J}_t} \mathbf{N}_{j,t}\, \boldsymbol{\tau}_{j,t}$, where $\boldsymbol{\tau}_{j,t} = \bar{\mathbf{J}}_j^\top \mathbf{F}_{j,t}$ and $\mathbf{N}_{j,t}$ is the recursively computed null space projector.

Operational-space forces $\mathbf{F}_{j,t}$ are derived from desired task accelerations. For pose tracking, we use $\ddot{\mathbf{x}}_t = \mathbf{K}_p(\mathbf{p}_t^{\text{des}} - \mathbf{p}_t^{\text{curr}}) - \mathbf{K}_d \dot{\mathbf{p}}_t^{\text{curr}}$, where $\mathbf{K}_p$ and $\mathbf{K}_d$ are proportional and derivative pose gains. For force regulation at body part $b \in \mathcal{B}_t$, we use $\ddot{\mathbf{x}}_t = -\mathbf{K}_f(f_t(b) - f^{\max}(b))$, where $\mathbf{K}_f$ is a scalar force gain. Detailed derivations appear in Appendix A.1.

4.2 Learning to Rank using Contextual Bandits

Enumerating and testing all permutations of objectives in $\mathcal{J}_t$ is computationally infeasible. We therefore formulate control objective prioritization for H-OSC as a *learning-to-rank* problem guided by user feedback. We use a *contextual bandit* (CB) [29] to balance exploration and exploitation, refining the ranking policy iteratively from sparse feedback.

Preliminaries. At each timestep t, a contextual bandit observes a context $\mathbf{z}_t$, selects an action $\sigma_t \in \mathcal{A}_t$, and receives a reward $r_t(\mathbf{z}_t, \sigma_t)$. The goal is to minimize the cumulative regret $\text{Regret}(T) = \sum_{t=1}^{T} \max_{\sigma \in \mathcal{A}_t} \mathbb{E}[r_t(\mathbf{z}_t, \sigma)] - \sum_{t=1}^{T} \mathbb{E}[r_t(\mathbf{z}_t, \sigma_t)]$ LinUCB [30, 31] assumes a linear reward model $\mathbb{E}[r_t \mid \mathbf{z}_t, a] = \boldsymbol{\theta}_a^\top \mathbf{z}_t$ for each atomic action a, and selects actions using an upper confidence bound (UCB): $\text{UCB}_{t,a} = \hat{\boldsymbol{\theta}}_a^\top \mathbf{z}_t + \alpha\sqrt{\mathbf{z}_t^\top \mathbf{A}_a^{-1} \mathbf{z}_t}$, where $\mathbf{A}_a$ is a covariance matrix and $\alpha > 0$ controls exploration.

In our setting, the action is a full ranking σ_t. Standard LinUCB would choose a single best element a per round, which is insufficient for prioritizing multiple coupled objectives in H-OSC. LinUCB-Rank instead constructs σ_t sequentially, explicitly accounting for interdependencies: assigning higher priority to one objective reduces the null space available for those below it.

Algorithm 1 LinUCB-Rank

Require: Objective set $\mathcal{J}_t$, exploration parameter $\alpha > 0$
Initialize $(\hat{\boldsymbol{\theta}}_i, \mathbf{A}_i, \mathbf{b}_i)$ for all $i = 1, \ldots, |\mathcal{J}_t|$
for $t = 1$ to T **do**
  $\mathcal{J}_t^{(0)} \leftarrow \mathcal{J}_t,\ \sigma_t \leftarrow []$
  Initialize ranking history $\mathbf{z}_{t,h} \leftarrow \mathbf{0}$
  for $i = 1$ to $|\mathcal{J}_t|$ **do**
    Construct context $\mathbf{z}_{t,i}$ from state s_t
    for each $a_{t,i} \in \mathcal{J}_t^{(i-1)}$ **do**
      $\text{UCB}_{a_{t,i}} \leftarrow \hat{\boldsymbol{\theta}}_i^\top \mathbf{z}_{t,i} + \alpha\sqrt{\mathbf{z}_{t,i}^\top \mathbf{A}_i^{-1} \mathbf{z}_{t,i}}$
    end for
    $a_{t,i} \leftarrow \arg\max_{a_{t,i} \in \mathcal{J}_t^{(i-1)}} \text{UCB}_{a_{t,i}}$
    $\sigma_t \leftarrow \sigma_t + a_{t,i}$
    $\mathcal{J}_t^{(i)} \leftarrow \mathcal{J}_t^{(i-1)} \setminus \{a_{t,i}\}$
  end for
  Execute H-OSC with ranking σ_t
  Observe per-slot rewards $\{r_{t,i}\}_{i=1}^{|\mathcal{J}_t|}$
  for $i = 1$ to $|\mathcal{J}_t|$ **do**
    $\mathbf{A}_i \leftarrow \mathbf{A}_i + \mathbf{z}_{t,i}\mathbf{z}_{t,i}^\top$
    $\mathbf{b}_i \leftarrow \mathbf{b}_i + r_{t,i}\mathbf{z}_{t,i}$
    $\hat{\boldsymbol{\theta}}_i \leftarrow \mathbf{A}_i^{-1}\mathbf{b}_i$
  end for
end for

Contextual Bandits for Ranking. We propose *LinUCB-Rank* (Alg. 1), which extends LinUCB to learn a ranking policy $\pi_\sigma(t)$ by sequentially assigning positions within each timestep. At every timestep t, LinUCB-Rank builds a ranking $\sigma_t = (a_{t,1}, \ldots, a_{t,|\mathcal{J}_t|})$ by selecting, at each slot i, the objective $a_{t,i} \in \mathcal{J}_t^{(i-1)}$ with the highest UCB among the remaining candidates. While Alg. 1 assumes a fixed set of objectives, $\mathcal{J}_t$ may vary due to changing contacts $\mathcal{C}_t$. We therefore rank a *fixed surrogate set*, i.e., the set of body parts $\mathcal{B}$; run LinUCB-Rank over $\mathcal{B}$ to obtain an ordering ρ_t, then sort active objectives by this order.

The context $\mathbf{z}_{t,i}$, shared across all candidates at rank i, is derived from s_t and includes three parts: (1) **Max force per body part**, $\mathbf{z}_{t,f} \in \mathbb{R}^m$, with $[\mathbf{z}_{t,f}]_b = f_t(b)$; (2) **Pose error**, $\mathbf{z}_{t,p} = \|\mathbf{p}_t^{\text{curr}} - \mathbf{p}_t^{\text{des}}\|$; (3) **Ranking state**, $\mathbf{z}_{t,h} \in \mathbb{R}^m$, an indicator vector that encodes the positions already assigned to each body part (0 if unassigned).

After constructing σ_t, H-OSC executes it. The robot then observes sensor data and user feedback ϕ_t, which together define per-slot rewards $r_{t,i}$ for each element $a_{t,i}$. Each reward has two terms: (1) **Feedback alignment** ($r_{\phi_t,i}$), positive if $a_{t,i}$ matches the body part indicated in ϕ_t and penalized

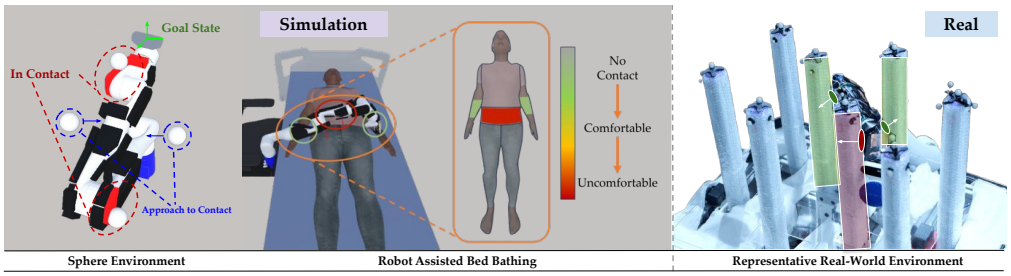

Figure 3: Simulation environment and Hardware testbed: (*Left*) Sphere environment with multiple spheres colliding with the robot's arm to test force modulation. (*Middle*) Simulated robot-assisted bed-bathing scenario with whole-arm contact to wipe a user's limbs. (*Right*) Real-world 3D goal-reaching maze with vertical cylinders, each representing a unique body part with its associated contact preference.

otherwise depending on rank deviation; (2) **Threshold violation** ($r_{t,f,i}$), penalizing excess force as $r_{t,f,i} = -w_f \max_{b\in\mathcal{B}}\big(0,\ f_t(b) - f^{\max}(b)\big)$, where $w_f > 0$. The total reward is $r_{t,i} = r_{\phi_t,i} + r_{t,f,i}$.

4.3 Simulation-in-the-Loop Preference Learning

At run time, H-OSC generates low-level control commands by executing an objective hierarchy produced by the ranking policy $\pi_\sigma(t)$. This policy is updated online by LinUCB-Rank from sparse user feedback. Because LinUCB-Rank is a contextual bandit, it must *explore* alternative orderings to align with user preferences. Performing this exploration directly on the user can induce large transients: in H-OSC, promoting one objective reduces the null space available to the rest, potentially increasing forces elsewhere.

To keep adaptation safe and data-efficient, PrioriTouch adopts a *simulation-in-the-loop* approach, performing high-risk exploration in a digital twin and deploying the converged policy on the real system. Specifically, the robot performs the following steps:

Simulation-in-the-Loop Preference Learning

1. **Record feedback:** Receive user feedback ϕ_t, update threshold $f^{\max}(b)$, and log state s_t.
2. **Retract safely:** Move the robot to a predefined safe configuration.
3. **Simulate exploration:** Set digital twin at state s_t.
4. **Learn preferences:** Iteratively refine policy $\pi_\sigma(t)$ using LinUCB-Rank (Sec. 4.2).
5. **Resume operation:** Upon convergence, resume real-world interaction with updated $\pi_\sigma(t)$.

Exploration occurs off-body in the twin, reducing the need for real-world user interactions to personalize the hierarchy. Only orderings that perform well under the simulated reward signals are deployed, yielding safer and more data-efficient adaptation.

5 Experiments

We evaluate *PrioriTouch* across three settings: (1) **simulation** to selectively evaluate learning and force modulation effects, (2) a **hardware testbed** to assess it on complex trajectories in the real world, and (3) **user studies** to model contact preferences and evaluate PrioriTouch in a real-world pHRI scenario. We use RCareWorld [32] for simulation and a Kinova Gen3 arm with distributed tactile sensing for all experiments; full system details are in the Appendix A.4.

Baselines. We compare LinUCB-Rank against the following methods:

- **Fixed Priority (*FP*)**: Maintains a static priority ordering derived from aggregate user-study preferences (Sec. 5.3), without adapting based on feedback.
- **Heuristic-based Priority (*HP*)**: Starts from the same initial ordering as *FP* but reorders objectives based on violations of force thresholds. For multiple violations, contacts are prioritized based on the magnitude of the violations.
- **LinUCB (Sorted)**: Independently learns contact force preferences using LinUCB, by sorting estimated upper confidence bounds. This baseline ignores dependencies between objectives.

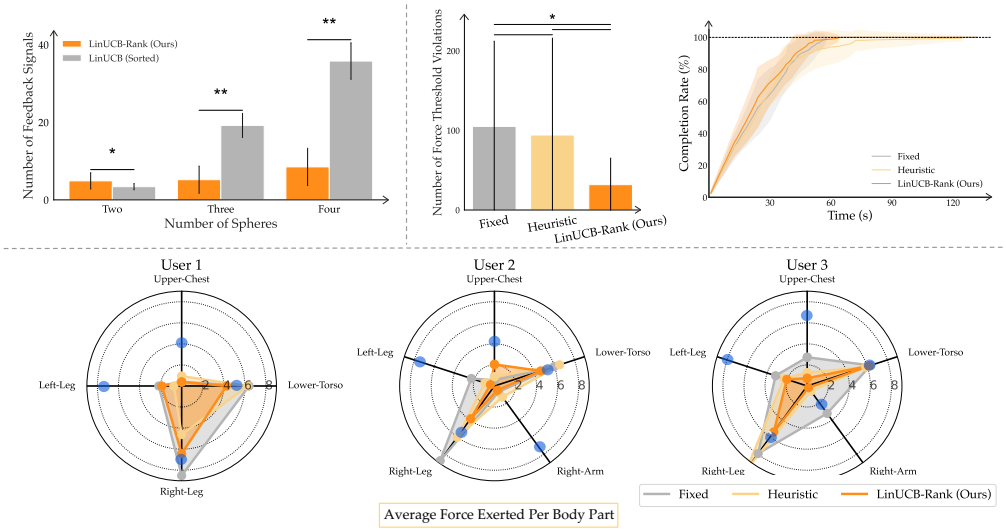

Figure 4: LinUCB-Rank requires fewer feedback signals (*Top-Left*), fewer violations (*Top-Right*), and obeys varying preferences related to force thresholds (denoted by a blue dot) across users (*Bottom*). $*$ and $**$ denote statistically significant differences with $p_{0.05}$ and $p_{0.005}$ respectively.

Evaluation Metrics. We assess performance using:

- Number of User Feedback Signals: Number of ϕ_t received before achieving correct ordering.
- Force Threshold Violation: Number of timesteps with $f_t(b) > f^{\max}(b)$.
- Time to Task Completion: Time taken to track the entire reference path $\mathcal{P}_{\text{path}}$.
- Average Force Exerted Per Body Part: How well a method respects user force thresholds for different body parts across different paths.

5.1 Simulation Experiments

(a) ***Sphere Environment.*** We simulate 2–4 spheres moving toward a stationary arm (Fig. 3, *Left*) to isolate force regulation. Each sphere represents a body part with predefined priorities and comfort thresholds; feedback is simulated accordingly (see Appendix A.2). As shown in Fig. 4 (*Top-Left*), *LinUCB-Rank* attains better sample efficiency than *LinUCB (Sorted)* as the number of spheres increases; with only two spheres, the simpler baseline is competitive, but the advantages of iterative ranking grow with multi-contact coupling.

(b) ***Robot-assisted Bed Bathing.*** We simulate a wheelchair-mounted arm performing a sponge-wiping task on a user lying in a hospital bed (Fig. 3, *Middle*). The path induces incidental whole-arm contacts with seven body regions. We evaluate *LinUCB-Rank* against *FP* and *HP* over four trajectories and three user models with varied preferences. Results (Fig. 4) show significantly fewer threshold violations than *FP* and *HP* (paired t-test, $p < 0.05$). Unlike *HP*, which reorders abruptly on violation spikes, *LinUCB-Rank* anticipates preferences and reduces excessive forces. While *FP* can finish faster by adhering to fixed priorities at the expense of comfort, *LinUCB-Rank* maintains comparable completion times while improving comfort. It also adheres more closely to varying user thresholds (Fig. 4, *Bottom*).

5.2 Hardware Testbed.

We deploy *PrioriTouch* in a challenging real-world 3D goal-reaching maze (Fig. 5), where vertical cylinders encode distinct body parts and preferences. The robot follows a predefined trajectory that necessarily induces whole-arm contact. When thresholds are exceeded, the robot retracts, updates the priority policy within a digital twin, and resumes with the refined policy, enabling steady progress toward the goal (see supplementary video on website).

5.3 User Studies

(a) ***Eliciting Population-Level Contact Preferences.*** We conducted a user study on robot-initiated contact in caregiving contexts (standing, bathing, dressing). Participants (n=98; ages 32–77, mean

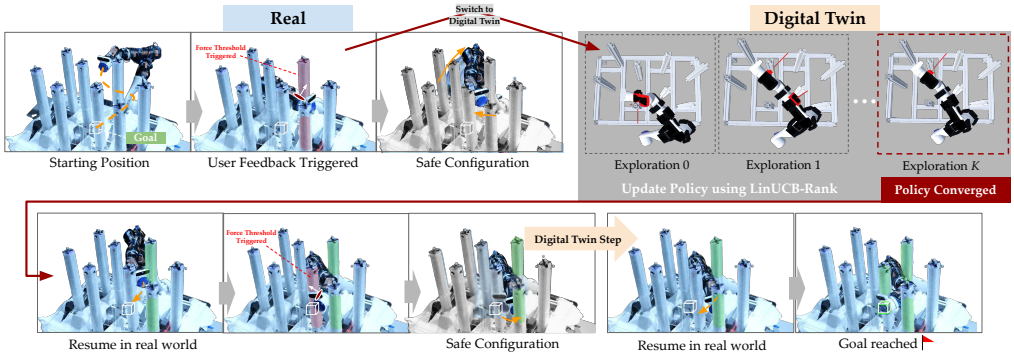

Figure 5: Real-world demonstration of PrioriTouch in a 3D goal-reaching maze. As the robot moves toward the goal, it periodically exceeds force thresholds (red cylinders), triggering user feedback. This feedback prompts LinUCB-Rank to update the priority policy within a digital twin. The robot then resumes with the refined policy, successfully progressing toward the goal.

58), recruited via Tetra Insights, included many with mobility limitations or disabilities. Each rated comfort for robot contact across 37 predefined body regions and three touch categories (functional assistance, emergency, sympathetic). Arms/hands and upper/lower back were most preferred for functional assistance, particularly among right-handed participants who needed support; sensitive regions (e.g., buttocks, genitals) were rarely selected. Fingers and toes were excluded because assistive tasks typically involve larger surface areas. See Appendix A.5 for details.

These findings directly inform our realistic user modeling approach, which includes: (1) a *base ranking* of contact openness by body part and (2) per-part *comfort force thresholds*. At run time, the robot enforces these thresholds to avoid exceeding comfort limits and uses the base ranking to resolve concurrent violations. We combine survey-derived preferences with biomechanical pain limits (see Table 8 in [5]) via $f^{\max}(b) = \gamma\, f^{\text{pain}}(b)\, S_r$, where $f^{\text{pain}}(b)$ is from [5], $\gamma = 0.7$ is a conservative base fraction, and S_r is the sensitivity ratio computed as the selection frequency for a body part divided by the maximum frequency across parts. This normalization assigns higher ratios to frequently selected regions, yielding a realistic basis for modeling contact preferences.

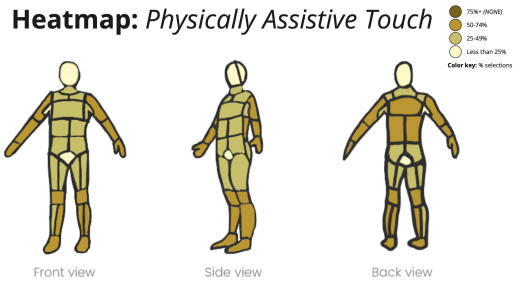

Figure 6: Outline of a human body divided into 37 regions; darker indicates greater touch acceptance, white indicates "no touch."

(b) *PrioriTouch Evaluation with Humans.* To validate the practical feasibility of our approach in caregiving scenarios, we conducted a user study with 8 participants (without visible mobility limitations) performing the robot-assisted bed-bathing task (Fig. 7). We use an RGBD camera (RealSense D455) and OpenPose to generate an aligned digital twin. In this evaluation, 7 out of 8 participants preferred our approach over the baseline (*HP*) for tasks involving WAM. Our method received higher ratings in perceived safety, comfort, and overall task performance (see Appendix A.6). Qualitative feedback was strongly positive, including comments such as: "*Both methods initially got stuck, but [ours] did pretty well on the second attempt*" (second attempt implies post policy update), and "*It performed the task very well, the contacts felt seamless and natural.*"

Figure 7: Experiment Setup: We use an overhead RGBD camera to perform pose estimation and spawn a digital twin for sim-in-the-loop preference learning.

6 Limitations

Although PrioriTouch provides a novel way of integrating user contact preferences into low-level control for whole-arm manipulation in pHRI, it has several limitations.

Tactile Sensing and digital twin fidelity. PrioriTouch benefits from distributed tactile sensing and a digital twin for simulation-in-the-loop learning. While our experiments indicate robustness to moderate modeling errors in collision modeling (see Appendix A.7), the use of more accurate tactile sensing and richer human avatars, obtained using real-time tactile and visual feedback [33, 34], would further improve sensitivity and sim-to-real transfer.

Handling ambiguous or noisy user feedback. Contact preference learning is currently driven by sparse real-time verbal feedback parsed into structured signals. Highly ambiguous early feedback can slow personalization; uncertainty-aware querying and complementary cues (e.g., visuo–tactile patterns) are promising extensions.

Access to reference path with whole-arm contact. We currently assume the robot has access to a feasible reference path that induces whole-arm contact, which is produced via teleoperation in our current workflow. Future work can explore contact-aware path planning, allowing *PrioriTouch* to support a broader range of assistive tasks (e.g., transferring, dressing) requiring more dexterous whole-arm maneuvers.

Real-world deployment. Lastly, we intend to deploy our method to assist individuals with mobility limitations. This will require unifying the improvements outlined above, including higher-fidelity sensing, better digital twins, and contact-aware planning, into a reliable whole-arm pHRI framework suitable for real-world use.

Acknowledgments

This work was partly funded by Toyota Research Institute (TRI), National Science Foundation IIS #2132846, and CAREER #2238792. This article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

[1] L. M. Hollinger and M. B. T. Buschmann. Factors influencing the perception of touch by elderly nursing home residents and their health caregivers. *International Journal of Nursing Studies*, 30(5):445–461, 1993.

[2] C. O'Lynn, A. Cooper, and L. Blackwell. Perceptions, experiences and preferences of patients receiving a clinician's touch during intimate care and procedures: a qualitative systematic review. *JBI Evidence Synthesis*, 15(11):2707–2722, 2017.

[3] E. De Luca, M. Wilson, M. R. Shaw, and T. T. Landis. "permission to touch": nurses' perspectives of interpersonal contact during patient care. *Western Journal of Nursing Research*, 44(5):456–465, 2022.

[4] T. L. Chen, C.-H. King, A. L. Thomaz, and C. C. Kemp. Touched by a robot: An investigation of subjective responses to robot-initiated touch. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 457–464, 2011.

[5] G. Pliske, M. Umbreit, S. Piatek, and F. Walcher. A statistical model to determine biomechanical limits for physically safe interactions with collaborative robots. *Frontiers in robotics and AI*, 8:667818, 2022.

[6] C.-H. King, T. L. Chen, A. Jain, and C. C. Kemp. Towards an assistive robot that autonomously performs bed baths for patient hygiene. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 319–324. IEEE, 2010.

[7] R. Madan, S. Valdez, D. Kim, S. Fang, L. Zhong, D. T. Virtue, and T. Bhattacharjee. Rabbit: A robot-assisted bed bathing system with multimodal perception and integrated compliance. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pages 472–481, 2024.

[8] F. Zhang, A. Cully, and Y. Demiris. Personalized robot-assisted dressing using user modeling in latent spaces. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3603–3610. IEEE, 2017.

[9] A. Kapusta, Z. Erickson, H. M. Clever, W. Yu, C. K. Liu, G. Turk, and C. C. Kemp. Personalized collaborative plans for robot-assisted dressing via optimization and simulation. *Autonomous Robots*, 43:2183–2207, 2019.

[10] R. Ye, S. Chen, Y. Yan, J. Yang, C. Ge, J. Barreiros, K. Tsui, T. Silver, and T. Bhattacharjee. Cart-mpc: Coordinating assistive devices for robot-assisted transferring with multi-agent model predictive control. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 667–676, 2025. doi:10.1109/HRI61500.2025.10974069.

[11] J. K. Salisbury. Whole-arm manipulation. In *Proc. of the 4th Int. Symp. of Robotics Research, Santa Cruz, CA*. Published by the MIT Press, Cambridge MA., 1987.

[12] L. Sentis and O. Khatib. Prioritized multi-objective dynamics and control of robots in human environments. In *4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, volume 2, pages 764–780. IEEE, 2004.

[13] B. Xu, L. Zhong, G. Zhang, X. Liang, D. Virtue, R. Madan, and T. Bhattacharjee. Cushsense: Soft, stretchable, and comfortable tactile-sensing skin for physical human-robot interaction. *arXiv preprint arXiv:2405.03155*, 2024.

[14] Z. Si, T. C. Yu, K. Morozov, J. McCann, and W. Yuan. Robotsweater: Scalable, generalizable, and customizable machine-knitted tactile skins for robots, 2023. URL `https://arxiv.org/abs/2303.02858`.

[15] A. Goncalves, N. Kuppuswamy, A. Beaulieu, A. Uttamchandani, K. M. Tsui, and A. Alspach. Punyo-1: Soft tactile-sensing upper-body robot for large object manipulation and physical human interaction. In *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, pages 844–851, 2022. doi:10.1109/RoboSoft54090.2022.9762117.

[16] Z. Liu, D. Chen, J. Ma, Z. Shen, T. Wu, Z. Jia, and Y. Jiang. A warm hug from a robot: A dual-mode e-skin with programming compliance. *Review of Scientific Instruments*, 93(11), 2022.

[17] P. Gliesche, C. Kowalski, M. Pfingsthorn, and A. Hein. Geometry-based two-contact inverse kinematic solution for whole arm manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8269–8274. IEEE, 2021.

[18] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp. Reaching in clutter with whole-arm tactile sensing. *The International Journal of Robotics Research*, 32(4):458–482, 2013.

[19] A. E. Block, S. Christen, R. Gassert, O. Hilliges, and K. J. Kuchenbecker. The six hug commandments: Design and evaluation of a human-sized hugging robot with visual and haptic perception. In *Proceedings of the 2021 ACM/IEEE international conference on human-robot interaction*, pages 380–388, 2021.

[20] J. Park and O. Khatib. Robot multiple contact control. *Robotica*, 26(5):667–677, 2008.

[21] M. Maroto-Gómez, S. M. Villarroya, M. Malfaz, Á. Castro-González, J. C. Castillo, and M. Á. Salichs. A preference learning system for the autonomous selection and personalization of entertainment activities during human-robot interaction. In *2022 IEEE International Conference on Development and Learning (ICDL)*, pages 343–348. IEEE, 2022.

[22] D. P. Losey and M. K. O'Malley. Including uncertainty when learning from human corrections. In *Conference on Robot Learning*, pages 123–132. PMLR, 2018.

[23] G. Canal, G. Alenyà, and C. Torras. Adapting robot task planning to user preferences: an assistive shoe dressing example. *Autonomous Robots*, 43(6):1343–1356, 2019.

[24] A. Narcomey, N. Tsoi, R. Desai, and M. Vázquez. Learning human preferences over robot behavior as soft planning constraints. *arXiv preprint arXiv:2403.19795*, 2024.

[25] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan. Learning from physical human corrections, one feature at a time. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 141–149, 2018.

[26] D. P. Losey and M. K. O'Malley. Learning the correct robot trajectory in real-time from physical human interactions. *ACM Transactions on Human-Robot Interaction (THRI)*, 9(1): 1–19, 2019.

[27] D. P. Losey, A. Bajcsy, M. K. O'Malley, and A. D. Dragan. Physical interaction as communication: Learning robot objectives online from human corrections. *The International Journal of Robotics Research*, 41(1):20–44, 2022.

[28] S. A. Mehta, F. Meng, A. Bajcsy, and D. P. Losey. Strol: Stabilized and robust online learning from humans. *IEEE Robotics and Automation Letters*, 2024.

[29] A. Bietti, A. Agarwal, and J. Langford. A contextual bandit bake-off. *Journal of Machine Learning Research*, 22(133):1–49, 2021.

[30] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[31] E. K. Gordon, X. Meng, T. Bhattacharjee, M. Barnes, and S. S. Srinivasa. Adaptive robot-assisted feeding: An online learning framework for acquiring previously unseen food items. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9659–9666. IEEE, 2020.

[32] R. Ye, W. Xu, H. Fu, R. K. Jenamani, V. Nguyen, C. Lu, K. Dimitropoulou, and T. Bhattacharjee. Rcareworld: A human-centric simulation world for caregiving robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 33–40. IEEE, 2022.

[33] A. Swann, M. Strong, W. K. Do, G. S. Camps, M. Schwager, and M. Kennedy. Touch-gs: Visual-tactile supervised 3d gaussian splatting. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10511–10518. IEEE, 2024.

[34] Z. Chu, F. Xiong, M. Liu, J. Zhang, M. Shao, Z. Sun, D. Wang, and M. Xu. Humanrig: Learning automatic rigging for humanoid character in a large scale dataset. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 304–313, 2025.

[35] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

A Appendix

A.1 Detailed Derivation of Hierarchical Operational Space Control (H-OSC)

H-OSC executes multiple objectives with a strict priority ordering. Consider an n-DOF robot manipulator with dynamics

$$\mathbf{M}(\mathbf{q}_t)\ddot{\mathbf{q}}_t + \mathbf{C}(\mathbf{q}_t, \dot{\mathbf{q}}_t) + \mathbf{g}(\mathbf{q}_t) = \boldsymbol{\tau}_t,$$

where $\mathbf{q}_t \in \mathbb{R}^n$ is the joint configuration, $\mathbf{M}(\mathbf{q}_t) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\mathbf{C}(\mathbf{q}_t, \dot{\mathbf{q}}_t) \in \mathbb{R}^n$ are Coriolis/centrifugal terms, and $\mathbf{g}(\mathbf{q}_t) \in \mathbb{R}^n$ are gravity torques.

Given a priority ranking $\sigma_t = (a_{t,1}, \ldots, a_{t,|\mathcal{J}_t|})$ over the active objectives $\mathcal{J}_t$, H-OSC computes torques by projecting lower-priority objectives into the null space of higher-priority ones:

$$\boldsymbol{\tau}_t = \sum_{j\in\mathcal{J}_t} \mathbf{N}_{j,t}\, \boldsymbol{\tau}_{j,t},$$

with each unprojected torque $\boldsymbol{\tau}_{j,t} = \bar{\mathbf{J}}_{j,t}^\top \mathbf{F}_{j,t}$, where $\bar{\mathbf{J}}_{j,t}$ is the null space projected Jacobian. The null space projectors are updated recursively as

$$\mathbf{N}_{j+1,t} = \mathbf{N}_{j,t} - \mathbf{J}_{j,t}^{\#}\bar{\mathbf{J}}_{j,t}\mathbf{N}_{j,t}, \quad \mathbf{N}_{1,t} = \mathbf{I}.$$

Here $\mathbf{J}_{j,t}$ is the Jacobian of objective j, and $\mathbf{J}_{j,t}^{\#}$ its dynamically consistent generalized inverse:

$$\mathbf{J}_{j,t}^{\#} = \mathbf{M}^{-1}\bar{\mathbf{J}}_{j,t}^\top \left(\bar{\mathbf{J}}_{j,t}\mathbf{M}^{-1}\bar{\mathbf{J}}_{j,t}^\top\right)^{-1}.$$

The operational-space dynamics for objective j at time t are

$$\boldsymbol{\Lambda}_{j,t}\ddot{\mathbf{x}}_{j,t} + \mu_{j,t} + \mathbf{g}_{j,t} = \mathbf{F}_{j,t},$$

where $\boldsymbol{\Lambda}_{j,t}$ is the operational-space inertia, $\mu_{j,t}$ the Coriolis/centrifugal effects, and $\mathbf{g}_{j,t}$ gravity terms.

Desired task accelerations $\ddot{\mathbf{x}}_{j,t}$ depend on the objective type:

- Pose-tracking objective: $J_{\mathbf{p}}(t)$: $\ddot{\mathbf{x}}_t = \mathbf{K}_p(\mathbf{p}_t^{\text{des}} - \mathbf{p}_t^{\text{curr}}) - \mathbf{K}_d\dot{\mathbf{p}}_t^{\text{curr}}$, where $\mathbf{K}_p, \mathbf{K}_d$ are pose gains.
- Force-regulation objective: $J_{\mathbf{f},b}(t)$ for body part $b \in \mathcal{B}_t$: $\ddot{\mathbf{x}}_t = -\mathbf{K}_f\left(f_t(b) - f^{\max}(b)\right)$, where $f_t(b)$ is the measured aggregate force and $f^{\max}(b)$ its comfort threshold.

Finally, operational-space forces $\mathbf{F}_{j,t}$ are mapped back into joint torques as $\boldsymbol{\tau}_{j,t} = \bar{\mathbf{J}}_{j,t}^\top \mathbf{F}_{j,t}$ and then combined using null space projectors to enforce the ranking σ_t.

A.2 Simulating User Feedback

We create a user model by initializing a base priority ID and comfort threshold for each body part. Given contact forces at timestep t, we compare the forces with the corresponding force thresholds. In case there is a body part for which the forces are above the threshold and the degree by which the forces are off. Using a sensitivity ratio (see Sec. 5.3), which is a function of the comfort threshold itself, we generate the feedback for this body part. In case of multiple violations, we use the base priority ID to decide which body part to generate feedback for. Feedback is generated for body parts with higher base priority unless the force violation is beyond a safety force limit (set manually and the same for each user).

A.3 Comparison of Feedback Mechanisms Using NASA-TLX

Our system currently solicits user feedback in a *descriptive* form, where users describe the affected body part and how they feel (e.g., "I feel slightly uncomfortable around my abdomen"). An alternative is *magnitude-based* feedback, in which users specify a numerical change in force (in Newtons). While magnitude-based feedback offers more precision, we hypothesize that most users prefer

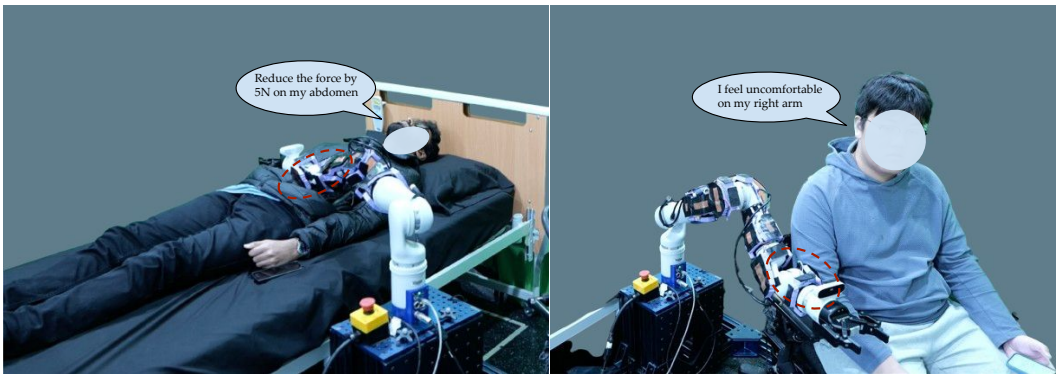

Figure 8: User study setup for comparing different feedback mechanisms in two different configurations. *Left* shows the participant lying down on the bed along with an example of the magnitude-based feedback, and *Right* shows the participant sitting on a wheelchair with an example of the descriptive feedback.

providing descriptive input due to the lower mental burden of recalling numeric force values. To investigate this, we conducted a Wizard-of-Oz study comparing these two feedback mechanisms via a modified NASA-TLX workload assessment.

Study Procedure.

We recruited 11 participants (8 male, 3 female; ages 21–28) without mobility limitations. Each participant experienced two configurations (see Figure 8): i) sitting in a wheelchair with the robot's forearm contacting their right arm, and ii) lying on a bed with the robot's forearm or upper arm contacting their lower abdomen. For each configuration, participants performed a separate trial under each of the two feedback mechanisms. At the end of every trial, they rated the method on a 5-point Likert scale for mental demand, hurriedness, and irritation. We repeated each configuration two times, counterbalancing the order of feedback mechanisms across participants. This study was approved by our organization's Institutional Review Board, and all participants provided written consent.

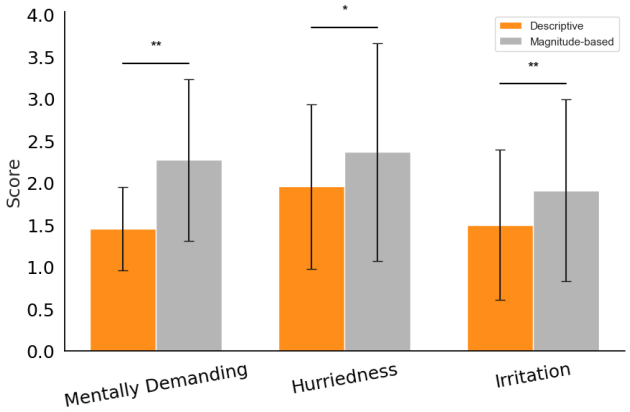

Figure 9: User study with 11 participants. Description-based feedback yields significantly lower cognitive workload (lower score is better) than magnitude-based controls.

Results and Analysis. Ten of the 11 participants preferred description-based feedback. Figure 9 shows that description-based feedback yielded significantly lower cognitive workload than the magnitude-based approach. A paired-sample *t*-test indicated statistically significant differences in mental demand ($p < 0.001$), hurriedness ($p < 0.05$), and irritation ($p < 0.001$) between the two methods. These findings support our hypothesis that descriptive feedback results in significantly lower mental burden, presenting a reasonable tradeoff between ease of feedback and precision of feedback.

A.4 Additional System Details

Sensors are modeled as soft bodies in Obi (Unity) and made using piezoresistive taxels (modified from [13]) in the real-world implementation. Each sensor provides contact location and force data. We integrate these sensor readings with H-OSC implemented using PyBullet [35]. H-OSC with LinUCB-Rank can operate at up to approximately 250 Hz, interfacing with a compliant low-level controller (1 kHz), ensuring robust handling of abrupt motions, such as muscle spasms. After stabilization, the system quickly refocuses on reference trajectories.

A.5 Additional Details of Contact Preference User Study

Study Procedure. Participants were recruited via Tetra Insights, a platform providing financially compensated, high-quality respondents. Our study was approved by our Institutional Review Board. Each participant was asked to envision a home-helper robot with a human-like form factor depicted through a hand-sketched illustration featuring inflated, soft materials. Participants indicated their comfort with robot-initiated touch on a graphical interface that included 37 predefined body regions, identified with input from two medical doctors (general practitioners). Participants also had the option to provide additional insights through open-ended responses.

Participant Demographics. The study included 98 adults (ages 32–77, mean age 58), of whom 42 participants were aged 65 or older and 56 were younger. Among participants, 67 identified as female (including 1 transgender female), and 31 identified as male. Forty-nine participants reported having a disability, 42 reported past injuries, and 7 chose not to disclose.

Detailed Findings. Overall, 40% of participants provided detailed open-ended responses elaborating on their choices. Arms and hands received the highest number of selections (779 total) across all touch categories. Right-handed participants specifically favored their dominant arm or hand for lighter supportive tasks, whereas the upper and lower back regions were predominantly selected for scenarios involving more substantial support or bracing. Regions such as the buttocks and genitals received minimal selections across all categories, highlighting privacy and comfort boundaries. Fingers and toes were deliberately excluded, as assistive interactions typically involve broader body regions.

A.6 Real-world user study with human subjects

Study Procedure. Participants first received an introduction describing the purpose of the study—evaluating two robotic arm control methods designed for robot-assisted caregiving involving whole-arm contact. Participants were informed about the specific caregiving scenario (bed bathing), their evaluation tasks (performance in clearing artificial dirt (ground coffee), perceived safety, and comfort), and how to provide verbal feedback during uncomfortable contacts. Before experimental trials, participants completed a supervised sample trial to practice providing feedback and simulating a person with mobility limitations. They used a provided digital questionnaire to record evaluations after each trial, identified via method IDs. In the actual trials, each participant experienced two distinct contact configurations across two methods, with two trajectories each (wiping arm and leg). Participants were repeatedly reminded to keep their responses independent of previous trials and to remain stationary during contact interactions. After completing all trials, participants filled out a final questionnaire before concluding the study.

Results. Below are the results from the study conducted with 8 participants with no visible mobility limitations.

Table 1: User–study ratings (mean ± SD on a 1–5 Likert scale).

	Task Perf.	**Safety**	**Comfort**
Heuristic	2.53 ± 0.94	4.59 ± 0.71	4.65 ± 0.61
LinUCB-Rank (Ours)	3.06 ± 0.97	4.65 ± 0.61	4.82 ± 0.39

In the post-study questionnaire, participants were asked to select which of the two methods they would prefer for caregiving applications involving whole-arm contact. Seven out of eight participants preferred our method.

A.7 Additional Experiments

Simulation-in-the-Loop Learning: Pauses and Accuracy.

Pauses: We measured execution times for preference learning convergence in simulation (52.4 ± 21.7s). These pauses reduced progressively (to $\sim$30s) as user-contact preference estimates improved, indicating suitability for online use.

Simulation Fidelity (Sim-to-Real Gap): To test robustness to simulation inaccuracies, we varied the fidelity of the digital twin's collision model (quality = 1: exact fit to visual mesh; quality = 0.75: 75% fit). Lower fidelity led to only a modest (12.21%) increase in required feedback signals, demonstrating our method's resilience.

Comparison with Alternative Baselines.

We evaluated PrioriTouch against two additional baselines:

- *Residual Reinforcement Learning:* Did not achieve stable convergence within our online-learning constraints, showing no force-violation reduction even after 20 minutes of training.
- *Jain et al.'s MPC-based whole-arm manipulation (IJRR 2013):* Adapted to our trajectories, this baseline resulted in roughly 3x more force-threshold violations than PrioriTouch.