MyoInteract: A Framework for Fast Prototyping of Biomechanical HCI Tasks using Reinforcement Learning

Ankit Bhattarai*
University of Cambridge
Cambridge, United Kingdom
ab2731@cam.ac.uk

Hannah Selder*
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)
Dresden/Leipzig, Leipzig University
Leipzig, Germany
hannah.selder@uni-leipzig.de

Florian Fischer*
University of Cambridge
Cambridge, United Kingdom
fjf33@cam.ac.uk

Arthur Fleig†
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)
Dresden/Leipzig, Leipzig University
Leipzig, Germany
arthur.fleig@uni-leipzig.de

Per Ola Kristensson†
University of Cambridge
Cambridge, United Kingdom
pok21@cam.ac.uk

Abstract

Reinforcement learning (RL)-based biomechanical simulations have the potential to revolutionise HCI research and interaction design, but currently lack usability and interpretability. Using the Human Action Cycle as a design lens, we identify key limitations of biomechanical RL frameworks and develop MyoInteract, a novel framework for fast prototyping of biomechanical HCI tasks. MyoInteract allows designers to setup tasks, user models, and training parameters from an easy-to-use GUI within minutes. It trains and evaluates muscle-actuated simulated users within minutes, reducing training times by up to 98%. A workshop study with 12 interaction designers revealed that MyoInteract allowed novices in biomechanical RL to successfully setup, train, and assess goal-directed user movements within a single session. By transforming biomechanical RL from a days-long expert task into an accessible hour-long workflow, this work significantly lowers barriers to entry and accelerates iteration cycles in HCI biomechanics research.

CCS Concepts

• **Human-centered computing** → **User models**; **Systems and tools for interaction design**.

Keywords

deep reinforcement learning, biomechanical models, simulated users

1 Introduction

Simulations are becoming central to HCI, allowing researchers to ask controlled "what-if" questions, test designs at scale for diverse user groups, and probe our understanding of interactive systems. As Murray-Smith et al. [39] argue, the ability to match user behaviour with a generative model is a strong test of understanding, making design and engineering more predictable. *Biomechanical Reinforcement Learning* (RL), which trains simulated users to control realistic musculoskeletal models, is a particularly promising direction for producing such generative models.

By simulating interaction at the level of muscles and body dynamics, biomechanical RL can predict motion trajectories and physical effort for complex tasks [4, 7, 24, 41]. Previous work has validated that these simulated users reproduce established motor patterns, such as the bell-shaped velocity profiles of aimed reaching [16, 36]. Crucially for HCI, synthesised trajectories align with Fitts' Law [16, 24, 36] and the Two-Thirds Power Law [16, 26], suggesting that simulated users can reliably predict plausible aimed body movements. Consequently, researchers have begun applying these models to button pressing [24], keyboard typing [22], joysticks [24], smartphone use [34], and virtual pointing techniques [25, 37], as well as using them to guide the ergonomic design of VR/XR interfaces [14, 18, 35] and adaptive systems [29, 30, 46]. In the future, such simulations could enable early-stage prototyping—testing layouts, comparing movement strategies, or exploring accessibility for users with diverse motor abilities—without the cost of human studies. Some visionary examples include a replicated prototype study for a VR game using simulated users [18], and Moon et al. [36], who have been able to train a state-of-the-art target inference classifier solely from simulation data.

Despite these promising advances, biomechanical RL currently remains a specialised technique restricted to experts, rather than a practical prototyping tool for the wider HCI community. The expertise required to configure a simulation is prohibitive as setting up a single experiment involves writing model and task specifications, scripting in Python, and designing custom reward functions. Furthermore, the feedback loop is excruciatingly slow: training a policy typically takes 12 hours to several days [18, 24, 36, 54], with little visibility during the training process into whether the configuration is succeeding. If the training produces implausible behaviour, diagnosis means restarting the training.

Viewed through Norman's action cycle [42], prototyping biomechanical HCI tasks suffers from severe *Gulfs of Execution and Evaluation* (see Figure 2). Consider a designer exploring widget placement

*Equal first author contribution.
†Equal last author contribution.

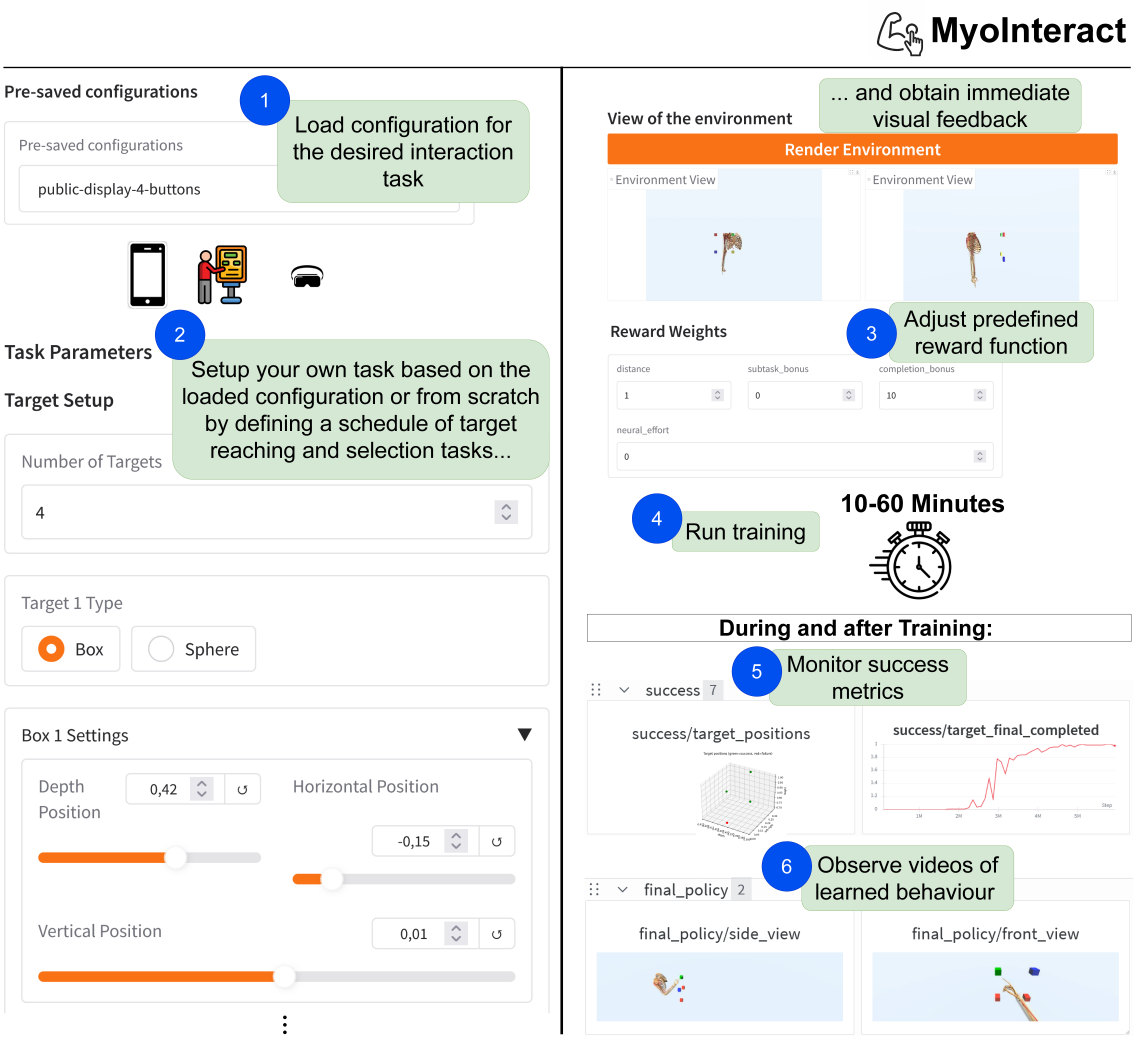


Figure 1: The GUI of *MyoInteract*, a framework that allows HCI researchers to setup, train, monitor, and evaluate biomechanical simulations of interactive user behaviour. Users can (1) choose from a list of pre-defined task configurations for different HCI contexts (Mobile Touch, Public Display, Augmented Reality) and (2) adjust them, or define entirely new interaction tasks that involve sequences of target acquisition and selection. They can double-check the rendered interaction environment and (3) adjust the proposed reward function. With *MyoInteract*, training (4) takes less than one hour (previously: 12–48 hours) and (5) can be continuously monitored via success metrics. (6) Videos enable qualitative inspection of learned behaviour. Advanced mode (not shown) offers the option of setting additional parameters relating to task setup, the biomechanical model, and the RL training procedure.

in Mixed Reality to minimise physical effort. Testing different configurations *in silico* faces a *Gulf of Execution*: the designer cannot simply place a target; they need to modify XML files and Python scripts to define the task, adjust RL parameters to ensure training succeeds, and understand low-level details of the simulation pipeline [24]. Once training begins, the process becomes a black box, creating a *Gulf of Evaluation*. The system offers no diagnostic insight into why a simulated user behaves in a certain way, or how this behaviour emerged. Currently, no systematic diagnosis and analysis tools exist to assess and compare the infinite number of possible design choices. Combined with day-long training times, these gulfs render iterative prototyping impossible. As long as setting up a simulation takes longer than running a human study, biomechanical RL cannot serve its purpose as a design tool. This paper asks: *What if these barriers could be collapsed, transforming biomechanical RL from a specialised research capability into an accessible, practical instrument for HCI prototyping?*

We present *MyoInteract*, a framework designed to address three design goals: reducing the gulf of execution, reducing the gulf of evaluation, and collapsing the temporal barrier of biomechanical user simulations. *MyoInteract* leverages GPU-accelerated physics to reduce training times from days to minutes, while preserving

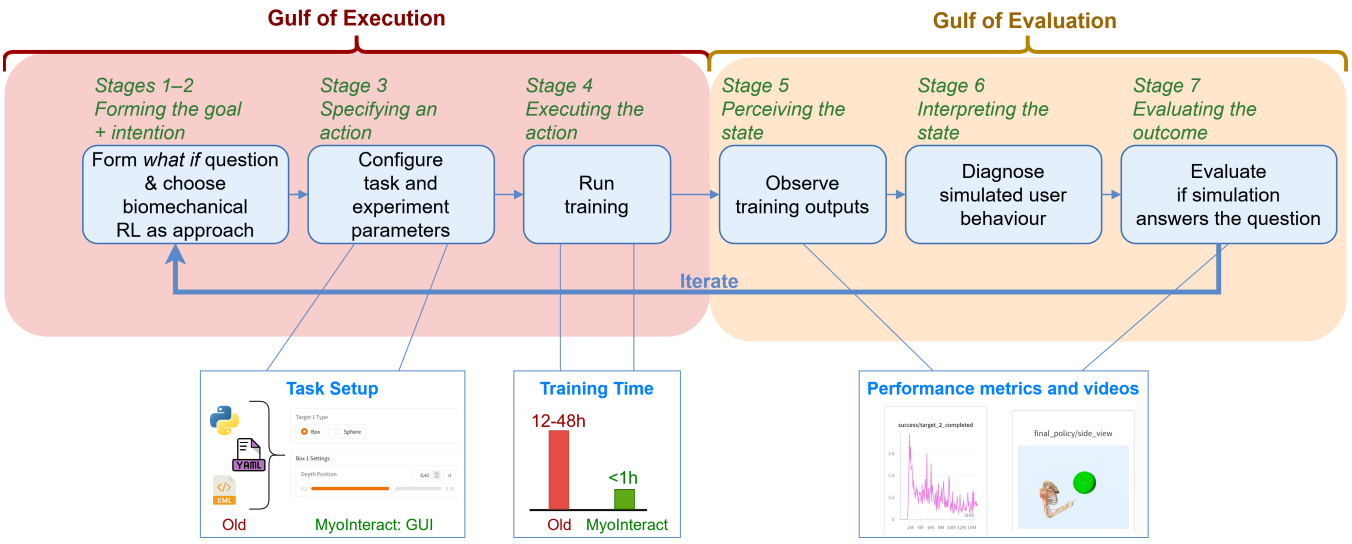


Figure 2: Workflow of the biomechanical RL simulations approach analysed using Norman's seven stages of action [42], illustrating barriers in both the Gulf of Execution (task specification) and Gulf of Evaluation (training feedback). The bottom panels show how MyoInteract addresses each barrier: a GUI replaces manual XML/Python/YAML editing, GPU acceleration compresses training from 12–48 hours to under one hour, and a suite of performance metrics and visualisations replaces the minimal outputs of prior systems (which provided only aggregate metrics such as mean reward and episode length). These features significantly shorten the action cycle, transforming biomechanical RL simulation from a days-long expert endeavour into an accessible, hour-scale workflow.

biomechanically plausible movement patterns. We combine this speed with a graphical interface (Figure 1) that allows users to configure tasks and parameters without extensive coding or RL expertise, and a real-time logging and visualisation suite that provides immediate and comprehensive insights into learned user behaviour. By reducing the feedback loop time by up to 98%, *MyoInteract* for the first time shifts biomechanical simulation into an interactive, iterative design activity.

We demonstrate this framework on sequential target acquisition and selection tasks, for which biomechanical RL simulations have been extensively validated [16, 18, 24, 34, 36], and report on a hands-on workshop with 12 HCI researchers and practitioners. We find that by collapsing the time and expertise barriers, *MyoInteract* transforms a days-long expert endeavour into an accessible, hour-scale workflow. This work **contributes**:

(1) ***MyoInteract***, a framework for fast prototyping of biomechanical HCI tasks that enables the configuration, simulation, and evaluation of interaction movements in less than an hour. The open-source code is available at https://github.com/MyoInteract/MyoInteract.
(2) **Findings** from a workshop with 12 HCI researchers, yielding empirical insights that the framework allows biomechanical RL novices to successfully deploy biomechanical user simulations, and providing suggestions for designing biomechanical user simulation tools.

The remainder of the paper situates this work in related literature (Section 2), illustrates our design rationale (Section 3), describes the framework and interface (Section 4), shows demonstrations and performance analyses (Section 5), presents findings from the workshop evaluation (Section 6), and concludes with a reflection on the opportunities and limitations of biomechanical simulations for HCI and tangible next steps to increase the validity and scope of computational user simulations (Section 7).

2 Related Work

The idea of simulating users as embodied agents has gained traction in and beyond HCI. Our work builds on the progression from inverse to forward biomechanical models in HCI, the technical foundations that enabled this shift, and the emerging need for accessible computational design tools.

2.1 From Inverse to Forward Biomechanical Models in HCI

Within HCI, research on biomechanical simulation has historically relied on *inverse models*, where muscle and joint forces are inferred from recorded motion to estimate fatigue and muscle utilisation [3, 4]. While useful for analysing existing movements, inverse models cannot predict how users would interact with novel interface designs.

Recently, advances in physics engines like MuJoCo [61] and specialised toolkits such as MyoSuite [8] and Hyfydy [52] have enabled a shift toward *forward models* that generate behaviour directly from muscle activation signals [22, 24, 25, 36]. This shift is critical: forward approaches allow researchers to evaluate interaction techniques and estimate physical effort *before* running user studies. However, forward models require a *controller* to determine muscle

activations of the musculoskeletal system on a moment-to-moment basis, a complex high-dimensional control problem.

Reinforcement Learning has emerged as the dominant solution for this control problem. Fischer et al. [16] demonstrated how RL could drive a torque-actuated human arm model to accomplish goal-directed movements, with trajectories that matched established motor control principles. Ikkala et al. [24] introduced *User-in-the-Box* (UitB), which combined muscle-level dynamics with perceptual feedback to simulate reaching and pointing behaviour. Alternative control approaches, such as Model Predictive Control [25], have also been explored, though RL has proven more scalable for complex, multi-goal tasks. These foundational works have since been extended to VR ergonomics [18] and goal inference [36], demonstrating the potential of biomechanical RL for analysing and improving interactive systems.

Beyond HCI, biomechanical models have been used to predict human movements and ergonomic states from muscle control signals [2, 33]. Biomechanical RL specifically has been adopted by the rehabilitation [13, 23, 43], neuroscience [7, 11, 52, 58], and robotics communities [45, 48] for movement prediction, prosthesis control, and motor learning research. This widespread adoption across applications confirms RL as a state-of-the-art method for generating plausible muscle-driven behaviour.

2.2 Complexity as a Barrier to Adoption

While the potential of biomechanical RL is clear, its adoption in HCI is hindered by its complexity. A critical review of the literature reveals that virtually no HCI studies have successfully employed frameworks like *UitB* or *MyoSuite* "out of the box". Instead, successful implementation typically requires (extensive) modification of reward functions, biomechanical models, and training procedures. Miki et al. [35] notably reported "poor generalizability of the [UitB] simulator", underscoring that even minor changes, such as altering arm length, require tedious manual parameter tuning and retraining.

Consequently, a significant portion of recent research has been dedicated solely to making these simulations work reliably. Selder et al. [54, 55] investigated the fragility of reward structures, deriving design guidelines to prevent simulation failures. Similarly, specific learning curricula are required to successfully train simulated users for dexterous tasks [34]. Further work has explored how motor constraints [10] and noise [12] influence learning efficiency. These works illustrate that biomechanical RL is currently less of a design tool and more of a technical research challenge.

2.3 The Wider Landscape of User Simulation

Parallel to biomechanics, a wider movement in HCI explores how simulation can model user behaviour more generally. Frameworks based on Computational Rationality [31, 44] and Active Inference [40] use optimisation to explain how users balance cognitive and motor costs. RL has also been employed to model adaptive decision-making in non-biomechanical contexts, such as touchscreen typing [57] and adaptive interface control [27]. More recently, Large Multimodal Models have been used as agentic user proxies [66] or to infer preferences from unstructured data [56].

These developments have broadened the scope of user simulation from the physical to the cognitive and social levels. However, these cognitive and social intents must ultimately be mediated by the physical body. Currently, we lack a framework for the *physical body* that supports *iterative* design, i.e., rapidly exploring how constraints like fatigue and effort shape interaction, rather than waiting days for a single analysis. What is missing is the speed and usability required to transform biomechanical simulation from a technical research method into a practical prototyping instrument. *MyoInteract* addresses this gap by providing GPU-accelerated training, real-time visual diagnostics, and a graphical interface that lets users configure and run biomechanical RL simulations without deep technical expertise.

3 Design Goals

Our design process was guided by the goal of making biomechanical simulation accessible to HCI researchers without sacrificing the fidelity required for valid user modelling. Using Norman's Action Cycle [42] as a diagnostic lens, we derived three aspirational **design goals** that shaped our development of *MyoInteract*:

- DG1 **Reduce the Gulf of Execution:** Eliminate the need for XML/YAML/Python coding by raising the level of abstraction, allowing researchers to specify *what* they want to test rather than *how* to implement it.
- DG2 **Reduce the Gulf of Evaluation:** Make the opaque process of RL training more observable by surfacing diagnostic information in real-time, helping users distinguish task design issues from configuration errors and training failures.
- DG3 **Collapse the Temporal Barrier:** Compress the long training times that compound both gulfs to enable iterative exploration within a single working session.

While our current implementation focuses on sequential pointing and selection tasks, the principles we derive are intended to generalise to broader biomechanical simulation workflows.

3.1 Design Principles and System Evolution

We developed *MyoInteract* iteratively, guided by six principles that emerged as we moved from internal prototypes to the final system. These principles address the barriers identified above, scoped to the domain of upper-body pointing and selection tasks.

DP1: Enable Rapid Iteration through Speed (→ DG3). For simulation to support design exploration, feedback loops must be short enough to permit multiple iterations within a session. In prior work, researchers had to commit to a single configuration and wait hours or days to assess viability. Drawing on recent advances in GPU-accelerated RL environments [60, 65], we prioritised reducing training time not merely for efficiency, but to fundamentally change the character of the workflow from batch analysis to interactive exploration. By compressing training to minutes (Section 4.1), users can test alternative designs, reward functions, and task variants iteratively.

DP2: Enable Composability through Task Decomposition (→ DG1). In prior frameworks like [24] and our early prototypes, each interaction scenario was implemented as a monolithic unit, requiring new

XML files and custom Python logic. This high "viscosity" [20] discouraged experimentation with complex sequences. We addressed this by decomposing tasks into sequential combinations of pointing and button-pressing primitives. Users now specify tasks through high-level configuration instead of code, and the system automatically generates the corresponding simulation logic (Section 4.2.2). This abstraction reduces setup time and lowers the expertise barrier.

DP3: Make Configuration Visible and Manipulable (→ DG1). While configuration files reduce code changes, they are prone to silent errors: users may be unsure which parameters are valid, how they affect behaviour, or which defaults apply without consulting documentation. Guided by Norman's principle of Visibility [42], we replaced text-based configuration with a graphical interface that exposes available parameters, displays current and default values, and enforces valid ranges through interactive controls (Section 4.3). Users can also preview target layouts spatially before training, reducing the likelihood of misconfigured tasks.

DP4: Provide Multi-Level Feedback for Diagnosis (→ DG2). RL failures are notoriously opaque: a success rate that plateaus at 70% may reflect unreachable targets, misspecified rewards or insufficient training. To support diagnosis, we provide feedback at multiple stages. Pre-training, targets are specified using numerical parameters that define their 3D position. To visualise how these parameters affect the spatial layout, we augment the GUI with a rendering feature that allows users to preview the task environment. During training, real-time dashboards (rather than upon completion) decompose aggregate metrics (e.g., per-target success rates, individual reward components). Originally developed as internal debugging tools, these proved essential for identifying bottlenecks (e.g., "the fourth target in a sequence is unreachable") that aggregate scores obscure (Section 4.4). Post-training, automatically generated videos reveal qualitative issues such as jittery motion or unnatural postures, which numerical metrics alone cannot easily surface.

DP5: Encode Domain Constraints to Prevent Errors (→ DG1). Biomechanical RL configurations are fragile; small parameter errors can invalidate results. Through iterative development enabled by faster training cycles, we identified common failure modes and encoded safeguards into the interface. For example, sliders enforce valid parameter ranges, target placements are constrained to plausible workspace regions, and the system recommends training durations based on task complexity. These constraints reduce the likelihood of wasted runs, particularly for novice users.

DP6: Support Progressive Disclosure (→ DG1). Early GUI iterations exposed all parameters simultaneously, overwhelming users, especially novices, who struggled to distinguish essential settings from advanced options. Inspired by progressive disclosure principles [32], we split the interface into a "Simple Mode" for task setup and an "Advanced Mode" for precise control. This structure supports a natural workflow: users begin with coarse task design and reveal additional control only when needed, balancing accessibility with expert capability.

4 MyoInteract: A GPU-accelerated Interaction Simulation Framework

In this section, we describe how the design principles outlined in Section 3 are realised in *MyoInteract*, a GPU-accelerated framework for biomechanical user simulation. Our design choices prioritise speed, composability, and observability.

4.1 System Architecture: Speed through Parallelization (DP1)

To enable rapid iteration (DP1), we built *MyoInteract* on MuJoCo-MJX[1], a GPU-accelerated reimplementation of the MuJoCo physics engine [61] in JAX [6]. Unlike CPU-based simulators, MJX enables thousands of parallel environments on GPUs/TPUs, dramatically increasing training speed. Policy learning uses PPO [51] implemented in Brax [19] (hyperparameters in Appendix B), enabling end-to-end GPU execution without CPU-GPU transfer bottlenecks.

While recent work has leveraged MJX for humanoid locomotion [60, 65] and pose estimation [15], to our knowledge *MyoInteract* is the first framework to apply GPU-accelerated biomechanical simulation specifically to interactive HCI tasks. This architectural choice is the technical foundation for DP1, providing the computational throughput necessary to shift the workflow toward interactive exploration.

The framework integrates with the open-source MyoSuite library [8], ensuring compatibility with a broad range of musculoskeletal models [63, 64] and emerging extensions such as fatigue models, assistive devices [59], and sample-efficient RL methods [12, 53].

4.2 Composable Interaction Modelling (DP2)

To realise the goal of task composability (DP2), *MyoInteract* provides a unified environment where users can configure the biomechanical model, the task, and the information flow between them (observations) without low-level coding. In the following, we describe these components conceptually, while implementation details, including parameter values, are provided in the appendix.

4.2.1 Biomechanical Model. *MyoInteract* is designed to support a range of musculoskeletal models. As the default for pointing tasks, we employ an adaptation of the MoBL-ARMS model [50], the same high-fidelity representation of the upper extremity with 26 muscles and fixed fingers provided in the UitB framework [24]. However, users can toggle between alternative models via the GUI, including the MyoArm[2] and a full-hand extension of the MoBL-ARMS model.

To capture the motor variability underlying the speed-accuracy trade-off [21, 62], we implement both signal-dependent and constant motor noise. Given the ongoing debate regarding the optimal noise structure for simulation realism [10, 16], we expose these parameters as configurable options in the GUI, allowing researchers to empirically test their effect on plausibility. Finally, to ensure robust policies that generalise across postures rather than overfitting to a single starting position, the system randomises initial joint states and muscle activations at the start of each episode.

[1] https://mujoco.readthedocs.io/en/stable/mjx.html
[2] https://github.com/MyoHub/myo_sim

4.2.2 Task Composability. We decompose interaction into two **composable primitives** (DP2) that form the foundational building blocks for the majority of mid-air HCI tasks:

- **Pointing:** The user must move the end-effector (fingertip) into a target sphere and maintain it for a specified dwell time. Targets are parametrised by size, 3D location, and colour.
- **Pressing:** The user must apply a specific activation force to a surface. Buttons are parametrised by location, orientation, geometry, colour, and force threshold.

The power of *MyoInteract* lies in **chaining** these primitives. Instead of implementing monolithic environments for every new task, researchers can define complex workflows as sequences of primitives – without modifying a single line of XML or Python code. For example, an AR interface can be defined as a sequence alternating between reaching virtual spheres (with 0.5s dwell) and pressing physical buttons (with 2N force). This compositional approach allows for the instantiation of a wide variety of HCI scenarios (Section 4.5). By decoupling task logic from implementation, researchers can focus on the *design* of the interaction sequence rather than the engineering of the simulation.

4.2.3 Observations. To control the biomechanical model, the simulated user relies on a state vector combining proprioception (joint angles, velocities, accelerations, muscle activations, fingertip position) and task information (target position/size, sequence progress, dwell timer). Crucially for design exploration, this observation space is configurable via the GUI (DP3). Users can selectively exclude specific signals to model different user capabilities or constraints, such as masking muscle states to model proprioceptive deficits.

4.2.4 Reward function. The agent's behaviour is shaped by a reward function composed of four weighted terms, where $(\cdot)$ is a placeholder for all relevant function arguments included in the current system state and muscle control vector:

- $f_{\text{distance}}(\cdot)$: a **distance term** representing the sum of the distances between the fingertip and the current target, as well as between the remaining targets;
- $f_{\text{subtask\_bonus}}(\cdot)$: a **subtask bonus** granted when the current target is successfully reached for the first time;
- $f_{\text{completion\_bonus}}(\cdot)$: a **task completion bonus** awarded upon successful completion of all subtasks;
- $f_{\text{effort}}(\cdot)$: an **effort penalty** that discourages biomechanically implausible or overly strenuous movements. This is achieved by penalising the squared muscle control values (*neural effort* costs [5, 24, 54]); however, the framework allows replacing this component with alternative effort models.

These components are derived from established guidelines [54], and their relative weights determine the emergent movement strategy (e.g., prioritising speed vs. effort). In *MyoInteract*, these weights are fully exposed in the GUI (DP3), enabling designers to interactively tune this trade-off.

To further simplify training, the system provides suggestions for training duration based on task complexity (e.g., adding 1M steps per additional target or additional 0.3 seconds of total dwell time). We iteratively found this heuristic yet sensible baseline to prevent incomplete training runs (supporting DP5).

4.3 Interface Design: Visibility, Progressive Disclosure, and Guidance (DP3, DP5, DP6)

To bridge the gap between high-level interaction design and low-level RL configuration, we developed a graphical interface that replaces manual scripting with direct manipulation. Built on Gradio [1], the GUI operationalises three key design principles: Visibility (DP3), by exposing parameters; Progressive Disclosure (DP6), by layering complexity; and Heuristic Guidance (DP5), by actively assisting with parameter and training decisions.

Upon startup, the interface presents users with a direct link to the logging dashboard (Section 4.4). Default parameter values and ranges are provided to guide users and reduce the need for parameter tuning (DP5). Configurations can be saved and reloaded to facilitate reuse and reproducibility. To scaffold the learning curve, we structure the interface into two distinct interaction modes:

Simple Mode (Figure 10). Designed for rapid prototyping, this mode exposes the essential parameters for defining interaction logic. Users configure primitives (e.g., spheres, boxes) through direct manipulation controls, adjusting properties such as position, size, and force thresholds via sliders with visual ranges such as in Figure 3. A built-in 3D renderer allows users to visually inspect the task environment from lateral and frontal perspectives before training begins, supporting DP4 pre-training.

Crucially, Simple Mode also exposes the reward structure. As established in Section 4.2.2, reward weights dictate the agent's emergent strategy. We do not hide this complexity, but make it transparent: adjusting a weight immediately updates the displayed reward equation, visualising the mathematical impact of the design choice (DP3). To further aid intuition, tooltips display the theoretical minimum and maximum values for each component based on the current task setup, helping users better calibrate competing objectives, such as speed versus effort.

Advanced Mode (Figure 11). Targeting more experienced users, this mode reveals deeper configuration layers (supporting DP6). It provides granular control over three critical dimensions:

- **Biomechanical Fidelity:** Users can adjust control time steps (i.e., the frequency with which the simulated user updates their muscle control), toggle signal-dependent and constant motor noise, and whether and how to reset the musculoskeletal model at the end of each episode (e.g., a fixed or random initial pose).
- **Observation Space:** Users can customise the agent's sensory inputs, selectively enabling or disabling specific proprioceptive signals to model user constraints (e.g., specific sensory deficits).
- **Training Dynamics:** To prevent incomplete training runs, i.e., too few training steps, the system implements Heuristic Guidance and Domain Constraints (DP5). It automatically calculates a recommended training duration based on task complexity (scaling with target count and dwell time). While novices can rely on this baseline to ensure convergence, experts retain the ability to override it, fine-tune RL

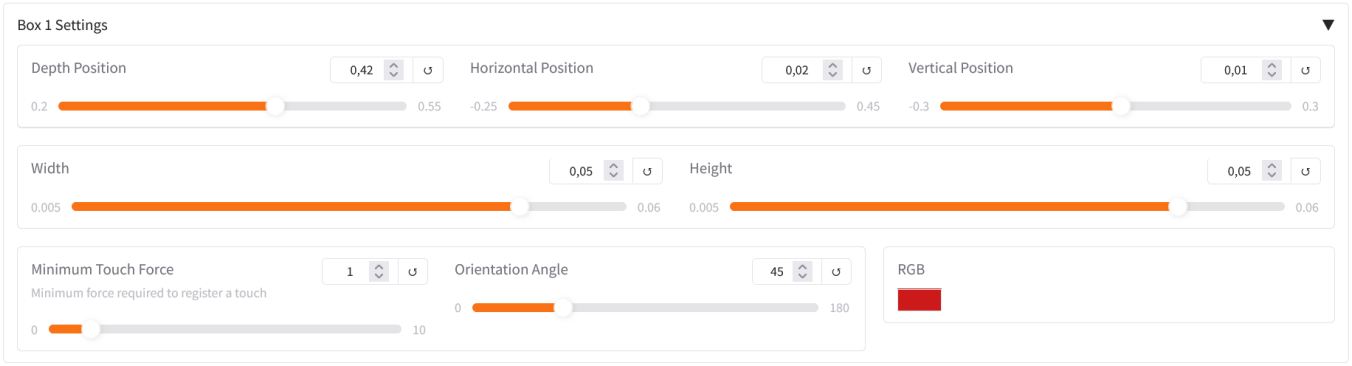


Figure 3: Box target settings in our GUI. Users can adjust position, size, minimum touch force, and orientation angle via sliders, and choose the object colour from a palette.

hyperparameters (e.g., batch size, number of parallel environments), and manage the frequencies to save the current policy (checkpoint frequency) and to evaluate it (evaluation frequency). Users can also load pre-trained policies to continue training and specify random seeds for reproducibility.

This layered approach ensures that while the system remains accessible to HCI researchers without biomechanical RL expertise, it retains the depth required for rigorous computational evaluation.

4.4 Metrics and Logging (DP4)

One of the primary challenges in applying RL to HCI is the “black box" nature of training. To address this, *MyoInteract* implements Observability (DP4) by recording a comprehensive suite of metrics designed to facilitate not just monitoring, but active debugging of policy behaviour. All metrics are automatically synchronised with Weights & Biases (WandB)[3], providing a centralised dashboard for real-time tracking.

Granular Performance Metrics. Standard RL metrics (e.g., total reward) are often insufficient for diagnosing failure in complex movement sequences. Therefore, we decompose performance into granular units:

- **Sequential Subtask Completion:** Beyond overall success rates, we log completion rates for each individual subtask (Figure 4). This provides direct insights into which parts of the tasks are difficult to achieve.
- **Spatial Error Analysis:** To identify biomechanical "blind spots", the system generates 3D visualisation showing the success rate for each of the target locations, as well as 2D plots correlating performance with target size. This enables the immediate identification of unreachable workspace areas or precision deficits.
- **Terminal Distance Error:** We log the remaining distance to targets at episode termination, distinguishing between “near misses" and complete tracking failures.

[3]WandB (https://wandb.ai/) is an online machine learning development platform that allows researchers to manage, track, and visualise their training in real time.

Reward Metrics. To make the agent’s training incentives more transparent, we log the individual contributions of each reward component (distance, completion bonus, subtask bonus, and effort) alongside the total accumulated reward (Figure 5). This decomposition is critical for closing the design loop: it allows researchers to see if an agent is ignoring the task to minimise effort (effort dominance) or moving erratically to maximise speed (distance dominance), facilitating informed tuning of the weights exposed in the GUI (Section 4.3).

By default, both performance and reward metrics are directly calculated from the training rollouts used to update the policy, i.e., they display the average value among all episodes generated since the last policy update step. This enables an efficient and direct inference of the current training performance, without additional evaluation overhead. To enable a more rigorous, comparable, and reproducible assessment of an (intermediate) policy’s performance and quality, we additionally enable adding regular **evaluations** at a user-defined frequency (e.g., every 2M steps) during the training.

Finally, for qualitative assessment of learned behaviour, we include two video renderings of the trained policy from different camera perspectives in the **final policy** section. Videos of intermediate training policies can also be generated from evaluation runs. These qualitative samples provide the necessary context to interpret the quantitative data, ensuring the learned behaviours are not only successful, but also biomechanically plausible. Since this comes at the expense of increased training time, it is disabled by default.

4.5 Example Scenarios

To demonstrate the expressivity of *MyoInteract’s* compositional primitives (DP2), we implemented three diverse HCI scenarios: *AR Interaction*, vertical *Public Display Interaction*, and horizontal *Mobile Keyboard Typing*. We also provide a walk-through on how the GUI can be used to create the *AR Interaction* task. The exact implementation details are provided in the appendix.

4.5.1 AR Interaction. Consider a designer prototyping an Augmented Reality interface where users alternate between virtual UI elements and physical controls—a common pattern in mixed reality workstations. How does interleaving physical and virtual

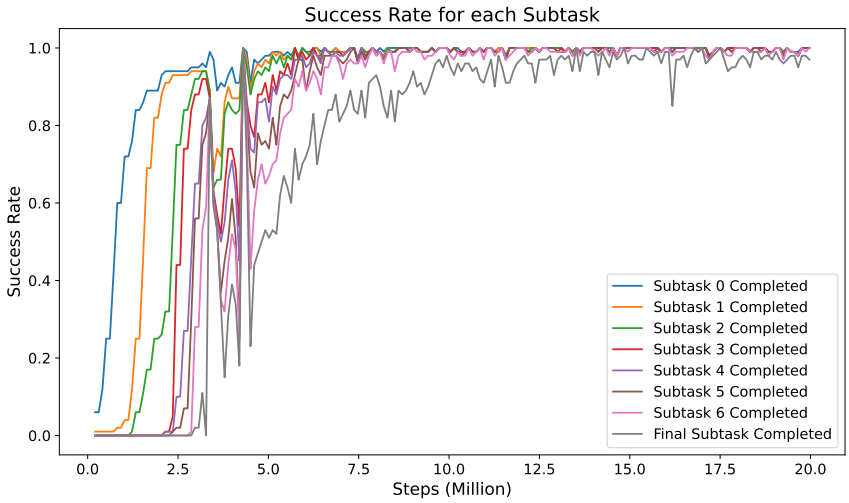


Figure 4: Analysis of subtask completion metrics for each target in the AR task. Note how the success rates of subtasks build upon each other sequentially.

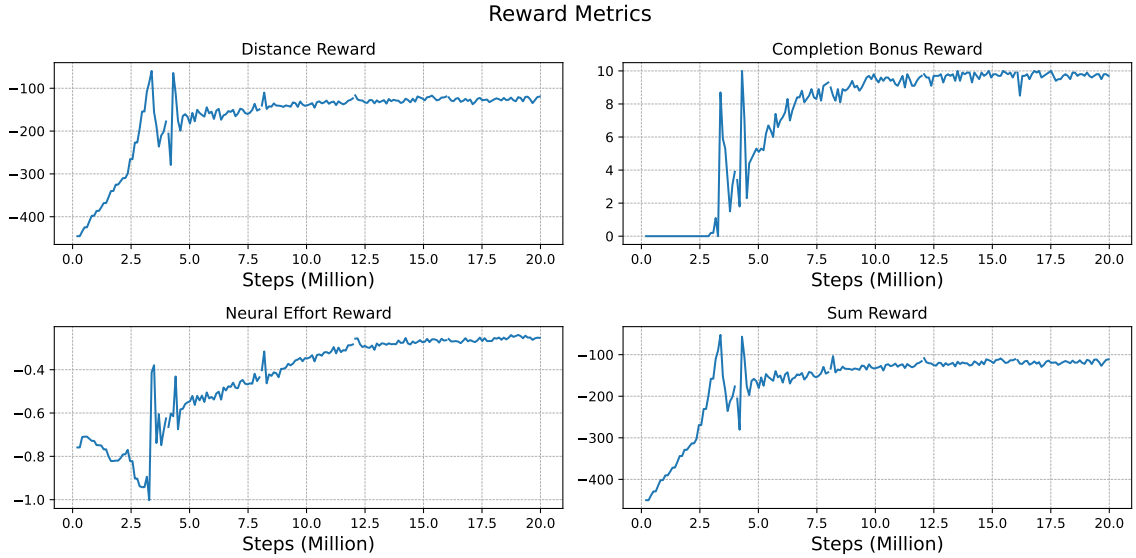


Figure 5: Analysis of various reward components across training for the AR task. In this example, the individual reward components converge after 10-14 million steps, respectively.

targets affect movement strategies? Figure 6 shows an example *AR task* constructed in *MyoInteract*, consisting of four virtual pointing targets and four physical buttons. The following is a quick walk-through of how users would use the GUI to create this task and train a RL agent:

Using the Simple Mode, the user decides the number and types of targets to be 8 (4 physical buttons and 4 virtual spheres), along with task-relevant parameters such as button positions and spawn regions for virtual spheres (step 2 in Figure 1). Before training, the environment is previewed to confirm that the sub-task targets are situated in reasonable locations, i.e., not beyond the limits of the arm or in difficult to reach positions such as behind the user. Reward weights can then be adjusted, for example to emphasise effort minimisation or task speed (step 3 in Figure 1). Training is launched (step 4 in Figure 1) and monitored in real-time using sub-task metrics and reward component plots (step 5 in Figure 1). After 2 million steps, these logs show (see Figure 4) that the first two targets are reliably reached, while learning is still ongoing for the third, indicating that the current configuration is suitable and training can continue. Conversely, flat reward curves early in training signal misconfigured tasks or rewards, prompting early termination and iteration. After training, automatically generated

(a) First Person View

(b) Side View

Figure 6: Example illustration of an augmented-reality (AR) task where the user has to sequentially complete pointing and button clicking tasks.

videos of the learned behaviour support qualitative assessment of the resulting movement strategies (step 6 in Figure 1).

4.5.2 Public Display Interaction. Public kiosks, ATMs and information displays require interaction with vertical touch surfaces, raising questions about how vertical target placement affects reachability and movement effort. Figure 7a illustrates an example public display task modelled with *MyoInteract*. The simulated user is tasked with pressing the buttons in the pre-specified order. By varying the height and spacing of the targets in the GUI, this setup allows researchers to rapidly estimate effort costs for different kiosk layouts without building physical mockups.

4.5.3 Mobile Keyboard Typing. Text entry on smartphones involves rapid, sequential pointing to small targets on a horizontal surface held below shoulder height. Figure 7b illustrates how *MyoInteract*'s combinatorial task setup can chain multiple pointing targets on a planar surface to simulate touchscreen typing. Target positions are sampled from user-defined ranges; by setting identical upper and lower bounds for the vertical axis, we constrain all targets to a single horizontal plane matching the screen surface. The depicted phone surface and screen elements (resembling an iPhone 17) are illustrative only, while the interaction logic is handled entirely by the standard pointing primitives.

5 System Verification

In the following, we demonstrate the speed improvement that *MyoInteract* provides over the current state-of-the-art in biomechanical interaction task simulations *User-in-the-Box (UitB)* [24], and verify that *MyoInteract* predicts biomechanically plausible movements.

5.1 Fast Biomechanical RL Simulations

To benchmark the training duration of biomechanical user simulations, we use the standard pointing task and adhere to the task configuration as implemented in the UitB codebase[4], i.e., the simulated user is asked to point to 10 targets in a row, without a reset in between. All experiments were conducted on identical hardware (NVIDIA RTX 5090 GPU). For each training, we determined the number of steps until the policy converged based on manual inspection of the reward loggings. We additionally ensured that the resulting policy achieved a success rate of at least 95%, which was the case for all runs.

Table 1 summarises the results. On our hardware, training the UitB pointing task took 6.9 hours on average—already faster than the originally reported 24-48 hours in [24], likely due to the use of more recent GPUs. To assess the upper bound of speed achievable within our system, we reimplemented the pointing task from [24] as closely as possible within our framework, ensuring that the reset methods, reward function, and control type closely match those of the original work. The only notable differences arise from slightly different implementations of the PPO algorithm (MyoInteract uses the Brax [19] RL library, while UitB relies on Stable Baselines 3 [49]), different RL hyperparameters due to increased parallelisation, and the direct provision of information about the target state instead of visual inference. This configuration, which we call *MyoInteract (Replication)*, **completes training in ten minutes on average**—a 98% reduction in training duration (more than 40x faster than the original).

However, this tuned environment is highly specific to the pointing task and does not generalise well to other types of interaction. To support broader use cases, we also evaluated our *MyoInteract (Default)* configuration, which employs the combinatorial task setup and reward formulation described in Section 4. This configuration supports a wider range of sequential, movement-based interactions—such as pointing combined with button clicking—without task-specific tuning. Even with this more general and thus more "usable" setup, training completes in approximately 36 minutes, representing a 91% reduction in training time (more than 10x faster) compared to UitB's 6.9-hour baseline.

In Table 2, we also show that the three demonstration examples introduced in Section 4.5 can successfully be trained in less than 30 minutes.

[4] https://github.com/User-in-the-Box/user-in-the-box

(a) In the public display task, the simulated user has to sequentially hit 4 buttons on a hypothetical public display. The grey display board is only shown for visualisation and does not affect the learned policy.

(b) In the mobile typing task, the simulated user has to sequentially point to each of the five randomly spawned targets on a virtual touch-screen surface resembling the size of an iPhone 17.

Figure 7: Example illustrations of the public display interaction and mobile keyboard typing tasks generated with *MyoInteract*.

Experiment	Num Runs	Num Steps (M)	Total Training Time (hrs)
UitB [24]	-	40-80	24-48
UitB (On Our Machine)	3	42.9 ± 9.0	6.9 ± 1.5
MyoInteract (Replication)	5	7.4 ± 0.8	0.17 ± 0.03
MyoInteract (Default)	5	34.9 ± 1.9	0.60 ± 0.03

Table 1: A comparison of training steps and wall times required for training until convergence in the default pointing task [24] (mean ± std. observed across the respective number of runs).

Experiment	Num Subtasks	Num Runs	Num Steps (M)	Total Training Time (min)
AR	8	5	9.9 ± 1.3	23.2 ± 2.2
Public Display	4	5	6.7 ± 2.5	14.9 ± 4.9
Mobile Typing	5	5	10.4 ± 4.1	11.7 ± 3.9

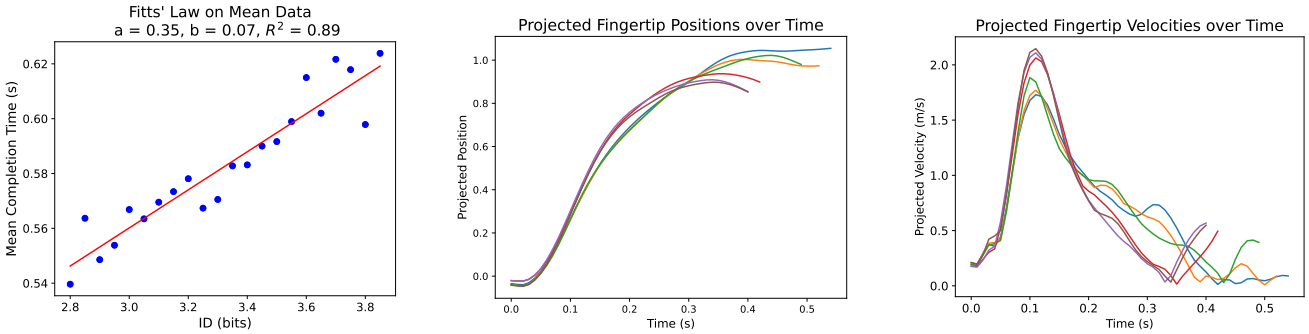
Table 2: Training time and steps required to train until convergence for three example interaction scenarios (mean ± std. observed across the respective number of runs).

5.2 Motion Evaluation

We also examine whether the movement policies generated by MyoInteract follow established movement regularities such as Fitts' Law. We use the *MyoInteract (Default)* policy trained on the default pointing task and evaluate it across a range of indices of difficulty (IDs), using a sampling frequency of 100 Hz [24]. As shown in Figure 8a, the policy trained with our framework adheres to Fitts' Law ($R^2 = 0.89$). Moreover, the fingertip moves smoothly toward the target and remains close for the set dwell time (Figure 8b), and the velocity profiles exhibit the typical bell-shaped pattern [38], followed by a second, smaller corrective submovement required to keep the fingertip inside the target for the required dwell duration ("settling phase") (Figure 8c). These results are consistent with previous findings [16, 24, 36], demonstrating a comparable level of movement regularity to previous biomechanical simulations, but at much faster speeds. Similar results were obtained for the *MyoInteract (Replication)* policy (see Appendix C).

6 Workshop Evaluation

To assess the usability and potential of our framework and GUI, we conducted an evaluation study with 12 HCI researchers, most of them with little to no prior knowledge in RL and computational user simulations. We were mainly interested in whether participants were able to setup and run trainings, and how useful the provided GUI features and metrics were in monitoring and assessing the quality of learned user behaviour. In addition, we asked participants about their opinion of biomechanical RL as an HCI research and design method, and how they would integrate biomechanical user simulations into their workflow.



(a) Fitts' Law Evaluation **(b) Projected Positions** **(c) Projected Velocities**

Figure 8: Biomechanical validation of the policy trained using MyoInteract (Default). (a) Average movement times (blue dots) plotted against IDs of targets. The observed movement times follow a clear linear relationship (red line), as predicted by Fitts' Law, with a fit of $R^2 = 0.89$. (b) Projected fingertip positions for a trained policy, moving from the initial position (0) to the target position (1) and settling there, for six different targets. (c) Corresponding projected fingertip velocities, exhibiting the characteristic bell-shaped profile of aimed reaching movements [38].

6.1 Methodology and Procedure

The study was exploratory in nature and primarily focused on qualitative findings, complemented by selected quantitative measures. It was approved by the departmental ethics committee in advance. The study was conducted as part of a two-hour workshop session on biomechanical RL, which included a brief tutorial (20 mins) on the basics of biomechanical modelling, reinforcement learning, and reward function design, ensuring participants had a common foundational understanding before engaging with the study tasks.

Each participant completed the study individually following a structured procedure. After the tutorial, each participant received a link to the graphical user interface described above, running on a cloud-hosted server with GPU access (NVIDIA RTX 5090). They were then asked to open the GUI and were given a short introduction into the possible task setups and other parameters that can be modified in the simple mode. Afterwards, each participant was asked to define an interaction task of their choice including up to 10 subtasks. Participants with a specific interaction or interface concept in mind were encouraged to recreate it using the provided tools. Participants were allowed to use one of the preloaded configurations (see Section 4.5), but then had to modify at least two task parameters. While the simple GUI was sufficient for completing the task, more experienced users could optionally explore the advanced parameter settings (see Section 4.3).

Participants were given 15 minutes to configure their desired task, adjust relevant parameters, and start training. Afterwards, they were instructed to verify that the training was running correctly and that the results were being logged as expected. During training, which lasted between 20 and 60 minutes for their chosen configurations, they could either take a break or monitor progress using the provided metrics (Section 4.4). After training completed, participants spent approximately 5 minutes evaluating the resulting policy based on the generated plots and videos (steps 5 and 6 in Figure 1) and reflecting on how the policy might be further improved.

Following the practical task, participants completed an online questionnaire consisting of open-ended questions and 5-point Likert scales. The questionnaire included both standardised instruments (demographic questions) and task-specific items addressing (1) the process of defining and training RL tasks, (2) the usability of the interface for policy setup and training, and (3) the monitoring and evaluation of trained policies using WandB. Participants were also asked to reflect on whether the trained policy met their expectations and how they might adjust parameters in a subsequent iteration. An overview of all questions can be found in Appendix D.

In the final stage of the study, participants took part in a group discussion on the potential use cases of the framework, its relevance to their own research or projects, and desired features for improving its usefulness. The group discussion was recorded with participants' consent and subsequently transcribed.

6.2 Participants

This workshop study included 12 participants (9 male, 3 female) with an average age of 28.5 years. Their levels of expertise ranged from novice to expert across reinforcement learning (8 novices, 3 intermediate, 1 expert), biomechanical modelling (9 novices, 2 intermediate, 1 expert), and cognitive modelling (7 novices, 3 intermediate, 1 expert). Participants were not compensated for their participation.

6.3 Results

We structure our results across the themes we identified from participants' responses.

Reinforcement Learning Setup and Training. Participants generally found the interface intuitive and accessible for both defining a reinforcement learning task and training a simulated user. Participants were able to configure and manipulate tasks effectively, fostering an accessible and engaging learning experience: *"The interface made it easy to input necessary parameters without requiring extensive background knowledge in RL. The parameter setup was*

intuitive and accessible, allowing me to configure the task even as a non-expert in this field." (P12). Participants also appreciated the system's guided workflow: *"I found it useful that the interface guided me step by step and presented the process in a straightforward way. The clear sequence of actions and visual feedback made the training easy to follow and reduced confusion during the setup."* (P5).

Several participants highlighted certain GUI elements. For example, P11 mentioned, *"defining a task was intuitive, especially because there was immediate visual feedback available in the notebook cell"*, while P5 found that, *"this visual interaction helped me understand how different parameter settings might influence the task design"*. The use of parameter sliders was consistently considered intuitive and informative, enabling participants to explore meaningful value ranges without prior knowledge (P3, P5, P7). For example, P5 emphasised that sliders *"made it very clear and intuitive to define the Reinforcement Learning task, since I could easily adjust values without needing to type or remember specific commands"*. Users also found the customisable configuration of object properties (P9) and the comprehensive parameter view (P10) particularly useful, supporting an efficient task setup. Additional features such as reward function visualisation and parameter configuration via dials were seen as valuable tools for refining training tasks (P8, P11).

Quantitatively, participants rated the process of defining and setting up a reinforcement learning task as relatively easy (mean = 3.58, SD = 0.90) and not particularly difficult (mean = 2.33, SD = 0.89). Likewise, training a simulated user was perceived as easy (mean = 3.75, SD = 0.97) and not difficult (mean = 2.18, SD = 0.75). These ratings indicate that participants were generally able to engage effectively with the system and complete the setup and training without major obstacles.

Policy Monitoring and Analysis. Participants emphasised the need for clear and interpretable **visualisations** to effectively monitor training progress and assess policy quality. Video demonstrations and visual footage of the simulated user's behaviour were consistently regarded as intuitive means of understanding policy performance (P3, P10, P11, P12). For example, P11 found, *"the visual footage helpful to intuitively get a sense of how successful the policy was"*. Participants relied on success rates, loss curves, episode lengths and reward component values to track learning progress and evaluate training outcomes (P1, P5, P6, P7, P8).

While some participants noted that the existing metrics were already sufficient for their analysis needs (P7, P8), others suggested extensions to further improve monitoring and analysis. These included visualising intermediate policies during training to illustrate performance improvement over time (P1), incorporating additional metrics such as arm movement speed (P4), and providing clearer descriptions and prioritisation of metrics to help users focus on the most relevant information: *"In general I would have needed way less metrics and a focus on the most important ones would have been nice. Also a better description of the metrics, so what exactly am I seeing here."* (P8). Other suggestions included reward *space* visualisation using a Principal Component Analysis (P10) and methods to link metrics directly to observed agent behaviour for more intuitive interpretation (P11).

Quantitative ratings reflected these qualitative findings. Participants found it slightly easier on average to assess the quality of the trained policy (mean = 3.75, SD = 1.14) than to monitor the training progress (mean = 3.42, SD = 1.08), with corresponding difficulty ratings of 2.25 (SD = 1.22; assessing) and 2.67 (SD = 0.99; monitoring), respectively.

Policy Performance and Next Steps. Most participants reported being satisfied with the performance of the trained policy (P1, P2, P5, P7, P9, P10, P11, P12). Several participants proposed refinements to further improve results. Most suggestions focused on modifying the reward function weights, e.g., to reduce motion jitter or encourage faster, more natural movements (P1, P4, P7). Others emphasised changes in the task configuration, such as adjusting the target distribution (P10), using *"more uniform target settings for each targets to make some generalised human motion for diverse ranges"* (P10), *"narrowing down the sampling coordinates a bit more"* (P4), or reducing task complexity (P8). Given that the policies already achieved strong results, some participants expressed interest in further exploring the boundaries of the simulated user: *"I hope to try out more bizarre/ambitious training setup next time."* (P11), for instance by using smaller objects or targets placed at larger distances (P3). One participant noted that *"It was beyond my expectation. It seemed to have understand physical constraints well because sometimes movement was blocked by torso."* (P9).

Suggestions for UI Improvement. Participants identified several opportunities to improve the interface and further simplify both the definition of tasks and the training of simulated users. A key theme concerned the need for more intuitive and informative visual feedback (P1, P3, P7, P8, P9, P10, P11). Participants suggested aligning parameter sliders with a live scene preview featuring automatic rendering: *"I would rather have a direct manipulation interface for adjusting object positions, or at the very least put the object sliders side-by-side with the scene preview and have it auto-render."* (P1), or showing 3D views: *"It would be better if I could set the parameters with corresponding 3D GUI, which can make the usage more intuitive."* (P10).

Participants also emphasised that additional guidance during task configuration, such as more recommendations on useful parameter ranges or default values, could help further reduce uncertainty and support informed decision-making (P5). One participant suggested adding a preview function to completely eliminate the need for full training in certain cases: *"visualisation of expected changes of how the change in parameters would influence the learning task and training would help understand."* (P3). Other participants requested improved real-time training feedback (P11, P12) and the ability to adjust parameters *during* training (P4). Suggestions include adding interface elements such as a progress bar or estimated completion time, as well as clear indicators showing whether the training is running or visualisations are being generated (P12). Participants also recommended prioritising a small set of core metrics (e.g., loss, accuracy, precision) to support clarity and reduce cognitive load, particularly for non-expert users.

Potential Use Cases. Participants discussed the practical potential of MyoInteract. One participant observed that *"in the real world of medical service or games or whatever, there are some actual behaviours that is resembled to those simple button-pressing behaviours"*, while another highlighted the PlayStation joystick as an example of a

forearm-bending task comparable to those studied. These comments motivated suggestions to embed additional real-world tasks and input devices into the framework.

7 Discussion

Previously, training simulated users via biomechanical RL took hours to days and required substantial expertise in computational user models and how they are implemented. Researchers who wanted to test a design hypothesis or low-fidelity prototype in silico could not resort to a default user model and train it "out of the box", but had to first familiarise themselves with the implementation and code details of complex simulation frameworks and physics engines such as UitB [24], MyoSuite [8], or MuJoCo [61], before they could setup and start hour-long training runs that lack substantial feedback on the progress and quality of the simulation. Our main motivation for developing MyoInteract was to bridge these *gulfs of execution and evaluation* [42] and make biomechanical RL simulations accessible to researchers in the field of interaction design. In the following, we reflect on the extent to which MyoInteract fulfils these design goals, the potential it unleashes for HCI research and interaction design prototyping, and lessons learned that we believe generalise to biomechanical user simulations in the scope of HCI. We conclude with a discussion on current limitations and future extensions.

7.1 Reflections on the Design Goals

We designed *MyoInteract* to lower the entry barrier to biomechanical RL for interaction design research by tackling three design goals (DGs). Using Norman's action cycle [42], this involves addressing the gulfs of execution (DG1) and evaluation (DG2) while also reducing the temporal costs that compound both gulfs (DG3). The workshop study provides evidence of how well the system met these three design goals and reveals important limitations and opportunities for improvement.

A primary bottleneck in prior biomechanical RL workflows is the time required to train a policy for a given interaction task, which often forces researchers to wait hours or days before assessing its predictions. Consequently, we built upon recent advances in GPU-accelerated physics simulation [60, 65] and developed a biomechanical framework that achieves substantial training speed-ups through massive parallelisation. In fact, our implementation of the standard pointing task from [24] reduced the **training time (DG3)** by a factor of 10-40x. Moreover, all workshop participants were able to complete their training runs within 20-60 minutes, across a range of task configurations. Notably, the improvements in speed did not compromise the biomechanical plausibility of the predicted user trajectories, as was demonstrated in Section 5 through a Fitts' Law study and a qualitative analysis of the predicted target acquisition and selection movements. This shows that MyoInteract has successfully achieved design goal DG3.

Moreover, we applied the design principles of *Abstraction via Task Decomposition*, *Visibility via GUI*, *Constrained Interaction*, and *Progressive Disclosure* (see Section 3.1) to reduce the **gulf of execution (DG1)**. Workshop participants described the GUI as intuitive and accessible, and several participants remarked that they were able to configure and run trainings "even as a non-expert in the field". Features particularly appreciated include GUI sliders (DP3), (adaptive) default values (DP3, DP5), and the guiding structure and modes of the GUI (DP6), which allowed the participants to focus on testing different interactions and task configurations that interested them, rather than getting bogged down in technical implementation details. Feedback from participants also revealed current limitations of MyoInteract. While the GUI provides renderings of the environment from two different camera perspectives, several participants expressed an interest in an interactive 3D view of the interaction environment that enables direct manipulation of target positions and sizes. Similarly, being able to setup target configurations interactively via sliders, as provided by our GUI, is an advantage over using Python scripts and config files, however, our study revealed a clear demand for more direct and "natural" manipulation methods. In summary, MyoInteract was successful in reducing the gulf of execution, while the workshop study revealed promising directions for further improvements that need to be investigated in more detail.

To reduce the **gulf of evaluation (DG2)**, MyoInteract provides *Multi-Level Feedback* (DP4) before, during, and after training, including environment previews, decomposed metrics, and videos showing the learned behaviour. Participants placed a high value on the provided visual feedback. In particular, videos of trained policies were considered crucial for evaluating the performance of trained policies. Decomposed metrics, such as per-target success rates and individual reward components, enabled analytical inspection and allowed participants to identify potential shortcomings and ideate parameter improvements for future iterations. From the workshop study, we learned that intermediate visualisations during the training can complement these metrics, resulting in a comprehensive set of options for monitoring and assessing simulation results. Moreover, some participants found the number of available metrics overwhelming and expressed a desire for stronger guidance on how to interpret these quantities. These findings suggest that our approach effectively reduced the gulf of evaluation by exposing relevant training feedback, but that further addressing this gulf also requires careful consideration of when and how such feedback is presented which should be further examined.

Overall, the workshop evaluation indicates that MyoInteract is making meaningful progress towards increasing the usability and utility of biomechanical RL for interaction design, while highlighting specific areas that can be further improved. The feedback from participants provides valuable guidance for refining the balance between speed, abstraction, and interpretability in future iterations of the framework.

7.2 MyoInteract as an Enabling System for HCI Research and Beyond

Biomechanical RL's potential for HCI has been demonstrated in isolated studies, yet its broader value remains unproven—largely because the barriers to entry have prevented widespread exploration. By removing these barriers, our framework lowers both the expertise and computational thresholds for entry, and makes biomechanical RL an interactive tool for HCI researchers to test a variety of interaction designs in silico. For example, MyoInteract can be used to investigate the ergonomics of different UI layout within an XR application, learn about the effect of design choices on users'

movements (e.g., to which extent they bend the elbow or move the arm back and forth), or identify the optimal design of a set of buttons (location, size, physical properties), ensuring reachability, precision, and performance. More generally, we believe that MyoInteract can be useful for testing any interaction designs that involve an infinite parameter space and where ergonomics and biomechanical measures such as movement patterns or physiological effort are essential design factors.

Moreover, MyoInteract can be used to assess and improve the quality of biomechanical user simulations. Given that our framework offers a large range of options for adjusting parameters of the simulated user and provides a fast and usable (though clearly simplified) workflow for biomechanical RL simulations that enhances the current state of the art, we believe that MyoInteract will be valuable not only to the HCI community, but also to simulation engineers and researchers developing computational models of user behaviour. Ultimately, MyoInteract has the potential to catalyse collaborations between interaction designers and other disciplines adjacent to HCI, including computational user modelling, cognitive psychology, sports and rehabilitation, and neurorobotics.

7.3 Design Reflections for Biomechanical User Simulations for HCI Tasks

The design and evaluation of MyoInteract revealed insights that transcend this framework and extend to biomechanical user simulations in the broader context of interaction design and prototyping.

Iteration costs shape feasible opportunities. Reducing training time from days to minutes did not merely accelerate existing workflows—it enabled qualitatively different ones. Rather than committing to a single configuration, users can run multiple simulations within a single session, using outcomes from one run to inform the next. This shift enables exploratory behaviours such as comparing task variants, probing edge cases, and incrementally refining rewards or layouts—activities central to interaction design but previously infeasible with long training cycles. More generally, in biomechanical user simulations, the cost of iteration determines not only the efficiency, but also whether iterative exploration is possible at all.

Lowering interaction friction accelerates learning without removing complexity. MyoInteract demonstrates that lowering entry barriers is less about simplifying underlying models than about how they are exposed. Making options visible, defaults explicit, and ranges constrained enables productive use before full conceptual understanding. Progressive disclosure supported task setup, but participants' responses also showed that unprioritised metric visibility can be overwhelming. This suggests that reducing friction in biomechanical user simulation frameworks requires not only accessible interfaces, but deliberate curation of what information is surfaced, and at which stage of the workflow.

Black-box methods demand intermediate and immediate observability. Biomechanical simulation methods remain difficult to interpret, especially when training fails or produces unexpected behaviour. By exposing decomposed metrics during training, MyoInteract supported systematic diagnosis rather than ad-hoc speculation, and enabled earlier intervention when runs were clearly unproductive. Simulation methods that operate as black boxes should provide intermediate, interpretable feedback to support sense-making and debugging during training, rather than deferring insight to post-hoc evaluation.

Expressivity and usability trade off against each other. Configuration-driven task authoring allowed non-programmers to create simulations, but necessarily constrained the space of expressible interactions. Similar tensions surfaced throughout the system: exposing more parameters or metrics increases potential insight and control, yet raises cognitive load and can hinder effective use. These trade-offs have no universal solution and require context-dependent design decisions. Our workshop highlights both successful choices and missteps, underscoring the importance of iterative refinement grounded in user feedback.

7.4 Limitations and Future Work

Although MyoInteract shows great promise for HCI research and interaction design, there are some clear limitations that future work should address. Most importantly, the range of interaction tasks captured by MyoInteract is currently limited to sequential target acquisition and selection via direct manipulation. While direct and sequential target selection forms the basis of most GUI-based interaction and allows investigating the ergonomics of XR, mobile, and other touch- and clicking-based interfaces, future work should further aim to increase the task scope. Suggestions from our workshop participants include modelling input devices such as joysticks or VR controllers and indirect manipulation methods such as raycasting or Go-Go [47].

In addition, our framework currently only focuses on biomechanical aspects of human-computer interaction. While these are essential for a range of movement-based interaction tasks [4], extending musculoskeletal simulations executed in a physics engine such as MuJoCo with computational models of human cognition and perception would enable more holistic simulations that capture the entire sensorimotor control loop [17]. Future work should explore opportunities for integrating models from computational rationality [9, 28, 44] with biomechanical RL simulations, and align the observation space with multimodal perception models. This particularly includes modelling how users visually sense and perceive their environment, beyond the prevailing monocular, static, and low-resolution vision model currently implemented in simulation frameworks such as UitB [24].

MyoInteract currently trains each configuration from scratch, though our unified observation space and reward structure has the potential to learn skills that transfer across tasks. Future work should investigate this potential, exploring for example warm-starts or the training of foundational policies that adapt to unseen tasks without fine-tuning.

Finally, our framework has only been tested with a single biomechanical model of the upper extremity. Notably, MyoInteract comes with an experimental hand model based on the *MyoArm* model, which in principle allows creating user simulations for more dexterous tasks such as gamepad control or multi-touch typing. Moreover, users with more biomechanical expertise can readily add other models, e.g., two-armed upper body models as the one used in [22], or the hand, back, or leg models provided by the MyoSim library[5],

[5] https://github.com/MyoHub/myo_sim

which MyoInteract is fully compatible with. It would be interesting to compare these models in terms of speed, scope, and robustness, and explore how to best integrate these models into MyoInteract regarding the design trade-off between complexity and usability.

8 Conclusion

In this paper, we introduced *MyoInteract*, a novel framework for fast prototyping of biomechanical HCI tasks. Using the Human Action Cycle as a design lens, we identified three fundamental barriers which affect the accessibility, iterability and interpretability of biomechanical simulations in HCI. We reduce the temporal cost by utilising GPU acceleration, which reduces training times by up to 98%, enabling practical iterations for the first time as training times reduce from days to under an hour, or even just a few minutes. We address the gulf of execution by utilising a combinatorial task setup and GUI to construct a system, which enables exploration of more expressive interaction interfaces than previous systems, while being easy to learn and use. Multi-level feedback, including real-time metrics, provides a greater degree of insight into the black-box nature of RL training. This enables users to identify and iterate on poorly configured experiments or models early on rather than waiting for flawed runs to terminate, thus tackling the gulf of evaluation. As demonstrated through a workshop evaluation with 12 participants, *MyoInteract* can be learned by HCI researchers with little to no expertise in biomechanical modelling and RL within two hours, allowing them to successfully setup user simulations for a range of relevant HCI interaction tasks. Ultimately, *MyoInteract* does more than just accelerate computation; it accelerates insight. By making the physical body computationally accessible, we hope to establish biomechanics not just as a method for explaining movement-based interaction, but as a generative tool for designing it.

Acknowledgments

This work was supported by EPSRC grant EP/W02456X/1. The authors gratefully acknowledge the computing time made available to them on the high-performance computer at the NHR Center of TU Dresden. Hannah Selder and Arthur Fleig acknowledge the financial support by the Federal Ministry of Research, Technology and Space of Germany and by Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus in the programme Center of Excellence for AI-research „Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig“, project identification number: ScaDS.AI.

References

[1] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Y. Zou. 2019. Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. *CoRR* abs/1906.02569 (2019). arXiv:1906.02569 http://arxiv.org/abs/1906.02569

[2] Marko Ackermann and Antonie J Van den Bogert. 2010. Optimality principles for model-based prediction of human gait. *Journal of biomechanics* 43, 6 (2010), 1055–1060.

[3] Myroslav Bachynskyi, Antti Oulasvirta, Gregorio Palmas, and Tino Weinkauf. 2014. Is motion capture-based biomechanical simulation valid for HCI studies? study and implications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3215–3224. doi:10.1145/2556288.2557027

[4] Myroslav Bachynskyi, Gregorio Palmas, Antti Oulasvirta, Jürgen Steimle, and Tino Weinkauf. 2015. Performance and ergonomics of touch surfaces: A comparative study using biomechanical simulation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1817–1826.

[5] Bastien Berret, Enrico Chiovetto, Francesco Nori, and Thierry Pozzo. 2011. Evidence for Composite Cost Functions in Arm Movement Planning: An Inverse Optimal Control Approach. *PLoS Computational Biology* 7, 10 (Oct. 2011), e1002183. doi:10.1371/journal.pcbi.1002183

[6] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. *JAX: composable transformations of Python+NumPy programs*. http://github.com/jax-ml/jax

[7] Vittorio Caggiano, Sudeep Dasari, and Vikash Kumar. 2023. MyoDex: a generalizable prior for dexterous manipulation. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) *(ICML'23)*. JMLR.org, New York, NY, USA, 20 pages.

[8] Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. 2022. MyoSuite: A Contact-rich Simulation Suite for Musculoskeletal Motor Control. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference (Proceedings of Machine Learning Research, Vol. 168)*, Roya Firoozi, Negar Mehr, Esen Yel, Rika Antonova, Jeannette Bohg, Mac Schwager, and Mykel Kochenderfer (Eds.). PMLR, New York, NY, USA, 492–507. https://proceedings.mlr.press/v168/caggiano22a.html

[9] Suyog Chandramouli, Danqing Shi, Aini Putkonen, Sebastiaan De Peuter, Shanshan Zhang, Jussi Jokinen, Andrew Howes, and Antti Oulasvirta. 2024. A Workflow for Building Computationally Rational Models of Human Behavior. *Computational Brain and Behavior* 7, 3 (Sept. 2024), 399–419. doi:10.1007/s42113-024-00208-6 Publisher Copyright: © The Author(s) 2024..

[10] Jhon P.F. Charaja, Isabell Wochner, Pierre Schumacher, Winfried Ilg, Martin Giese, Christophe Maufroy, Andreas Bulling, Syn Schmitt, Georg Martius, and Daniel F.B. Haeufle. 2024. Generating Realistic Arm Movements in Reinforcement Learning: A Quantitative Comparison of Reward Terms and Task Requirements. In *2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, Piscataway, NJ, USA, 562–568. doi:10.1109/biorob60516.2024.10719719

[11] Alberto Silvio Chiappa, Pablo Tano, Nisheet Patel, Abigaïl Ingster, Alexandre Pouget, and Alexander Mathis. 2024. Acquiring musculoskeletal skills with curriculum-based reinforcement learning. *Neuron* 112, 23 (2024), 3969–3983.

[12] Alberto Silvio Chiappa, Alessandro Marin Vargas, Ann Zixiang Huang, and Alexander Mathis. 2023. Latent exploration for reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 2466, 23 pages.

[13] Omar Coser, Christian Tamantini, Paolo Soda, and Loredana Zollo. 2024. AI-based methodologies for exoskeleton-assisted rehabilitation of the lower limb: a review. *Frontiers in Robotics and AI* 11 (2024), 1341580.

[14] Joao Marcelo Evangelista Belo, Anna Maria Feit, Tiare Feuchtner, and Kaj Grønbæk. 2021. XRgonomics: Facilitating the Creation of Ergonomic 3D Interfaces. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 290, 11 pages. doi:10.1145/3411764.3445349

[15] Pouyan Firouzabadi, Wendy Murray, Anton R Sobinov, JD Peiffer, Kunal Shah, Lee E Miller, and R James Cotton. 2024. Biomechanical Arm and Hand Tracking with Multiview Markerless Motion Capture. In *2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 1641–1648.

[16] Florian Fischer, Miroslav Bachinski, Markus Klar, Arthur Fleig, and Jörg Müller. 2021. Reinforcement learning control of a biomechanical model of the upper extremity. *Scientific Reports* 11, 1 (2021), 14445.

[17] Florian Fischer, Arthur Fleig, Markus Klar, and Jörg Müller. 2022. Optimal Feedback Control for Modeling Human–Computer Interaction. *ACM Transactions on Computer-Human Interaction* 29, 6 (Dec. 2022), 1–70. doi:10.1145/3524122

[18] Florian Fischer, Aleksi Ikkala, Markus Klar, Arthur Fleig, Miroslav Bachinski, Roderick Murray-Smith, Perttu Hämäläinen, Antti Oulasvirta, and Jörg Müller. 2024. SIM2VR: Towards Automated Biomechanical Testing in VR. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. ACM, Pittsburgh PA USA, 1–15. doi:10.1145/3654777.3676452

[19] C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. 2021. *Brax - A Differentiable Physics Engine for Large Scale Rigid Body Simulation*. http://github.com/google/brax

[20] T.R.G. Green and M. Petre. 1996. Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. *Journal of Visual Languages & Computing* 7, 2 (1996), 131–174. doi:10.1006/jvlc.1996.0009

[21] Christopher M Harris and Daniel M Wolpert. 1998. Signal-dependent noise determines motor planning. *Nature* 394, 6695 (1998), 780–784.

[22] Lorenz Hetzel, John Dudley, Anna Maria Feit, and Per Ola Kristensson. 2021. Complex Interaction as Emergent Behaviour: Simulating Mid-Air Virtual Keyboard Typing using Reinforcement Learning. *IEEE Transactions on Visualization and Computer Graphics* 27, 11 (2021), 4140–4149. doi:10.1109/TVCG.2021.3106494

[23] Seongwoong Hong and Sukyung Park. 2024. Biomechanical Optimization and Reinforcement Learning Provide Insights into Ankle-to-Hip Strategy Transitions in Human Postural Control. *Scientific Reports* 15 (2024). doi:10.1038/s41598-025-97637-5

[24] Aleksi Ikkala, Florian Fischer, Markus Klar, Miroslav Bachinski, Arthur Fleig, Andrew Howes, Perttu Hämäläinen, Jörg Müller, Roderick Murray-Smith, and Antti Oulasvirta. 2022. Breathing Life Into Biomechanical User Models. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. ACM, Bend OR USA, 1–14. doi:10.1145/3526113.3545689

[25] Markus Klar, Florian Fischer, Arthur Fleig, Miroslav Bachinski, and Jörg Müller. 2023. Simulating Interaction Movements via Model Predictive Control. *ACM Transactions on Computer-Human Interaction* 30, 3 (June 2023), 1–50. doi:10.1145/3577016

[26] Francesco Lacquaniti, Carlo Terzuolo, and Paolo Viviani. 1983. The law relating the kinematic and figural aspects of drawing movements. *Acta Psychologica* 54, 1 (1983), 115–130. doi:10.1016/0001-6918(83)90027-6

[27] Thomas Langerak, Sammy Christen, Mert Albaba, Christoph Gebhardt, Christian Holz, and Otmar Hilliges. 2024. MARLUI: Multi-Agent Reinforcement Learning for Adaptive Point-and-Click UIs. *Proceedings of the ACM on Human-Computer Interaction* 8, EICS (June 2024), 1–27. doi:10.1145/3661147

[28] Richard L Lewis, Andrew Howes, and Satinder Singh. 2014. Computational rationality: Linking mechanism and behavior through bounded utility maximization. *Topics in cognitive science* 6, 2 (2014), 279–311.

[29] Yi Li, Florian Fischer, Tim Dwyer, Barrett Ens, Robert Crowther, Per Ola Kristensson, and Benjamin Tag. 2025. AlphaPIG: The Nicest Way to Prolong Interactive Gestures in Extended Reality. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 627, 14 pages. doi:10.1145/3706598.3714249

[30] Yi Li, Benjamin Tag, Shaozhang Dai, Robert Crowther, Tim Dwyer, Pourang Irani, and Barrett Ens. 2024. Nicer: A new and improved consumed endurance and recovery metric to quantify muscle fatigue of mid-air interactions. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–14.

[31] Yi-Chi Liao and Christian Holz. 2025. Redefining Affordance via Computational Rationality. In *Proceedings of the 30th International Conference on Intelligent User Interfaces (IUI '25)*. Association for Computing Machinery, New York, NY, USA, 1188–1202. doi:10.1145/3708359.3712114

[32] William Lidwell, Kritina Holden, and Jill Butler. 2010. *Universal principles of design, revised and updated: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design.* Rockport Pub.

[33] Ramona Maas and Sigrid Leyendecker. 2013. Biomechanical optimal control of human arm motion. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics* 227, 4 (2013), 375–389.

[34] Michał Patryk Miazga and Patrick Ebel. 2025. Increasing Interaction Fidelity: Training Routines for Biomechanical Models in HCI. , Article 99 (2025), 3 pages. doi:10.1145/3746058.3758385

[35] Takeshi Miki, Florian Fischer, John J Dudley, and Per Ola Kristensson. 2025. Enhancing XR Accessibility through Anthropometrically Diverse Biomechanical Simulation. In *2025 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 1186–1187. doi:10.1109/VRW66409.2025.00239

[36] Hee-Seung Moon, Yi-Chi Liao, Chenyu Li, Byungjoo Lee, and Antti Oulasvirta. 2024. Real-time 3D Target Inference via Biomechanical Simulation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–18. doi:10.1145/3613904.3642131

[37] Hee-Seung Moon, Antti Oulasvirta, and Byungjoo Lee. 2023. Amortized Inference with User Simulations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–20. doi:10.1145/3544548.3581439

[38] Pietro Morasso. 1981. Spatial control of arm movements. *Experimental brain research* 42, 2 (1981), 223–227.

[39] Roderick Murray-Smith, Antti Oulasvirta, Andrew Howes, Jörg Müller, Aleksi Ikkala, Miroslav Bachinski, Arthur Fleig, Florian Fischer, and Markus Klar. 2022. What simulation can do for HCI research. *Interactions* 29, 6 (Nov. 2022), 48–53. doi:10.1145/3564038

[40] Roderick Murray-Smith, John H. Williamson, and Sebastian Stein. 2025. Active Inference and Human–Computer Interaction. *ACM Trans. Comput.-Hum. Interact.* 32, 6, Article 63 (Dec. 2025), 45 pages. doi:10.1145/3762812

[41] Masaki Nakada, Tao Zhou, Honglin Chen, Tomer Weiss, and Demetri Terzopoulos. 2018. Deep learning of biomimetic sensorimotor control for biomechanical human animation. *ACM Transactions on Graphics* 37, 4 (Aug. 2018), 1–15. doi:10.1145/3197517.3201305 Publisher: Association for Computing Machinery (ACM).

[42] Donald Norman. 2002. *The design of everyday things.* Basic Books, London, England.

[43] Katharine Nowakowski, Karim El Kirat, and Tien-Tuan Dao. 2022. Deep reinforcement learning coupled with musculoskeletal modelling for a better understanding of elderly falls. *Medical & Biological Engineering & Computing* 60, 6 (2022), 1745–1761.

[44] Antti Oulasvirta, Jussi P. P. Jokinen, and Andrew Howes. 2022. Computational Rationality as a Theory of Interaction. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–14. doi:10.1145/3491102.3517739

[45] Tianhu Peng, Lingfan Bao, and CHengxu Zhou. 2025. Gait-Conditioned Reinforcement Learning with Multi-Phase Curriculum for Humanoid Locomotion. *arXiv preprint arXiv:2505.20619* (2025).

[46] Luka Peternel, Cheng Fang, Nikos Tsagarakis, and Arash Ajoudani. 2019. A selective muscle fatigue management approach to ergonomic human-robot comanipulation. *Robotics and Computer-Integrated Manufacturing* 58 (2019), 69–79.

[47] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. The go-go interaction technique: non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*. 79–80.

[48] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. 2024. Real-world humanoid locomotion with reinforcement learning. *Science Robotics* 9, 89 (2024), eadi9579.

[49] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research* 22, 268 (2021), 1–8. http://jmlr.org/papers/v22/20-1364.html

[50] Katherine R Saul, Xiao Hu, Craig M Goehler, Meghan E Vidt, Melissa Daly, Anca Velisar, and Wendy M Murray. 2015. Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. *Computer methods in biomechanics and biomedical engineering* 18, 13 (2015), 1445–1458.

[51] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347 (2017). arXiv:1707.06347 http://arxiv.org/abs/1707.06347

[52] Pierre Schumacher, Thomas Geijtenbeek, Vittorio Caggiano, Vikash Kumar, Syn Schmitt, Georg Martius, and Daniel FB Haeufle. 2025. Emergence of natural and robust bipedal walking by learning from biologically plausible objectives. *iScience* 28, 4 (2025).

[53] Pierre Schumacher, Daniel F.B. Haeufle, Dieter Büchler, Syn Schmitt, and Georg Martius. 2023. DEP-RL: Embodied Exploration for Reinforcement Learning in Overactuated and Musculoskeletal Systems. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*. ICLR, Appleton, WI, USA, 1–29. https://openreview.net/forum?id=C-xa_D3oTj6

[54] Hannah Selder, Florian Fischer, Per Ola Kristensson, and Arthur Fleig. 2025. Demystifying Reward Design in Reinforcement Learning for Upper Extremity Interaction: Practical Guidelines for Biomechanical Simulations in HCI. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, Article 95, 17 pages. doi:10.1145/3746059.3747779

[55] Hannah Selder, Florian Fischer, Per Ola Kristensson, and Arthur Fleig. 2025. What Makes a Model Breathe? Understanding Reinforcement Learning Reward Function Design in Biomechanical User Simulation. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 592, 10 pages. doi:10.1145/3706599.3719699

[56] Omar Shaikh, Shardul Sapkota, Shan Rizvi, Eric Horvitz, Joon Sung Park, Diyi Yang, and Michael S. Bernstein. 2025. Creating General User Models from Computer Use. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, Article 35, 23 pages. doi:10.1145/3746059.3747722

[57] Danqing Shi, Yujun Zhu, Jussi P. P. Jokinen, Aditya Acharya, Aini Putkonen, Shumin Zhai, and Antti Oulasvirta. 2024. CRTypist: Simulating Touchscreen Typing Behavior via Computational Rationality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–17. doi:10.1145/3613904.3642918

[58] Merkourios Simos, Alberto Silvio Chiappa, and Alexander Mathis. 2025. Reinforcement learning-based motion imitation for physiologically plausible musculoskeletal motor control. *arXiv preprint arXiv:2503.14637* (2025).

[59] Chun Kwang Tan, Cheryl Wang, Shirui Lyu, Balint K Hodossy, Pierre Schumacher, Elizabeth B Wilson, Vittorio Caggiano, Vikash Kumar, Dario Farina, Letizia Gionfrida, et al. 2025. Myoassist 0.1: Myosuite for Dexterity and Agility in Bionic Humans. In *2025 International Conference On Rehabilitation Robotics (ICORR)*. IEEE, 437–442.

[60] William Thibault, William Melek, and Katja Mombaur. 2024. Learning Velocity-Based Humanoid Locomotion: Massively Parallel Learning with Brax and MJX. In *Climbing and Walking Robots Conference*. Springer, 278–283.

[61] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Piscataway NJ USA, 5026–5033. doi:10.1109/IROS.2012.

6386109

[62] Robert J van Beers, Patrick Haggard, and Daniel M Wolpert. 2004. The role of execution noise in movement variability. *J. Neurophysiol.* 91, 2 (Feb. 2004), 1050–1063.

[63] Rohan Walia, Morgane Billot, Kevin Garzon-Aguirre, Swathika Subramanian, Huiyi Wang, Mohamed Irfan Refai, and Guillaume Durandau. 2025. Myoback: A musculoskeletal model of the human back with integrated exoskeleton. *bioRxiv* (2025), 2025–03.

[64] Huawei Wang, Vittorio Caggiano, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. 2022. MyoSim: Fast and physiologically realistic MuJoCo models for musculoskeletal and exoskeletal studies. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 8104–8111.

[65] Kevin Zakka, Baruch Tabanpour, Qiayuan Liao, Mustafa Haiderbhai, Samuel Holt, Jing Yuan Luo, Arthur Allshire, Erik Frey, Koushil Sreenath, Lueder A Kahrs, et al. 2025. Mujoco playground. *arXiv preprint arXiv:2502.08844* (2025).

[66] Xinnong Zhang, Jiayu Lin, Xinyi Mou, Shiyue Yang, Xiawei Liu, Libo Sun, Hanjia Lyu, Yihang Yang, Weihong Qi, Yue Chen, et al. 2025. Socioverse: A world model for social simulation powered by llm agents and a pool of 10 million real-world users. *arXiv preprint arXiv:2504.10157* (2025).

A Technical Details for MyoInteract

A.1 Biomechanical Model

In the biomechanical model adapted from the UitB framework[6], the torso, wrist, and fingers are fixed, resulting in a model with five degrees of freedom—three at the shoulder, one for elbow flexion, and one for forearm pronation-supination—driven by 26 Hill-type muscles.

Motor execution noise is defined as follows:

$$a = \text{clip}_0^1\left((1 + k_{\text{SDN}} \cdot z) \cdot \tilde{a} + k_{\text{CN}}\right), \quad z \sim \mathcal{N}(0,1), \tag{1}$$

where $\tilde{a}$ and a denote the intended and the applied muscle control signals, respectively, z is a standard normal random variable, and $k_{\text{SDN}} \geq 0$ and $k_{\text{CN}} \geq 0$ are the selected signal-dependent and constant noise levels.

Muscle controls are updated at a frequency of 20 Hz by querying the current (during training) or trained (after training) policy with the latest observation. For the MuJoCo physics simulation, a higher internal update rate of 500Hz is used to ensure stability.

We implement three reset modes for the biomechanical model that determine the initial state at the beginning of each episode: *zero*, *epsilon-uniform*, and *range-uniform*. In the *zero* reset, all joint angles and velocities are initialised to zero, resulting in a relaxed body pose with the arm hanging down. The *epsilon-uniform* reset initialises the model with randomly perturbed joint angles and velocities around the zero pose (±0.05 rad and ±0.05 rad/s, respectively). Finally, the *range-uniform* reset selects the joint angles uniformly randomly from the valid joint range for each joint individually, while the velocities are selected as in the *epsilon-uniform* reset mode. For all modes, initial muscle activations are randomly sampled within the unit interval. The non-zero resets intend to reduce overfitting of the trained movements to a specific pose.

A.2 Task Setup

We define each pointing target using the following specification:

- Size: Each target is a sphere of a randomly-sampled radius $r \in [r_{min}, r_{max}]$, where the user can specify the lower and upper limits r_{min} and r_{max}.
- Location: Each target location is randomly drawn from a 3D space with boundaries $[x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}]$ specified by the user.
- Dwell duration: The minimum time (in seconds) the end-effector must be kept inside the target sphere to successfully complete the task. By default, this is set to 250 ms (corresponding to 5 time steps).
- Colour: The colour of the target sphere. While the target colour does not affect the learned behaviour of simulated users without vision, this property can be useful to setup task environments with multiple targets (see above in Section 4.5).

We model buttons as MuJoCo box geometries (i.e., cuboids) with the following properties:

- Size: Each button is a cuboid of user-specified size $w \times h \times d$ (width × height × depth), where w and h denote the width and depth of the button surface with d as its depth.
- Location: The 3D location of the centre of the button (x, y, z).
- Minimum touch force: The minimum force the end-effector must exert on the button to successfully complete the button clicking task.
- Orientation angle: The orientation angle of the button around the horizontal y-axis (in degrees), defining its tilt.
- Colour: The colour of the button.

A.3 Demonstration Tasks Implementation Details

A.3.1 AR Interaction. We have eight targets consisting of four buttons and four pointing targets, where the buttons are fixed in position while the pointing targets are randomly spawned in a target area. The task consists of repeated pointing and button pressing tasks where the buttons are represented as square of widths 5cm boxes with their $[y, z]$ coordinates in cm set to $[0.15, -0.29]$, $[0.15, 0.01]$, $[-0.15, -0.29]$, $[-0.15, 0.01]$, on a flat vertical plane 42cm in front of the user's shoulder respectively. The boxes representing the buttons are at 45 degrees around the horizontal and require a minimum touch force of 1N. The spherical pointing targets, which all have a dwell duration of 25ms, have their coordinates and radius are sampled according to:

- Target 1: $x \sim U(0.225, 0.35)$, $y \sim U(-0.15, 0.15)$, $z \sim U(-0.3, 0.3)$, $r \sim U(0.04, 0.08)$
- Target 3: $x = 0.3$, $y \sim U(-0.15, 0.15)$, $z \sim U(0, 0.3)$, $r \sim U(0.04, 0.08)$
- Target 5: $x = 0.3$, $y \sim U(-0.1, 0.1)$, $z \sim U(0, 0.3)$, $r \sim U(0.04, 0.08)$
- Target 7: $x \sim U(0.225, 0.35)$, $y \sim U(-0.15, 0.15)$, $z \sim U(-0.3, 0.3)$, $r \sim U(0.05, 0.15)$

A.3.2 Public Display Interaction. The buttons are represented as vertical square boxes with face widths of 5cm with their $[y, z]$ coordinates in cm set to $[-0.15, 0.01]$, $[0.15, 0.01]$, $[-0.15, -0.29]$, and $[0.15, -0.29]$, on a flat vertical plane 42cm in front of the user's shoulder respectively. All of these buttons required a minimum touch force of 1N to successfully register the button being pressed.

A.3.3 Mobile Keyboard Typing. The simulated user is tasked with pointing to five small targets located on a horizontal plane, resembling five-character keyboard typing on a mobile device. We use spheres of radius 1cm to define the pointing targets which have a dwell duration of 50ms. The spheres are randomly spawned on a horizontal plane of size 15 cm × 7.2 cm located 30cm below and 30cm front of the user's shoulder.

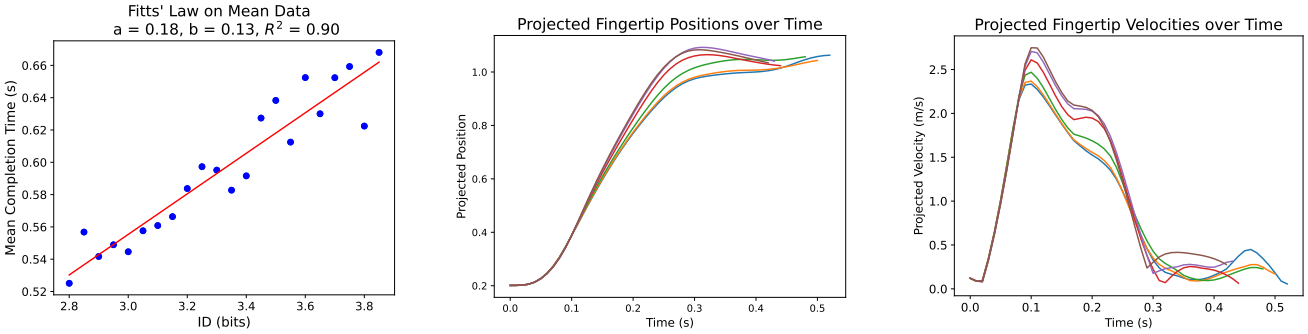
[6] https://github.com/User-in-the-Box/user-in-the-box

B Brax PPO Hyperparameters

Parameter	Value
Number of Environments	1024
Batch Size	128
Number of Mini-batches	8
Number of Updates per Batch	8
Unroll Length	10
Entropy Cost	0.001
Discount Factor	0.97
Learning Rate	0.0003
Reward Scaling	0.1
Clipping Epsilon	0.3
Policy Network MLP Layer Sizes	[128, 128, 128, 128]
Value Network MLP Layer Sizes	[256, 256, 256, 256, 256]

Table 3: Brax PPO hyperparameters used as defaults by our framework.

C Biomechanical Validation for MyoInteract (Replication)



(a) Fitts' Law Evaluation **(b) Projected Positions** **(c) Projected Velocities**

Figure 9: Biomechanical validation of the policy trained using MyoInteract (Replication).
(a) Average movement times (blue dots) plotted against IDs of targets. The observed movement times follow a clear linear relationship (red line), as predicted by Fitts' Law, with a fit of $R^2 = 0.90$. (b) Projected fingertip positions for a trained policy, moving from the initial position (0) to the target position (1) and settling there. (c) Corresponding projected fingertip velocities, exhibiting the characteristic bell-shaped profile of aimed reaching movements [38].

D Study Questions

Table 4: Overview of questionnaire items. The scales range from 1 (strongly disagree) to 5(strongly agree).

Question	Type
Policy Evaluation	
Please describe whether the trained policy satisfied your expectations. If the policy was not successful, how would you change the parameters in the next step?	text
RL Setup and Training	
It was easy for me to define and set up a Reinforcement Learning task.	scale
It was difficult for me to define and set up a Reinforcement Learning task.	scale
It was difficult for me to train a simulated user.	scale
It was easy for me to train a simulated user.	scale
Which aspects of the UI did you find useful for (1) defining a Reinforcement Learning task and (2) training a simulated user?	text
What could be improved in the UI to simplify (1) defining a Reinforcement Learning task and (2) training a simulated user?	text
Policy Monitoring and Analysis	
It was difficult for me to monitor the progress of the training.	scale
It was easy for me to monitor the progress of the training.	scale
It was difficult for me to assess the quality of the trained policy.	scale
It was easy for me to assess the quality of the trained policy.	scale
Which metrics and logs did you find useful to (1) monitor the training progress and (2) assess a learned policy?	text
Which other metrics and logs would you find useful to (1) monitor the training progress and (2) assess a learned policy?	text
Demographics	
Age	text
Gender	text
What is your experience level with Reinforcement Learning?	scale
What is your experience level with Biomechanical Models?	scale
What is your experience level with Cognitive Models?	scale

E GUI

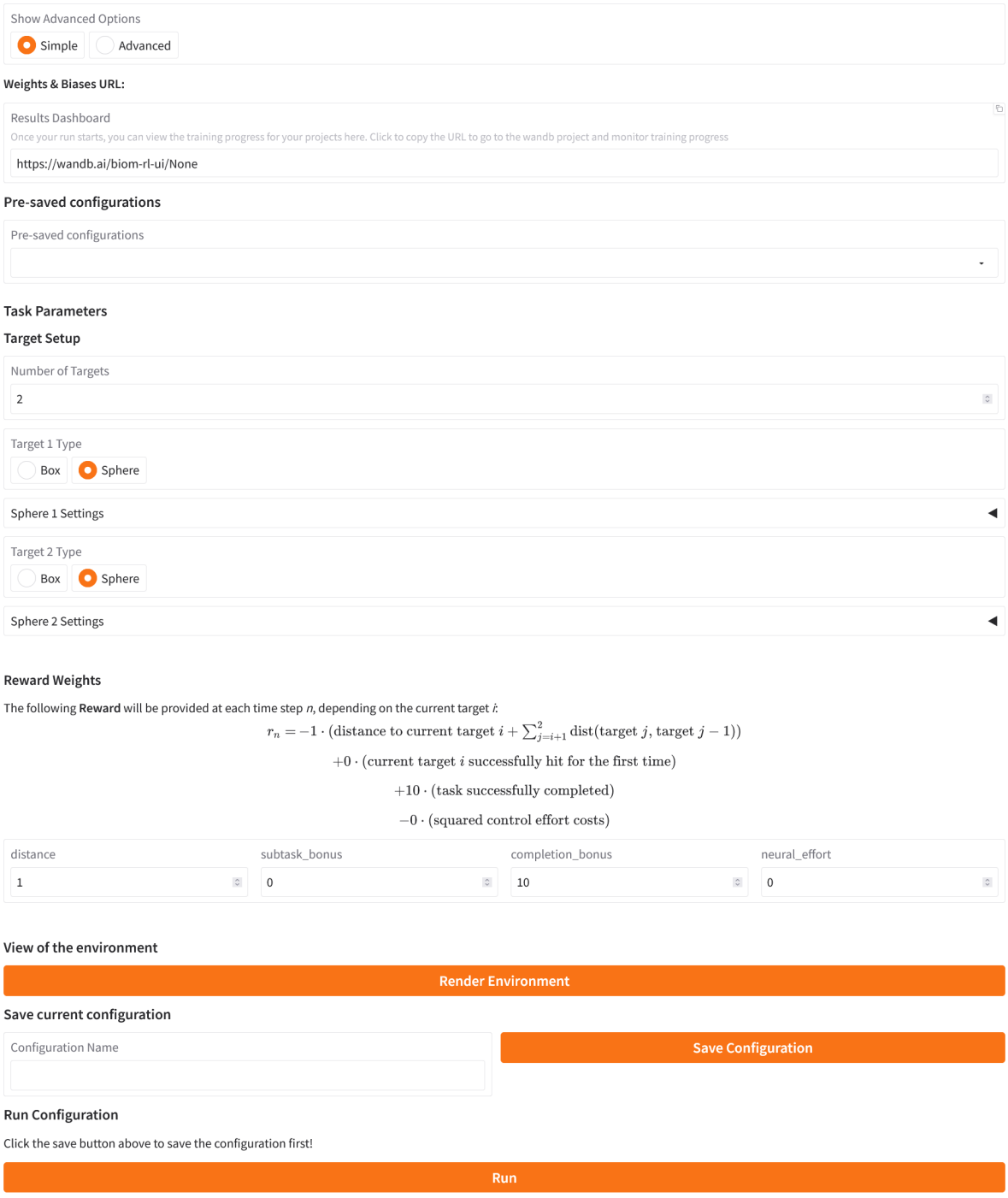


Figure 10: Simple version of our GUI for configuring reinforcement learning experiments, adjusting task and reward parameters, rendering the environment, and running training sessions.

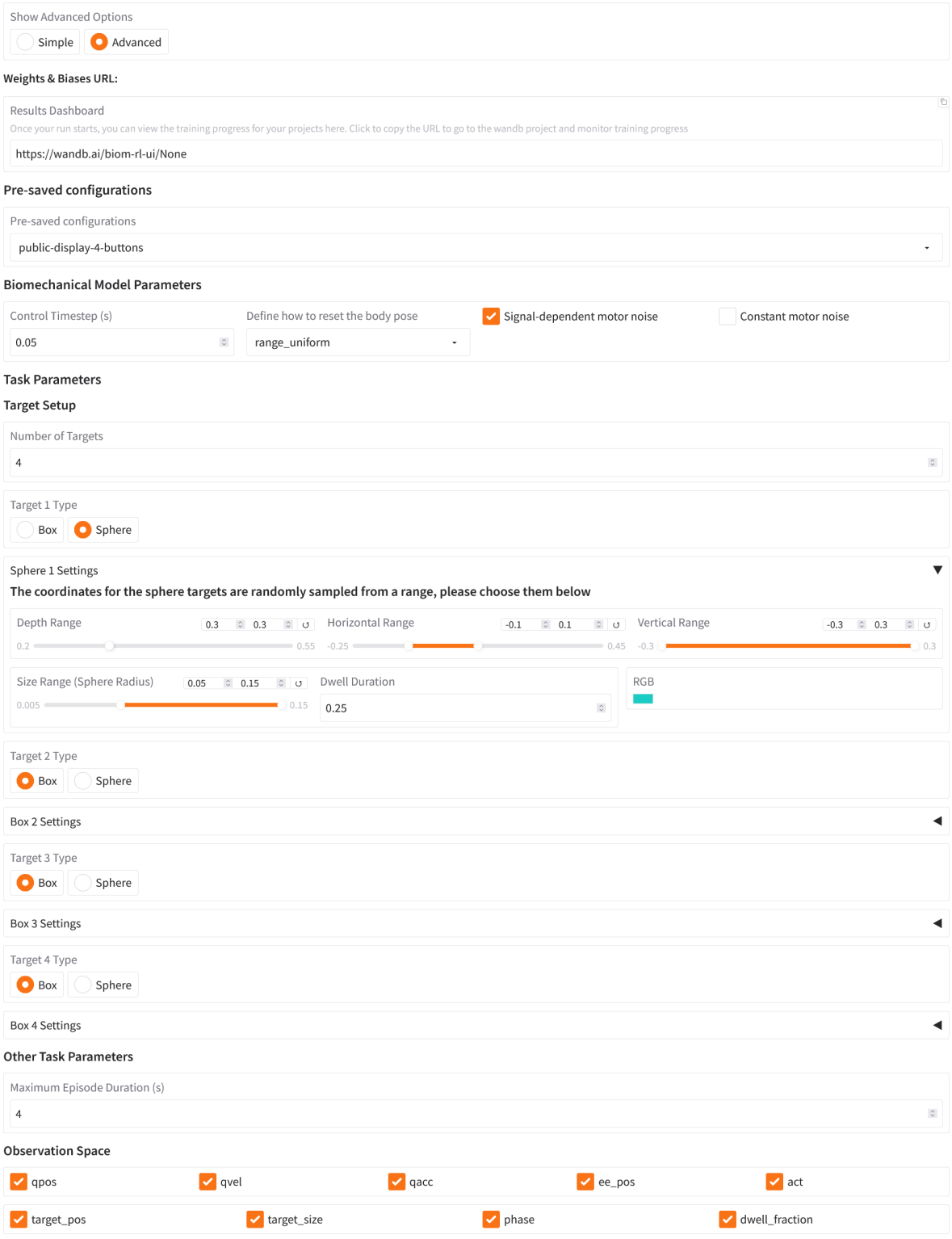


Figure 11: Advanced version of our GUI for configuring reinforcement learning experiments, adjusting task and reward parameters, rendering the environment, and running training sessions. In addition to the settings present in Simple Mode, this version enables modifying biomechanical properties, observation space components RL hyperparameters, and initialising the training with previously saved checkpoints (continued on next page).

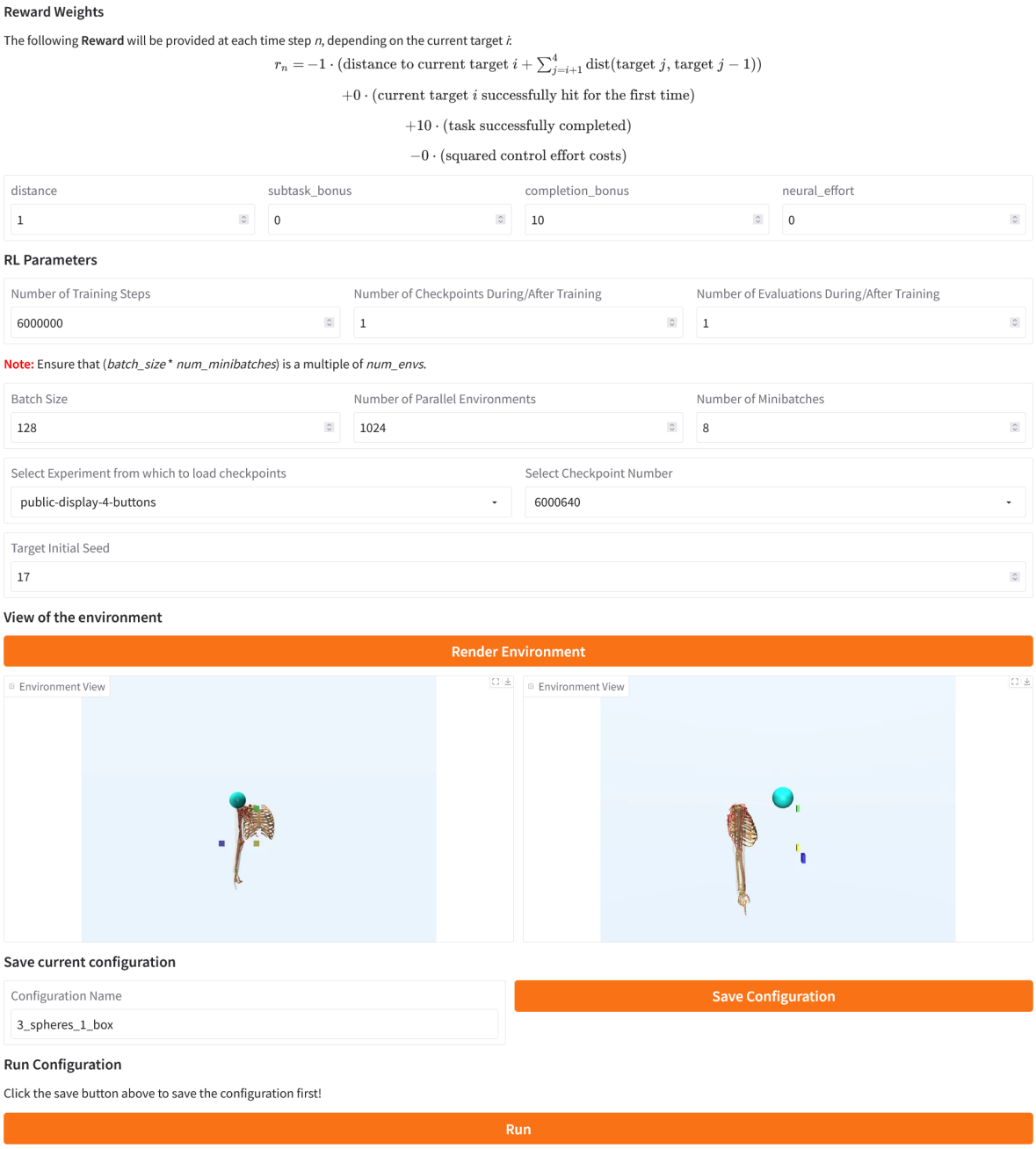


Figure 11: Advanced version of our GUI (continued from previous page).