Moon, Hee-Seung; Liao, Yi-Chi; Li, Chenyu; Lee, Byungjoo; Oulasvirta, Antti

# Real-time 3D Target Inference via Biomechanical Simulation

*Published in:*
CHI '24: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems

*DOI:*
10.1145/3613904.3642131

Published: 11/05/2024

*Document Version*
Publisher's PDF, also known as Version of record

# Real-time 3D Target Inference via Biomechanical Simulation

Hee-Seung Moon
Aalto University
Finland
hee-seung.moon@aalto.fi

Yi-Chi Liao
Aalto University
Finland
yichi.mdp@gmail.com

Chenyu Li
Aalto University
Finland
chenyu.li@aalto.fi

Byungjoo Lee
Yonsei University
Republic of Korea
byungjoo.lee@yonsei.ac.kr

Antti Oulasvirta
Aalto University
Finland
antti.oulasvirta@aalto.fi

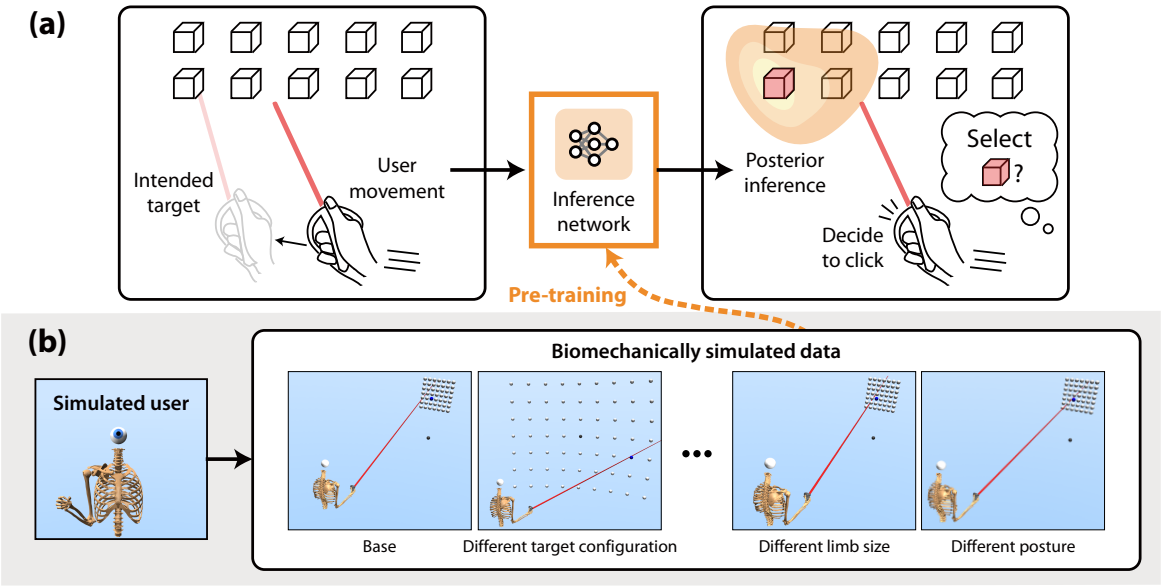


**Figure 1: We present a novel simulation-based target inference approach. In contrast to the existing data-based methods that use human data for training, our inference model is trained with a large and diverse amount of realistic simulated motions: (a) A user's target selection can be assisted by an inference network that proactively infers the user's intended target from their prior movements. (b) Our simulated user, based on a human biomechanical model, closely mimics user motion during target selection tasks, accommodating various task configurations and human motor variations.**

## ABSTRACT

Selecting a target in a 3D environment is often challenging, especially with small/distant targets or when sensor noise is high. To facilitate selection, target-inference methods must be accurate, fast, and account for noise and motor variability. However, traditional data-free approaches fall short in accuracy since they ignore variability. While data-driven solutions achieve higher accuracy, they rely on extensive human datasets so prove costly, time-consuming, and transfer poorly. In this paper, we propose a novel approach that leverages biomechanical simulation to produce synthetic motion data, capturing a variety of movement-related factors, such as limb configurations and motor noise. Then, an inference model is trained with only the simulated data. Our simulation-based approach improves transfer and lowers cost; variety-rich data can be produced in large quantities for different scenarios. We empirically demonstrate that our method matches the accuracy of human-data-driven approaches using data from seven users. When deployed, the method accurately infers intended targets in challenging 3D pointing conditions within 5–10 milliseconds, reducing users' target-selection error by 71% and completion time by 35%.



*CHI '24, May 11–16, 2024, Honolulu, HI, USA*

ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3642131

## CCS CONCEPTS

• **Human-centered computing** → **Pointing devices**; **Interaction techniques**; • **Computing methodologies** → **Machine learning**; **Modeling and simulation**.

## KEYWORDS

Target selection, target inference, biomechanical simulation, amortized inference.

**ACM Reference Format:**

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 (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3613904.3642131

## 1 INTRODUCTION

Selecting a target is a fundamental task in human–computer interaction. In traditional desktop environments, users frequently engage in target selections by using a mouse pointer to navigate dense menus with high efficiency and accuracy. In contrast, fast and accurate target selection in virtual- and augmented-reality (VR/AR) environments remains challenging, because of several factors: i) sensor limitations causing imprecision and lag [73, 90], ii) the absence of haptic feedback [88], iii) complications related to depth perception [83, 98], and iv) inherent noise in motor behavior [20, 86]. Prior studies show that target-selection performance in VR is particularly difficult when targets are small or distant [5, 55].

*Target inference* is the problem of identifying user's intended target before the cursor arrives at the target, using as input sensor data gathered during movement. The inference can inform assistive mechanisms, for expedited target selection [3, 31, 60, 65, 93, 100]. However, accurate target inference is not straightforward. The main challenge arises from the inherent variability of human movement. When selecting a given target, users differ in their trajectories toward it in response to their preferences (e.g., prioritizing speed vs. accuarcy of selection), biomechanical factors (strength, limb lengths, posture, etc.), and contextual factors. Even a single user selecting the same target twice exhibits variability.

Previously, target-prediction methods have focused on user motions' endpoints, representing potential endpoints for each target through Gaussian models [4, 30, 89, 102]; likelihood-based inference techniques are then applied that inversely infer the target from the endpoints. However, this approach, by excessively simplifying human motion, compromises accuracy in capturing users' intentions, particularly from high-variability motions. More recently, deep neural networks have been trained to predict the intended targets from trajectory data in a supervised manner [18, 49]. Successes notwithstanding, this approach can be heavily dependent on extensive training datasets collected from humans. Inadequate training data can lead to poor inference performance when used for new users or conditions. Therefore, a varied user pool is needed for capturing variability within the population.

Can we generate substantial and realistic movement data to train accurate inference models without involving human participants? In this paper, we introduce a novel target-inference method that employs simulation, grounded in a *biomechanical model*, to generate realistic *human motion priors*. Our key novelty lies in leveraging simulators to generate training motion data, mimicking the complexity and variability of human movements. We exploit a natural assumption: users' movements align more closely with biomechanical optima than with random motions. By estimating these optima through biomechanical simulation, we enable model-based inference that by design accounts for human-like variations in body posture, size, motor noise, etc.

Our method constructs a simulated user capable of visually perceiving the task environment as humans do and performs motor actions in alignment with human kinematic joint movements. Accordingly, we obtain a control policy for the simulated user that captures rational decision-making at every timestep, ultimately reproducing the human target-selection behavior. The simulated users permit gathering high-volume motion data while incurring little cost. So that the data reflect the full spread of human behaviors, our process considers various physical attributes (e.g., motor noise) and preferences (e.g., desired speed–accuracy tradeoff). Systematically altering the settings for these attributes lets us generate a rich set of trajectories. These trajectories are then used to train a neural proxy model that identifies the probability distribution of intended targets in light of the observed trajectory thus far. The model thus derived infers the target in milliseconds probabilistically. Finally, deploying the inference model aids in users' target selection in 3D environments, in real-time.

Our simulation-based target-inference approach offers clear benefits. Relative to pre-existing data-driven approaches, this method does not require gathering human data from the real world, so it affords higher efficiency, scalability, and significantly reduced costs. The method adapts to new task environments such as different arrangements of target objects or new interaction techniques. Furthermore, our model specifies its confidence in the inferences. That allows the target-selection assistance technique to ascertain the optimal moment to assist users in selecting the most likely targets while minimizing any adverse effects if the inference is not a high-confidence one. This is a crucial advantage over heuristics-based approaches, like proximity-based techniques, where uncertainty information is often ignored.

We evaluated three key aspects of our method experimentally: i) the quality of our simulator's motion replication, ii) inference performance with human data, and iii) improvement in users' target selection when the inference methods are deployed in assistance techniques. In a VR setting using raycasting-based selection, our simulator faithfully replicated human users' performance dynamics for different levels of selection difficulty (Study 1). The inference network, trained solely on the simulated data, infers users' intended targets within 5–10 ms per timestep. Each inference process operates using the partial trajectory data observed from the beginning of each trial, without requiring knowledge of the trajectory's total length. With human trajectory data, the network achieved an accuracy of 88% when it observed the first 80% of each trial (Study 2). We also showed human-data-driven approach's performance significantly depends on the volume of training data: to achieve accuracy levels similar to or higher than ours, a minimum of seven users, each providing 250 trials, was required. This inference process improved target-selection performance considerably (Study 3): when

targets are densely arranged, human users were 71% more accurate and 35% faster than with naive selection, and accuracy was 10% higher than with pre-existing forms of heuristic assistance. While our method's accuracy was comparable with the heuristic baseline, it enhanced user performance by making use of the confidence estimates provided by the network.

To sum up, this paper presents three main contributions. We release our dataset and code as open-source.[1]

(1) **A simulation-based target-inference method:** To the best of our knowledge, this is the first paper to train a target-inference model using synthetic data from biomechanical simulations. Sharing our end-to-end implementation and its evaluation offers valuable insights for this line of research.
(2) **Realistic simulation of target-selection motion:** Unlike previous efforts to generate end-point predictions, our approach replicates human-like motion with bodily variability during target selection by employing biomechanical models.
(3) **Demonstration of efficacy in target-selection assistance:** Our approach improves selection techniques by leveraging the inference outputs. With the high-speed inference, it accommodates rapid visual fresh rates of VR environments. We empirically show that integrating our inference into VR selection techniques significantly enhances user performance.

## 2 RELATED WORK

### 2.1 Techniques Facilitating Target Selection

Reducing the burden on the user in target selection can improve overall efficiency of tasks with an extensive range of interfaces, from traditional desktop ones [31, 57, 93, 100, 103] and touchscreens [7, 53, 94] to immersive VR systems [2, 5, 32, 55, 97]. Researchers following the principles of Fitts' law [26] have attempted to decrease the Index of Difficulty by enlarging targets [59, 60] or the cursor's interactive area [15, 31, 65, 93], for more efficient selection processes. Others modify transfer functions for quicker cursor movement [3, 9, 93, 100] or introduce shortcuts during approach movements [1, 53].

Effective facilitation techniques require accurately predicting users' intended targets, however. The traditional procedure relies on proximity-based heuristics [1, 31]. These often identify the closest target as the one intended. A more complicated form is Bubble Cursor [31], whose interactive area (bubble radius) varies dynamically with the context Lu et al. [55] have expanded this concept for 3D selection tasks. In high-target-density interfaces, the proximity-based "nearest neighbor" strategy inevitably proposes many wrong targets, causing unwanted distractions [97]. This shortcoming led to algorithmic attempts to improve motion end-point predictions by relying on the observed fractions of trajectories [3]; e.g., Lank et al. [48] predicted the pointing target by quadratic extrapolation of the cursor velocity based on observation. However, the algorithms often fall short of grasping the vast variability in human behavior.

Recent efforts have turned to neural networks. They process multiple channels of information (cursor [8] and hand motions [18, 38, 49], gaze [39], etc.) for more accurate evaluation of intentions. Recurrent neural networks [18, 95] have demonstrated effective handling of sequential data for prediction of user intention, with meta-learning techniques [64] further enhancing the model's ability to make efficient personalized predictions. These human-data-driven approaches all face a great obstacle, though, in the labor-intensive data collection required, both initially and often in light of new task conditions. We sought to address this challenge by using simulation-based data to facilitate target selection.

### 2.2 Biomechanical Simulation of User Motion

Data-driven methods improve inference of human intentions by utilizing extensive human-motion datasets that capture both *intra-user* (differences in a single user's motions) and *inter-user* (differences across multiple users) variability. Our novel approach achieves precise inference by implementing realistic motion simulation that has two following features: 1) utilizing a state-of-the-art human biomechanical model [77] and physics engine [84] to guarantee coherent bodily movements that honor human physical constraints and 2) biomechanics-informed replication of human motion's variability. To address intra-user variability, which arises partly from motor noise during muscle/joint actuation [52, 58, 87], we modeled motor control's constant and signal-dependent noise both [78, 86], sensitized to the latter's recognized role in the speed/precision compromise inherent to motion [36]. Tackling inter-user variability involves diverse limb-joint configurations, reward formulations, and motor-noise levels.

One way to address a user's goal-directed behavior with biomechanics is to frame it as an optimal-control problem [24]. Following the assumption that users aim to minimize internal costs (e.g., jerk of the end effector) when pursuing their goals, this optimization utilizes feedback from visual perception, proprioception, and other sensory channels. While classical closed-loop optimal-control techniques, such as linear-quadratic-Gaussian (LQG) control and model predictive control (MPC), have served simulation of human motion in HCI [25, 46, 58, 74], the computation required at each timestep for motion optimization renders their use with high-dimensionality models impractical. This constraint has prompted a shift toward deep reinforcement learning (RL). Through RL, the control policy (which, given sensory input, selects optimal actions) is derived as a deep neural network. Applying this paradigm in RL-driven biomechanical simulations dovetails with the emerging user-modeling framework, *computational rationality* [17, 22, 43, 44, 69]. Such simulations have already proven effective in modeling mid-air pointing [16, 24], keyboard use [37], jumping [42], gait [51], and a suite of interactive tasks [40] addressed by Ikkala et al. These foundations supported our work to develop biomechanical simulation for inferring user-intended targets via realistic motion data.

### 2.3 Probabilistic Inference with User Simulation

We also incorporate probabilistic inference to minimize risks of inference errors by accurately estimating the probability distribution for relevant variables [103]. This facilitates intelligent target-selection assistance; for instance, the system might offer shortcuts only when its predictions pass a certain confidence threshold [97]. Especially in traditional settings, Bayesian inference commonly serve such probabilistic reasoning [4, 30, 102, 103]. Informed by prior factors such as use frequency, it link users' actions to likely targets, such as intended buttons [96, 102] or words [28, 30]. However,

[1] https://github.com/hsmoon121/3d-target-inference

this approach is available in models only where the user actions and targets can be easily paired through likelihood functions. One common approach is to model the endpoints corresponding to individual keys by using simple Gaussian distributions [4, 102]. Ziebart et al. [103] exploited a simple linear relationship between 2D interface states and user cursor actions to estimate a target's posterior distribution from partial cursor trajectory.

The complexity of today's computational models for 3D pointing (e.g., arising from hierarchical structures with RL-based policies [16, 24, 40]) complicates applying traditional forms of probabilistic inference. Against this backdrop, likelihood-free inference [19], which employs iterative simulations to identify the most plausible parameter distribution that could account for the behaviors observed, represents a viable alternative. Conventional forms of these methods, such as approximate Bayesian computation [6, 34], are hampered by a need for substantial computation power and time (often hours to days [45, 62]). Recently introduced *amortized inference* techniques [19, 29, 75] appear more promising: Modern machine-learning approaches enabled real-time variational approximation of complex probability distributions. They used a neural proxy model that effectively maps observed behaviors to an approximate posterior distribution of the parameters. This approach has already enhanced inference process with several HCI simulation models [63], delivering inferences in tens of milliseconds. We extend it to real-time 3D target inference, addressing key challenges such as real-time deployment, data discrepancy between simulation and humans, and user variability.

# 3 SIMULATION-BASED TARGET INFERENCE

We formulate the target-inference problem as identification of the posterior distribution of the user-intended target point by considering the ongoing trajectory of the end effector (in essence, to the on-screen cursor/pointer). Our method is flexible and suited for environments where pointing is done through human motion alone or with devices like VR controllers. The key steps of our method can be summarized thus:

(1) **Biomechanical simulations:** The first step constructs *computational agents* that, bounded by human biomechanical constraints, simulate realistic human motor behavior for the intended interaction. Dynamically adjustable parameters for several latent factors (such as limb length, the noise of motor control, and kinematic constraints) account for intra- and inter-user variability as the agent generates human-like motion toward various targets. The *action policy*, governing the perceptual control of biomechanics in the interactive tasks specified, is obtained through RL with the agents pursuing maximal utility analogously to how humans do.
(2) **Training the inference model with the simulated data:** At its core, our inference model is a deep neural network. Trained with the simulated data of the computational agent, it employs state-of-the-art density-estimation techniques to approximate probabilistic inference, thereby expediting the target-selection procedure.
(3) **Deploying the inference model to the end users:** Once trained, the inference model can compute posterior distributions of the predicted target position all in milliseconds. These distributions specify not only the most likely target but also a confidence level that can inform the system's decision on when to provide assistance.

## 3.1 Step 1: Biomechanical Simulations

We assume that humans' target-selection behavior unfolds as a sequence of decisions. At each timestep, the decision continually refines the action in light of real-time sensory feedback (e.g., on the distance between the target and the end effector). Our agent emulates this complex dynamic through a computationally rational agent's decision-making [69]. Concretely, the agents perceive the interactive environment through vision and proprioceptive feedback. Then, the action policy determines the action, which gets translated into movement through biomechanical models (see Figure 2(a)). This can be formulated as an RL problem within a partially observable Markov decision process, or POMDP.

Our focus in this paper is on human upper-limb interaction. Humans' upper extremities are typically characterized by seven degrees of freedom: three in the shoulder (elevation plane, shoulder elevation, and shoulder rotation), one in the elbow (elbow flexion), and three in the wrist (forearm rotation, wrist flexion, and wrist deviation). We chose an implementation of the *Upper Extremity Dynamic Model* [77], which recent research has exploited extensively to simulate human interaction — with actuation either directly at the joints [24, 37] or through the tendons [40]. In contrast to conventional linked-segment models with their basic skeletal framework, biomechanical models provide physiologically accurate joint movements with inter-segmental coupling and empirically derived angle and torque limits. For integration with RL, we employ a biomechanical model converted for use with the computationally efficient physics engine MuJoCo [40, 84].

Below, we present the RL problem formulation that captures the agent's target selection in an interactive task environment, then introduce the settings that permit realistic motor variation.

*3.1.1 RL formulation.* Within the POMDP framework[2], an agent performs an *action* based on its current *observation*, which encompasses only partial information on the full task state. In consequence of the action, the agent receives a *reward*, alongside a new observation, from the updated state. The following key components characterize our setting:

- *Observation:* The agent's observations of the task state come from two primary types (inspired by prior work [40]): visual and proprioceptive. A forward-facing eye 20 cm above the agent's neck captures visual feedback, as $180 \times 120$ RGB-D images of the environment in front of it, and the proprioceptive feedback encompasses information on each joint's rotational angle, angular velocity, and acceleration.
- *Action:* Our action space comprises: 1) seven action commands of actuating corresponding *joints* and 2) a command for click decision, both with ranges of -1 to 1. The action commands for each joint directly determine the torque applied to each joint, scaled for the respective biomechanics limits; this is inspired by the setting of Hetzel et al. [37],

[2]We refer the reader to Sutton and Barto [82] for the general formulation of POMDP and RL problems.

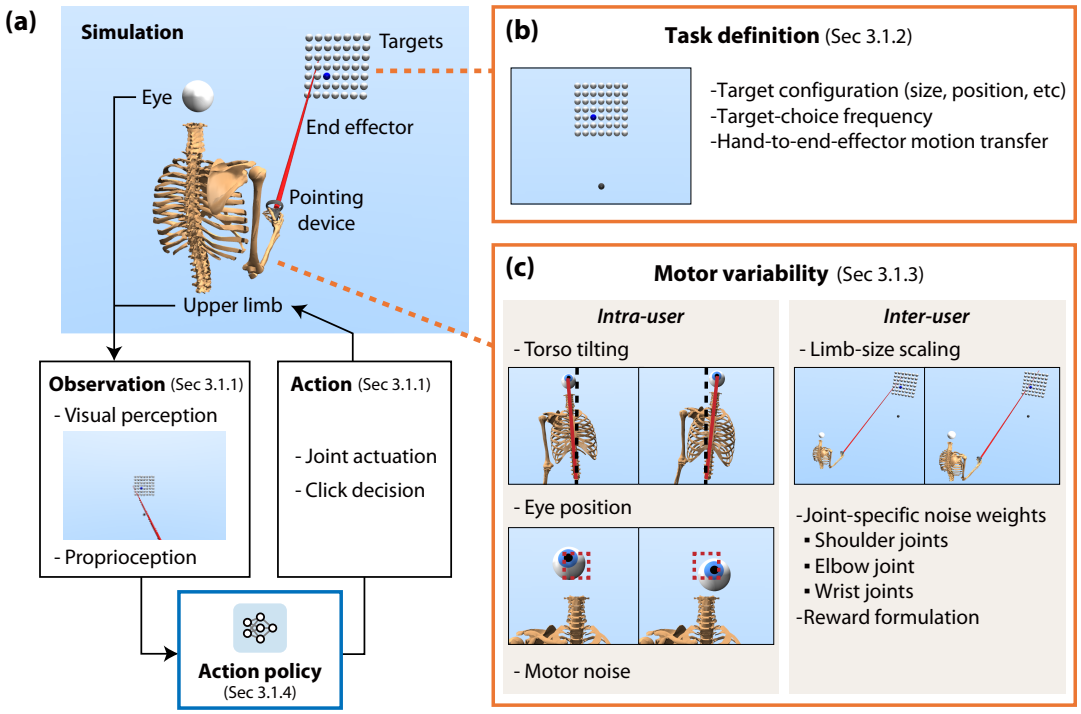


**Figure 2: (a) Our biomechanical simulation involves the complete perception–action loop, from observing the simulated environment to generating actions through a learned action policy. This simulation approach accounts for a set of latent factors that (b) define the target-selection task and (c) yield various human motor variations.**

which afforded more efficient training than muscle-based actuation. The click decision command triggers the simulated user's click after applying random time noise (as simplified implementation of prior models of click timing [71]).

- *Reward:* Each target selection can have its own task-specific reward formulation shaping the agent's behavior strategy. This reward structure's weighting for the agent's selection success/failure, elapsed time, and motor effort ultimately influences tradeoffs (e.g., prioritizing successful selections over speed or fatigue factors). From among the various means of evaluating optimal motor effort, we opted for a well-established and simple measure: jerk (change in acceleration) at the end effector [27, 85].

*3.1.2 Interactive task.* An interaction mechanism on top of the biomechanical model specifies how upper-limb movements translate to end-effector movements. For instance, in VR, raycasting techniques are commonly used to map the hand's orientation to a ray-style cursor. Meanwhile, transfer functions specific to indirect pointing devices (mice, trackpads, etc.) mediate the cursor's on-screen position. Also crucial is addressing the target's configuration with other onscreen elements, which entails specifying target sizes and positions that match real-world use cases while simultaneously considering distractors' possible exacerbation of task difficulty.

*3.1.3 Latent factors for motor variability.* Our model captures a broad spectrum of latent factors that contribute to both intra- and inter-user variation. Table 1 provides an exhaustive list of the components our research covered.

- *Intra-user variability:* Within-individual variations arise from two sources: motor noise and posture shifts. We model motor noise via both signal-dependent and constant components. In our control system, the action, $\boldsymbol{a}$, is influenced by noise added to the agent's decision $\boldsymbol{a}^*$ thus:

$$\boldsymbol{a} = \min\left(\max\left(\boldsymbol{a}^* \cdot (1 + \epsilon_{sig}) + \epsilon_{con}, -1\right), 1\right),$$

  where $\epsilon_{sig}$ is the signal-dependent and $\epsilon_{con}$ the constant noise term. It samples both from Gaussian distributions with a mean of 0 and different standard deviations (0.103 and 0.185), following van Beer et al.'s example [86]. Several mechanisms account for natural postural deviations not included in the biomechanical action space: in each trial, we randomly sample 1) the eye position, for perturbations to eye–hand separation caused by neck-tilting, and 2) torso tilt (while the spine is kept fixed), for considering variations that might arise from changes in body posture.
- *Inter-user variability:* A parameterized simulation model permits simulating user-to-user physical differences and representing user preferences. A parameter for limb scale gets applied first, adjusting the overall kinematics relative to the external environment; next, noise-scaling factors are added

**Table 1: A list of the latent variables accounted for to address both intra- and inter-user motor variability.**

| Notation | Meaning | Type | Distribution |
|---|---|---|---|
| $\epsilon_{sig}$ | Signal-dependent motor noise term | Intra-user | $\mathcal{N}(0, 0.103^2)$ |
| $\epsilon_{con}$ | Constant motor noise term | Intra-user | $\mathcal{N}(0, 0.185^2)$ |
| $\delta_{eye}$ | Deviation in eye position from the upright point (m) | Intra-user | $\mathcal{U}([-0.02, 0.02]^3)$ |
| $\phi_{tor}$ | Angular deviation of the torso from vertical angle (°) | Intra-user | $\mathcal{U}([-1, 1]^3)$ |
| $s_{limb}$ | Global scaling coefficient for limb sizes | Inter-user | $\mathcal{U}([0.85, 1.10])$ |
| $\sigma_{sho}$ | Coefficient for scaling the noise for shoulder joints | Inter-user | $\mathcal{U}([0.25, 1.25])$ |
| $\sigma_{elb}$ | Coefficient for scaling the noise for the elbow joint | Inter-user | $\mathcal{U}([0.25, 1.25])$ |
| $\sigma_{wri}$ | Coefficient for scaling the noise for wrist joints | Inter-user | $\mathcal{U}([0.25, 1.25])$ |
| $w_{fail}$ | Penalty coefficient for failed selections | Inter-user | $\mathcal{U}([0.1, 1.0])$ |

for each joint (shoulder, elbow, and wrist), to capture its motor-precision variations; and, finally, we adjust the penalty for unsuccessful selections (a weight parameter for reward formulation), to reflect the cautiousness behind each user's decision on clicking.

*3.1.4 Policy training.* We utilize proximal policy optimization (PPO) [79] to optimize the neural-network-based action policy of the agent. This deep RL algorithm is suitable for tasks with continuous action spaces, contributing to its widespread use in human-modeling research [40, 43]. Specifically, we engineer the policy network to accept given user-specific free parameters ($s_{limb}$, $\sigma_{sho}$, $\sigma_{elb}$, $\sigma_{wri}$, $w_{fail}$) along with the observation variables. By optimizing the policy network across episodes featuring diverse user parameter values, we develop a generalized action policy for the agent that accommodates a wide range of user attributes [47, 62, 63].

## 3.2 Step 2: Training of the Inference Network

We employ neural density estimation [21, 75] to obtain the posterior distribution for the intended target position from observed user trajectories (Figure 3). Recently published work [63] inspired us to extend the method for efficiently inferring not just the free parameters of simulation models (e.g., characteristics of the simulated user) but also the exact positions of intended targets. This broadening of focus is justified in that the target positions can be viewed as a form of parameter, one representing the task environment in each trial. Accordingly, the same density-estimation techniques can be applied for our aim.

The core strength of our inference network lies in its ability to extract essential information from input data to accurately represent complex probability distributions beyond simplistic assumptions such as Gaussian models'. Here, the input data $\boldsymbol{y}$ include not just the trajectory of the end effector's 3D position but also the size and position details of interactive objects (potential targets) within the task environment. The output is a posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$, where $\boldsymbol{\theta}$ represents the intended target position. To generate this complex distribution computationally, our inference network employs normalizing flows [21, 70, 76]. Starting with a basic normal distribution, it applies a series of bijective transformations, each modeled by a neural network and conditioned on the input data $\boldsymbol{y}$. These steps progressively shape the distribution into more intricate forms, approximating $p(\boldsymbol{\theta}|\boldsymbol{y})$. Additionally, an encoder network can preprocess the input data before feed-in to the normalizing flows. This encoder network can range from simple multi-layer perceptrons to Transformers or other advanced architectures suited to handling time series or multiple trials. Descriptions elsewhere provide further implementation and training details [63].

Training the inference network relies on a simulated dataset composed of pairs of target positions $\boldsymbol{\theta}$ and corresponding synthetic observations $\boldsymbol{y}$. Factors such as the locations where targets spawn and their frequency of being chosen for targeting can influence this prior. For instance, user commands in menu-selection tasks may show a bias toward specific items [23] while word and letter frequency influence presses in keyboard interfaces [28]. These variations in the prior distribution inevitably affect the posterior that the network learns, in line with Bayes' theorem.

## 3.3 Step 3: Deployment for User Assistance

Once trained, our inference model generates posterior distributions of the target positions in light of the given portion of the user's trajectory. The operation, conducted via a single forward pass through the neural network, takes mere milliseconds. Importantly, this probabilistic distribution provides more than the most probable target; it also assigns a confidence value to the prediction. Consider an interface populated with $N$ selectable objects, each at position $\boldsymbol{\theta}_i$, for $i = 1, \ldots, N$. For a given observed input $\boldsymbol{y}_t$ at timestep $t$, the most probable target $i_t^*$ is identified thus:

$$i_t^* = \text{argmax}_i \, p(\boldsymbol{\theta}_i|\boldsymbol{y}_t)$$

The procedure for calculating the exact $p(\boldsymbol{\theta}_i|\boldsymbol{y}_t)$ by means of the normalizing flows is detailed in Supplement A. The confidence level ($C_t$) denotes the certainty ratio for the most likely target $i_t^*$:

$$C_t = \frac{p(\boldsymbol{\theta}_{i_t^*}|\boldsymbol{y}_t)}{\sum_{i=1}^{N} p(\boldsymbol{\theta}_i|\boldsymbol{y}_t)}$$

Accordingly, $C_t$ equips us with a probabilistic metric for the trust we can place in the model's prediction at the moment in question. As Figure 3(b) illustrates, the confidence level rises over time as the inferred posterior distribution narrows its focus to the correct target. With our inference approach, the related information is accessible at each timestep with minimal lag (~10 ms). This permits ready integration of the confidence measurement into existing systems, enhancing target selection processes in real time [53, 97].

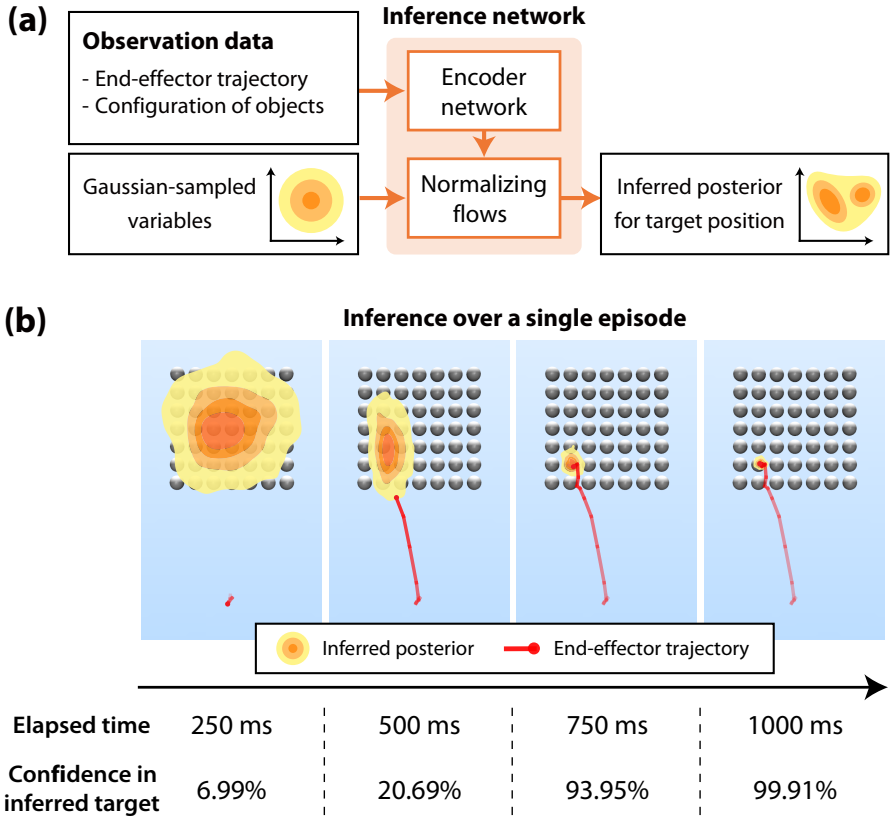


**Figure 3: (a) Our inference network derives the posterior distribution of the target position from observed user motion. (b) With the inferred posterior, the system not only identifies the most probable target but also provides a confidence level for that target, in real time (5–10 ms). The above posteriors are based on a human participant trajectory collected in Study 1.**

## 4 OVERVIEW OF STUDIES

Our method is composed of three key steps. To fully validate our approach, our evaluation is also comprised of three distinct studies, each corresponding to one step in the method. Our evaluation of validity focused on raycasting-based pointing, which is a representative and ubiquitous target selection method that can be found in a wide range of VR/AR applications. Together, these efforts cover the full implementation and validation process, from building the biomechanical simulator to training and deploying our inference network in end-user target-selection scenarios.

- **Study 1 (Evaluating the simulator)**: We verified how well the simulated motion replicates the motions of human users. We first developed a simulator for the raycasting-based target-selection task, which allows us to gather simulated motion data. Then, we gathered human participants' motion data for the same selection task. Finally, we compared data from two sources.
- **Study 2 (Evaluating the inference)**: Next, after training the inference network on the simulated dataset, we evaluated the accuracy and efficiency of the inference network in inferring the target from human participants' motion data.
- **Study 3 (Evaluating the enabled assistance)**: We deployed the trained inference network and utilized its inference to assist target selection. Our method was designed to offer selective suggestions, displaying the inferred results only when the inference was deemed reliable. We evaluated how this approach improved the human users' speed and accuracy in selecting targets.

### 4.1 Task: Raycasting Selection

Raycasting has become established as a standard technique for interacting with objects in VR [2, 5, 55, 61, 92]. It employs a cursor that resembles a stare emanating from a controller, whereby users can engage with distant objects. For simplicity, our task setting assumed that all interactive objects are positioned on a spherical surface, consistently at five meters from the user's eye level. This setup mirrors a typical VR scenario in which interface elements are arranged on a single plane, for minimal occlusion. Accordingly, the position of the end effector here is determined by the point at which the ray and the surface intersect. The user's objective is to trigger a click when the end effector is within the target area.

*4.1.1 Task configuration and procedure.* We implemented a target-selection task described by Lu et al. [55]. This task comprises a grid

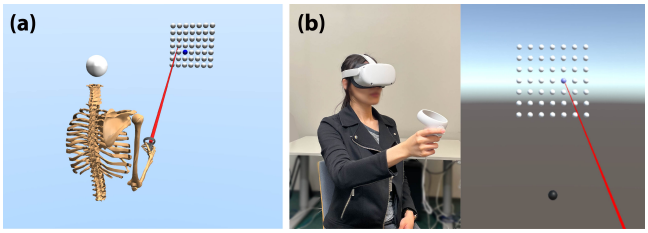


**Figure 4: Overview of studies: (a) We developed a simulator to replicate user behavior during VR target selection tasks, and trained an inference network using the simulated dataset. (b) We then gathered motion data from participants performing the same task using the Meta Quest 2 device. This data was used to evaluate both our simulator (Study 1) and the inference network (Study 2). Finally, we tested our inference-based assistance in target selection scenarios with human users (Study 3).**

containing spherical objects where one (colored blue) is designated as the target while the others (colored white) serve as distractors. We set two distinct grid configurations (*Dense* and *Wide*) and two target sizes (*Large* and *Small*). As Figure 5 shows, the *Dense* configuration represents a scenario with densely arranged objects, with a $7 \times 7$ grid whose spacing between objects is a visual angle of 1.44°, and the *Wide* configuration disperses the objects across the user's entire field of view, with a $9 \times 7$ grid that has 6° spacing. Target size is either *Large* (width: 0.10 m, visual size: 1.15°) or *Small* (width: 0.06 m, visual size: 0.69°). To modulate selection difficulty target-specifically, we established a consistent beginning point by means of a starting object. In this setting, users initiate a trial by directing the end effector through the starting object, after which the *selection target* — the target that the participant should select — is indicated (in blue, as opposed to white) on the grid. The starting object is positioned either below 13.5° from the grid center for the *Dense* type or at the center for the *Wide* type. The width of the starting object is 0.10 m (1.15°). We followed the principles established by Lu et al. [55], whereby each selection target must have four adjacent distractors. Since a target in the outermost layer or adjacent to the starting object is not surrounded by four distractors, it is not chosen as a selection target. This left 25 potential targets for *Dense* and 26 for *Wide*. For each trial, we sampled the target uniformly from the candidate targets.

*4.1.2 Transfer to simulation.* We implemented the identical target selection task environment in MuJoCo for simulation. Our simulated agent has a 3D model with a VR controller (Meta Quest 2) attached to its right hand, which serves as the origin of the ray projection. Hence, the upper-limb movements dictate the ray's direction and origin, thereby determining end-effector position. We set the decision-making interval to 50 ms. We defined the reward formulation for the task such that the simulated agent receives a reward signal at each timestep $t$, denoted as $r_t$, as follows:

$$r_t = \begin{cases} w_{success} - w_{effort} \cdot \|\boldsymbol{j}_t\|^2, & \text{if click is successful} \\ -w_{fail} - w_{effort} \cdot \|\boldsymbol{j}_t\|^2, & \text{if click is failed} \\ -w_{time} - w_{effort} \cdot \|\boldsymbol{j}_t\|^2, & \text{otherwise} \end{cases}$$

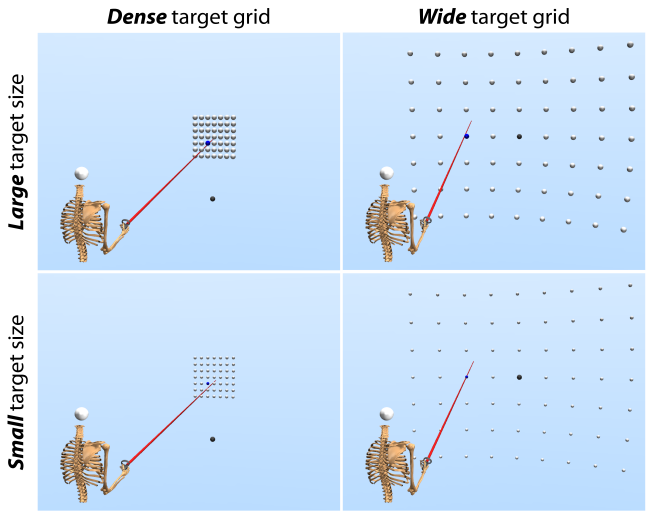


**Figure 5: Four target configurations factored by grid configuration (*Dense* or *Wide*) and target size (*Large* or *Small*).**

The reward coefficients, $w_{success}$, $w_{fail}$, $w_{time}$, and $w_{effort}$, correspond to the success, failure, elapsed-time, and motor-effort components, and $\boldsymbol{j}_t$ represents the timestep-specific jerk of the end effector, expressed in m/s$^3$. We chose the fixed settings $w_{success} = 10$, $w_{time} = 0.05$, and $w_{effort} = 0.0025$, while $w_{fail}$ is varied in line with sampled values as presented in Table 1. This reward formulation ultimately determines the simulated agent's strategy after convergence.

## 5 STUDY 1: EVALUATING USER SIMULATOR

A foundation of our target-inference method is the biomechanical simulation's capacity to replicate human users' motions faithfully under varying levels of selection difficulty. Study 1 validated this capacity through comparisons between the simulator-generated motions and human ones. We gathered data from participants performing the raycasting-based target-selection task. The task incorporated variations in target configuration (*Dense* and *Wide*) and sizes (*Large* and *Small*). Our simulator was achieved through RL (PPO [79]) in MuJoCo simulation, adhering to its formulation in Subsection 4.1. To expedite the learning process, we trained two distinct simulators for both the *Dense* and the *Wide* target configuration. The training took approximately 40 hours on a PC equipped with an Intel i9-13900K CPU and NVIDIA RTX 4090 GPU. See Supplement B for details.

### 5.1 Data Collection Method

*5.1.1 Participants.* Twenty participants were recruited (11 women and 9 men). Their age range is 21–45 (mean=26.2, SD=5.1). All participants had either normal or corrected-to-normal vision and were right-handed.

*5.1.2 Task.* The task and interface configuration were as presented in Section 4.1. Participants were instructed to select a specific target from among distractor objects in the VR environment (see Figure 4(b)). They had to point their ray at a fixed starting object,

**Table 2: Study 1: Comparison of human and simulated task performance across conditions. Our simulation's mean performance, by every metric and under all target conditions, fell within one standard deviation (SD) of the mean performance of each participant across the full set of human participants. These results are therefore highlighted in green.**

| Metric | Target Condition | | Mean of Human Data | SD of Human Data | Mean of Simulated Data |
|---|---|---|---|---|---|
| | Configuration | Size | | | |
| Completion time (second) | Dense | Large | 1.174 | 0.202 | 1.357 |
| | Dense | Small | 1.634 | 0.390 | 1.751 |
| | Wide | Large | 1.411 | 0.311 | 1.551 |
| | Wide | Small | 1.931 | 0.370 | 2.096 |
| Error rate | Dense | Large | 0.140 | 0.113 | 0.158 |
| | Dense | Small | 0.238 | 0.126 | 0.262 |
| | Wide | Large | 0.203 | 0.134 | 0.174 |
| | Wide | Small | 0.307 | 0.148 | 0.403 |

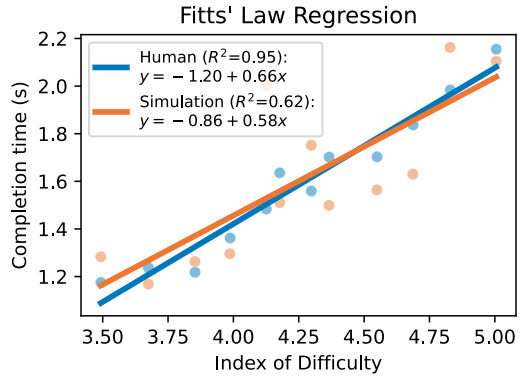


**Figure 6: Study 1: Our simulator's generated motion followed Fitts' law, closely mirroring the human participants' motion**

which activated a trial. Of the white objects in the grid, one object (the selection target) turned blue at the moment the trial began. When the end effector hovered over an object, that object turned light blue if it was the correct target and turned light green otherwise. A successful selection was accompanied by a tone, while an unsuccessful selection was indicated by a beep sound distinct from this. Each trial persisted until a successful selection was made. Participants were instructed to complete each trial "as quickly and accurately as possible."

*5.1.3 Study design and procedure.* The study employed a within-subject design with a 2 × 2 factorial structure: *Target Configuration* (*Dense* and *Wide*) × *Target Size* (*Large* and *Small*). We refer to each combination of Target Configuration and Target Size as a *condition*. Different conditions come with different levels of difficulty in the target selection.

All participants first signed the consent forms. Participants completed a practice block for each condition before the data collection, to familiarize themselves with all task conditions. Then, they went through eight sessions in the study proper, with two sessions per condition. The sequence of conditions was counterbalanced via a balanced Latin square design [12] to mitigate the influence of order effects and immediate carry-over effects. Each session comprised five blocks, and each block presented the participant with all possible selection targets in the trials (25 trials for *Dense*, 26 trials for *Wide*), appearing in a randomized order. Calibration was done before each block: the system measured the participant's eye level and then displayed the target grid at that height, to guarantee consistent positioning of the targets. Upon completion of each block, participants' fatigue levels were assessed on the Borg CR10 scale [11], a 10-point rating scale designed to quantify perceived human fatigue. Participants reporting fatigue levels of 6 or above were promptly granted breaks of at least three minutes to minimize the potential impact of fatigue. Also, participants were free to take additional rest breaks whenever needed. In all, each participant completed 1,020 trials (2 × 2 × 2 × 5 × (25 or 26)), with the full experiment lasting approximately an hour. The study adhered to the local protocols for ethics approval.

*5.1.4 Apparatus and implementation details.* Participants performed the task with a Meta Quest 2 at a 120 Hz refresh rate. The study software was implemented in Unity. Within the program, we tracked the trajectory of the end effector and recorded the execution of clicks for each trial at 50 ms intervals.

## 5.2 Results and Discussion

*Aggregated task performance.* Our simulator reached levels of task performance similar to humans' under varying conditions and differing levels of selection difficulty. In the four conditions, we generated a set number of trial data from our simulator and compared with human data, using two aggregated performance metrics: completion time and error rate. Completion time was measured from the moment a trial was initiated (i.e., the ray passing through the starting object) to the moment when a successful click occurred. The error rate was calculated as the ratio of the total number of unsuccessful clicks to the total click counts. Human participants exhibited longer completion times and higher error rates for more difficult selections; i.e., *Wide* configurations and *Small* targets introduced higher difficulty. This result is in line with Fitts' law (see Table 2 for the results). Using our simulator, we faithfully reproduced these dynamics. Simulated performance closely matched the

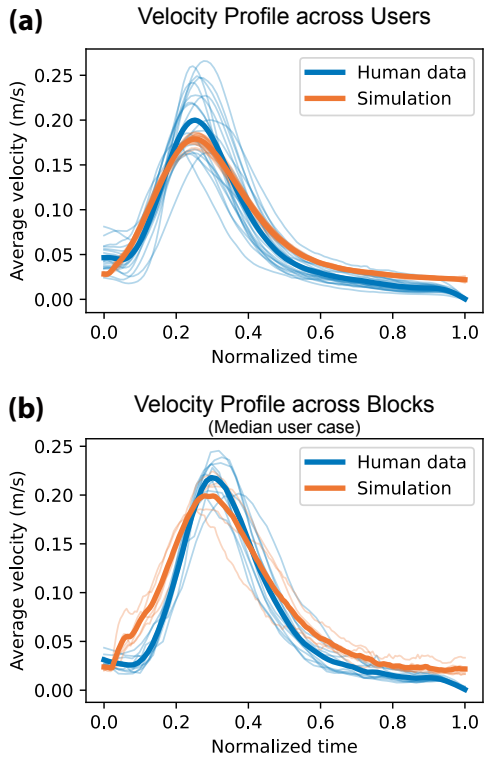


**Figure 7: Study 1: Our simulator faithfully replicates the intricate details found in motion trajectories, as evidenced by velocity–time functions. We normalized all movement times to a $[0, 1]$ range for easier comparison, with $0$ marking the start and $1$ the end of a movement. (a) The simulation closely matched the average velocity profile of individual participants. (b) With fixed user-specific parameters, the simulator accurately reproduced the variability in the velocity profile across an individual participant's blocks. The plot is from the participant with the median peak velocity across all participants.**

mean performance of participants in each condition, falling within one standard deviation of mean performance across all participants.

Our simulator consistently adhered to Fitts' law, faithfully reproducing the patterns observed in human participants' performance, even at a finer-grained level (see Figure 6). We binned all of the simulator's trials into 12 groups on the basis of the Index of Difficulty associated with each selection target's position (with equal-frequency binning). The analysis revealed a positive linear correlation between the completion time and the Index of Difficulty for each simulated point ($R^2$=0.62). This result is consistent with prior work [24, 40], which has demonstrated adherence to Fitts' law in biomechanical simulations of human pointing motion.

*Velocity profile.* The velocity–time functions summarize how the motion dynamics of users unfold over a trial [22, 66]. Figure 7(a) shows that our simulated trajectories replicate the overall velocity profile of human movements. Specifically, human and simulated users closely resemble each other in the magnitude of peak velocity and the normalized time at which this peak velocity was reached. Our simulator recorded a peak velocity of 0.179 m/s$^2$ at a normalized time of 0.252, on average. Both the magnitude and the time fall within standard-deviation range of the human data; the figures are $0.204 \pm 0.035$ and $0.254 \pm 0.008$, respectively.

Variability in velocity profiles is visible in the human data, both across users and within trials for a single user. Our simulator captures this complexity by sampling the latent variables listed in Table 1 for each individual trial (inter-user) or user (intra-user). For intra-user variation, the simulator closely mimics fluctuations observed from individual human users from one trial block to another. Specifically, the SD values for peak velocity and its occurrence time were 0.018 and 0.025, falling within the human-data range, at $0.025 \pm 0.009$ and $0.028 \pm 0.010$, respectively. Figure 7(b) showcases how our simulator replicated the intra-user variability of one participant, the one with the median peak velocity from among the 20 participants. As for inter-user variation, the simulator yielded SDs 0.005 and 0.008 for peak velocity and its timing, respectively, in contrast against the human data's values of 0.035 and 0.021. We discuss the factors that may have contributed to the higher inter-user variability observed in the human data in Section 8.

## 6 STUDY 2: EVALUATION OF THE INFERENCE MODEL

In the second study, we assessed the performance of our inference network, which was trained exclusively on simulated data from Study 1. Our inference model predicts the intended target by using any fraction of the trajectory and states a probabilistic confidence level for each prediction. With primary focus on investigating the accuracy and efficiency of the inference network as the trajectory progresses, we compared our inference model to three other approaches, including data-free and data-driven methods both. The baseline method that has the most potential to yield the best accuracy uses the same neural-network structure for inference but with training on *human motion data*, collected in Study 1. With this study, we also aimed to highlight the advantages of using simulated data over human data.

### 6.1 Experiment Method

*6.1.1 Evaluation data.* We evaluated inference performance utilizing the human-participant data from Study 1 ($N$=20). With each trajectory recorded at intervals of 50 ms, we extracted fractions from each trajectory at cumulative progression intervals, starting from 0–10%, and extending by 10% increments up to 100%.

*6.1.2 Inference methods.* We implemented four inference methods for the study, with our approach among them:

- *Nearest Neighbor*: Inspired by Bubble Cursor [31, 55], this method simply considers the object closest to the current end-effector position as the inference result.
- *Quadratic Regression* [48]: We adapted a method from Lank et al. Lank et al. [48] that predicts a trajectory's endpoint through quadratic extrapolation of the end-effector velocity. This approach does not offer probabilistic inference. For this study, we conducted 5-fold cross-validation, meaning that 16 of the 20 users were used for training data, while the

remaining four were used for testing, and this process was repeated five times.

- *Human-data-based Neural Inference*: We trained a baseline inference model, which applied the model structure employed in *Simulation-based Neural Inference*, using human data. This baseline represents the upper limit for our simulation-based approach's potential, as it captures human variability from authentic motion data. This approach enables probabilistic inference and offers a predicted confidence level for each inference. See Supplement C.1 for further details. As with Quadratic Regression, we used 5-fold cross-validation.
- *Simulation-based Neural Inference* (our approach): We trained an inference network based on the simulator constructed in Study 1. Similarly to Human-data-based Inference, this this method generates predicted targets and associated confidence levels. During the training, we sampled user-specific parameters (Table 1) for each trial and collected data accordingly. The entire simulation comprised approximately 65,000 trials, which took about two hours using the same PC as in Study 1. See Supplement C.2 for more details.

## 6.2 Results and Discussion

*Inference accuracy.* Figure 8(a) shows the accuracy of each inference method. Following the practice established by prior works [18, 103], we analyzed different method's accuracy at varying proportions of the observed trajectory from the onset, ranging from 10% to 100%. This allows for the assessment of comprehensive inference accuracy across trials with different durations due to varying target locations. Human-data-based Neural Inference consistently performed better than other methods; however, its advantage over Simulation-based and Nearest-Neighbor methods gradually became marginal as the end of a trajectory approached. Our Simulation-based Neural Inference, though slightly behind the Human-data-based approach, outperformed the Quadratic Regression method. Our method and the Nearest-Neighbor method showed overall comparable performance, with ours performing slightly better in the earlier stages and Nearest Neighbor doing slightly better in the final stage. Finally, Quadratic Regression consistently trailed behind the other methods; this is consistent with the literature, which has reported that it shows unstable performance [103].

Despite having similar accuracy to the Nearest-Neighbor method, Simulation-based Neural Inference offers the significant benefit of leveraging inference confidence. This approach enhances the system's ability to determine the optimal timing for using inferred results, leading to more effective assistance and reducing distractions from premature visualizations of targets [97]. Figure 8(b) illustrates the increasing confidence of two Neural-Inference methods as movements progress. Unlike these methods, the Nearest-Neighbor approach lacks a mechanism for accurately timing assistance.

*Inference efficiency.* Our method's neural inference demonstrated an average inference time of 5–10 milliseconds. Although this is slightly longer than the processing times of Nearest Neighbor or Quadratic Regression, which took less than one millisecond, our inference method still offers a remarkably high computation speed. This level of efficiency allows it to function in real-time scenarios, even with visual refresh rates of 120 Hz.

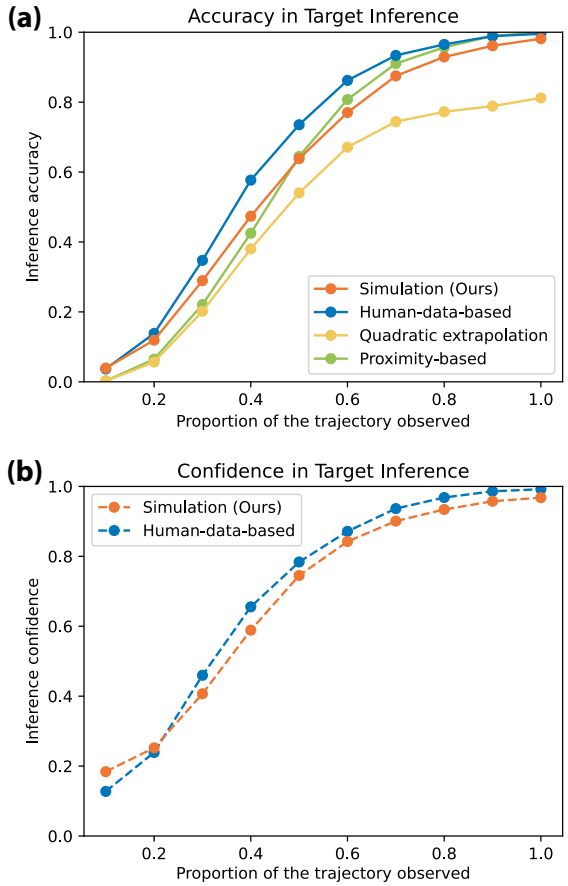


**Figure 8: Study 2: (a) Mean accuracy of intended-target classification, by inference method, as the proportion of the trajectory observed rises. (b) The neural inference methods provide internal confidence levels along with the inference.**

*The training data needed in human-data-based inference.* Simulated data can be generated infinitely from a trained simulator while humans' variability is captured via adjustment of user-specific parameters. In contrast, gathering data from humans comes with a cost proportional to the quantity of data. This makes human-data-based inference difficult to scale up. Here, we investigated the effects of training a given inference model with various quantities of human user data (see Figure 9). This highlighted the negative consequences when the body of training data is not large or variety-rich enough. Human-data-based Neural Inference exhibited a significant decline in performance as the number of training users or trials per user decreased: when limited to seven users or when there were fewer than 40 trials per user, it was less accurate than Simulation-based Neural Inference. This result highlights the cost that each new target-selection task brings in human-data-based inference, thereby establishing clear limits on transferrability due to the costs of data collection. Furthermore, predicting the "sufficient" number of users for reliable inference is challenging due to inherent uncertainties. Our method mitigates these challenges by offering scalability through the use of simulation-generated data.

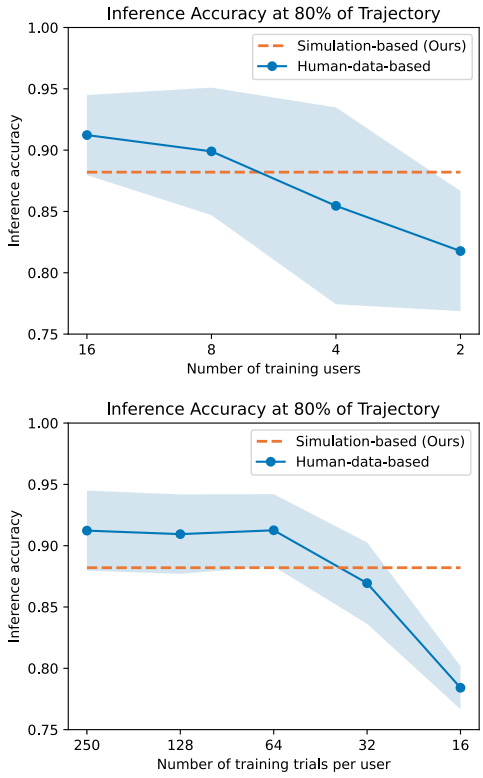


**Figure 9: Study 2: The performance of human-data-based neural inference with various numbers of training users (left) and training trials per user (right). The plots highlight the dependency of the method's performance on data availability, suggesting limited scalability. Shading denotes the standard deviation across five validation user sets.**

## 7 STUDY 3: REAL-TIME USER ASSISTANCE

Study 3 aims to demonstrate the efficiency and effectiveness of our inference method in assisting users with target selection tasks. This study implements a real-time interaction technique on a Meta Quest 2 device, comprising two key components: *inference* and *assistance*. The interaction operates by inferring the most likely target during a user's selection process at every timestep and assisting user to make more efficient selections by utilizing the inferred results. We built a visual-suggestion-based assistance wherein the user can visually check the inferred target and decide whether to select it.

Previous work has demonstrated that while predictive and heuristic techniques are highly accurate in less dense arrays, their effectiveness decreases in denser configurations [33, 55, 65]. To fully validate the performance of our simulation-based inference approach in a wide spectrum of tasks, this study compares our approach against various baseline techniques in two layouts: *Wide* and *Dense* (refer to Figure 5). We first assess how our approach enhances user assistance in the *Wide* layout, a setting representative of conventional scenarios where targets are adequately separated (Study 3A). Then, we shift our focus to the *Dense* layout, representing more challenging environments where traditional methods tend to struggle (Study 3B).

This study further investigates design options available for effectively incorporating the confidence levels into interaction techniques. To illustrate, in Section 7.2.3, we additionally present a new assistance technique, where an auto-click function is integrated into the visual suggestion. This feature allows the system to autonomously make decisions based on confidence levels of inference. We evaluated its impact on user performance improvements.

### 7.1 Study 3A: Wide Layout Targets

We first assessed our assistance performance in the *Wide* type of target layout. We implemented visual-suggestion-based assistance interaction wherein the inferred targets are shown to participants to support efficient selection. The system proactively provides visual suggestions with a sticky ray [81, 91], where the ray starts off in a straight line from the controller's current orientation and gradually curves at the end towards the inferred target. The inferred target is highlighted as if it were hovered over: in light blue if correct and light green otherwise. The selected target was the one inferred at the time when the participant triggered the selection. Unlike non-probabilistic inference methods (e.g., Nearest Neighbor), our probabilistic approach permits choosing to trigger visual suggestions only when the confidence value reaches a certain level, thereby avoiding distractions caused by unreliable suggestions.

*7.1.1 Participants.* We recruited 12 new participants (3 women and 9 men), ensuring none had previously participated in our Study 1. Their ages ranged from 19 to 29 (mean=24.75, SD=2.71). All participants had either normal or corrected-to-normal vision and were right-handed.

*7.1.2 Inference methods.* Since the *Quadratic Regression* method showed poorer overall inference performance than the other inference methods probed in Study 2, we excluded it from this study. Accordingly, our setup involved the other inference methods considered thus far: *Nearest Neighbor*, *Human-data-based Neural Inference*, and *Simulation-based Neural Inference* (i.e., our method). For a simple baseline, we added the basic target-selection scenario without inference, denoted as *None*. The system with Neural Inference gave the user visual suggestions only if the confidence values exceeded 50%. The study's non-probabilistic inference methods kept the suggestion active throughout the trials.

*7.1.3 Experiment design and procedure.* The study employed a within-subject design, featuring a $4 \times 2$ factorial structure: four *Inference Types* (None, Nearest Neighbor, Human-data-based Neural Inference, and Simulation-based Neural Inference) and two *Target Sizes* (Large and Small).

The task details were consistent with Study 1's, except for the addition of the assistance interaction. All participants signed consent forms. At the beginning of the experiment, participants were given a practice block reflecting each condition (Inference Type $\times$ Target Size). They were asked to perform trials as quickly and accurately as possible. They went through eight sessions, each with a distinct condition, in a counterbalanced order using a balanced Latin square [12]. Each session was arranged into three blocks. As in Study 1, participants calibrated their eye height, reported their fatigue levels, and were provided with breaks as desired after each block. Each participant completed 624 trials ($4 \times 2 \times 3 \times 26$), in

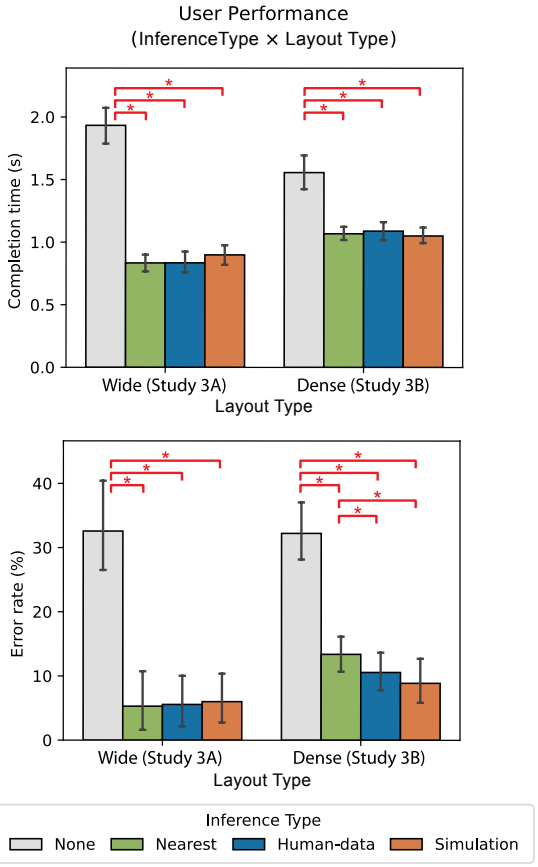


**Figure 10: Study 3: Our simulation-based inference approach significantly improved the speed and accuracy of users' target selection over the naive selection across two distinct target layout scenarios. No significant difference in performance was found between our assistance and human-data-based inference. An asterisk (*) indicates the statistically significant difference with $p < 0.05$ after adjustments using Bonferroni correction. Error bars denote 95% confidence intervals.**

approximately 30 minutes. The study adhered to the local ethical protocols for approval.

*7.1.4 Apparatus and implementation details.* The interaction was performed on a Quest 2 device. For the neural-inference methods, we converted the pre-trained network models, initially implemented in PyTorch, to Open Neural Network Exchange, or ONNX, format. This allowed us to run them on Unity's Barracuda engine. The experiment program was executed on a desktop PC equipped with an NVIDIA RTX 3080 GPU, wired to the VR device. This setup enabled the neural-inference network to operate in real time, with a latency of 5–10 ms per inference. User trajectory data were collected and used for inference at 50-ms intervals.

*7.1.5 Results.* We analyzed participants' task performance using a two-way (Inference Type × Target Size) repeated-measures ANOVA with Greenhouse—Geisser correction. The absence of significant effects of the block on both performance metrics ($p$>0.05) suggests that the learning effect was effectively minimized by the practice session, allowing us to use data from all blocks in the analysis. The ANOVA results showed a statistically significant effect of Inference Type: $F_{3,33}$=206.64, $p$<0.001 for completion time and $F_{3,33}$=132.94, $p$<0.001 for error rate. Post-hoc tests with Bonferroni correction showed significant differences between Inference Types (see Figure 10). All three inference methods led to significantly better task performance than the None condition in terms of both completion time and error rate (all $p$<0.001). There were no other significant differences between Inference Types. We report the details of further analysis with Target Size in Supplement D.1.

Overall, with large effective target sizes — in wide layouts where targets are sufficiently separated — all inference methods significantly enhanced user performance compared to naive selection. The error rate of the assisted target selections was less than 6% on average. Considering that participants were instructed to prioritize both speed and accuracy, it is plausible to expect even higher accuracy in scenarios where speed is less prioritized, as indicated in previous studies [99, 101]. Our method, based on simulated data, demonstrated performance comparable to inference methods trained on actual human data. As shown in previous work [55], the nearest neighbor approach exhibited a high level of assistance performance for targets with high effective size.

## 7.2 Study 3B: Dense Layout Targets

The visual-suggestion assistance with our inference method was evaluated with the *Dense* layout of targets, more challenging for target inference. We maintained consistency across task implementation, experiment design, and procedure, aligning them with Study 3A, except for the change of the target layout. Following the evaluation of visual-suggestion assistance, this study explores an alternative selection technique based on our inference outputs. We specifically examined auto-click features, offering a more active system engagement compared to the passive nature of visual-suggestion assistance.

*7.2.1 Participants.* Twenty new participants (13 women and 7 men; ages ranged from 18 to 37; mean=25.6, SD=4.1) were recruited. All had normal or corrected-to-normal vision, were right-handed, and had not participated in our previous studies.

*7.2.2 Results.* A two-way (Inference Type × Target Size) repeated-measures ANOVA with Greenhouse–Geisser correction revealed a significant effect of Inference Type on both completion time ($F_{3,57}$=82.31, $p$<0.001) and error rate ($F_{3,57}$=154.20, $p$<0.001). Post-hoc tests with Bonferroni correction identified significant differences among the inference methods (see Figure 10). Consistent with Study 3A, all inference methods significantly improved task performance compared to the None condition, in terms of both completion time and error rate (all $p$<0.001).

The key distinction from Study 3A was that both Human-data-based and Simulation-based Neural Inference methods exhibited lower error rates than Nearest Neighbor ($p$=0.012 when compared to Human-data-based; $p$<0.001 for Simulation-based Inference). No other significant differences were found between the inference types. These results indicate that marginal differences in inference accuracy didn't significantly impact assistance performance.

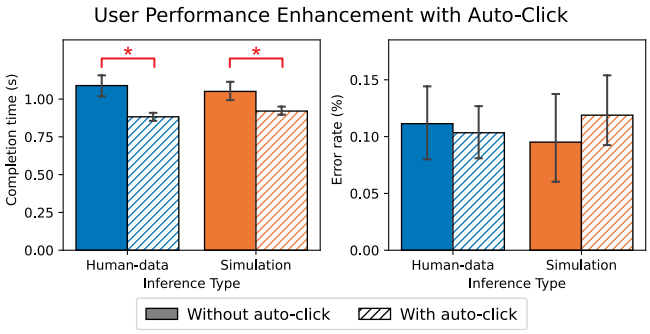


**Figure 11: Study 3: Both neural inference methods (human-data-based and simulation-based) improved completion times with confidence-based automated-click assistance, maintaining similar levels of accuracy. An asterisk (*) denotes $p < 0.05$. Error bars denote 95% confidence intervals.**

Our simulation-based inference outperformed the nearest-neighbor method in error rates, despite comparable levels of inference accuracy. Additionally, it matched the performance of human-data-based inference, despite slightly lower inference accuracy.

*7.2.3 Exploring the utility of confidence levels with auto-click.* Inference confidence levels offer various options for designing assistance interactions, ranging from passive to active system involvement. The visual suggestion represents passive usage, where the system proposes actions but the user retains decision-making control. In contrast, for clear user intents like text entry on a keyboard UI, the system can autonomously process inputs to enhance efficiency. Dwell-click [35, 56, 80] is a common example where the system identifies user intention and clicks based on the user's pointing duration. While dwell-click is prone to unintended activations [41], inference confidence can offer a more reliability for activation. A balance between passive and active engagement is also possible, for instance, by dynamically adjusting the dwell-click threshold using inference confidence [68].

As a demonstrative example, we tested an active assistance interaction: *auto-click*. Here, participants controlled a ray upon the same visual suggestions, but the system directly selected the inferred target when certain criteria were met. This auto-click feature was applied to all three inference methods. For neural-inference methods, the inferred target was auto-selected if confidence exceeded 90%. For Nearest Neighbor, we used a time-based criterion, auto-selecting the target if the inferred target remained unchanged for over 300 ms.[3]

Auto-click's performance was evaluated with the same twenty participants, following the same procedure. Significant effects were observed in task completion time ($F_{1,19}$=24.28, $p$<0.001) and error rate ($F_{1,19}$=41.69, $p$<0.001) with the auto-click feature.[4] The confidence-based auto-click significantly enhanced completion time for neural-inference methods (all $p$<0.001) without significant error rate differences (see Figure 11).[5] Nearest Neighbor's time-based auto-click led to a significantly better error rate ($p$<0.001) without affecting completion time (see Supplement D.2). The results support using confidence levels as criteria for auto-click. A key benefit of using confidence levels is their consistency. Unlike dwell-click thresholds that vary widely from 300 ms to 2 s depending on the input method [67], inference confidence offers a more stable threshold directly linked to the inference quality.

[3]The value was as in prior work of dwell-click with hand-held pointing devices [10, 72].
[4]A two-way (*With-or-without Auto-Click* × *Inference Type*) repeated-measures ANOVA with Greenhouse–Geisser correction was conducted.
[5]Post-hoc pairwise tests with Bonferroni correction was conducted.

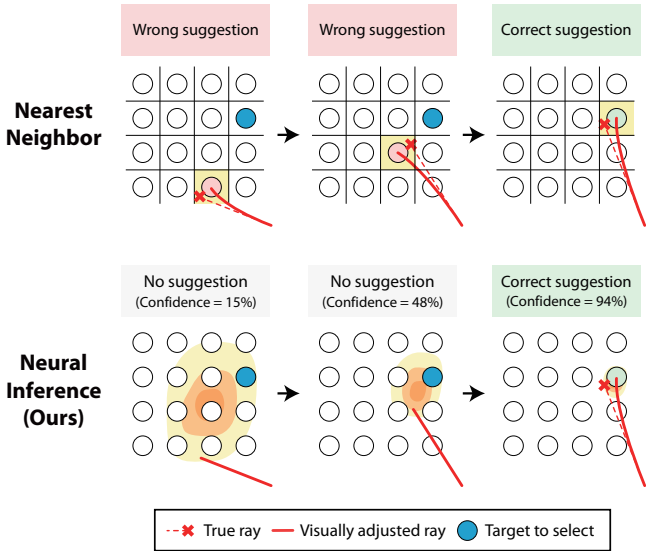


**Figure 12: Our neural inference approach enables the system to selectively offer visual suggestions to the user based on internally measured inference confidence. In contrast, existing heuristic assistance methods like nearest neighbor continuously offer visual suggestions, often leading to suggestions towards incorrect targets, thus hindering user performance. The orange contour, overlaying the target array, represents the system's internally measured inferred posterior, which is not visible to the actual participants.**

## 7.3 Discussion

*Contributing factors to superior assistance performance.* The key advantage of the neural inference methods over the nearest neighbor approach was *selective activation of visual suggestions* based on inference confidence (see Figure 12). The neural inference methods activated visual suggestion for only 43% of the duration, with an 81% accuracy in targeting the user's intended target. In contrast, the nearest neighbor method was accurate only 35% of the time. This selective feature was effective in reducing visual clutter and reducing users' clicks on less certain locations, especially in denser target configurations.

We analyzed the impact of each inference method on participants' cursor movements by measuring the number of submovements[6] and total travel distance. With assistance from the three methods, participants showed fewer submovements (2.93 ± 0.36)

[6]Following a previous practice [50], we identified each submovement's onset by locating the local minima in cursor speed, following smoothing with a Gaussian filter ($\sigma$ = 3).

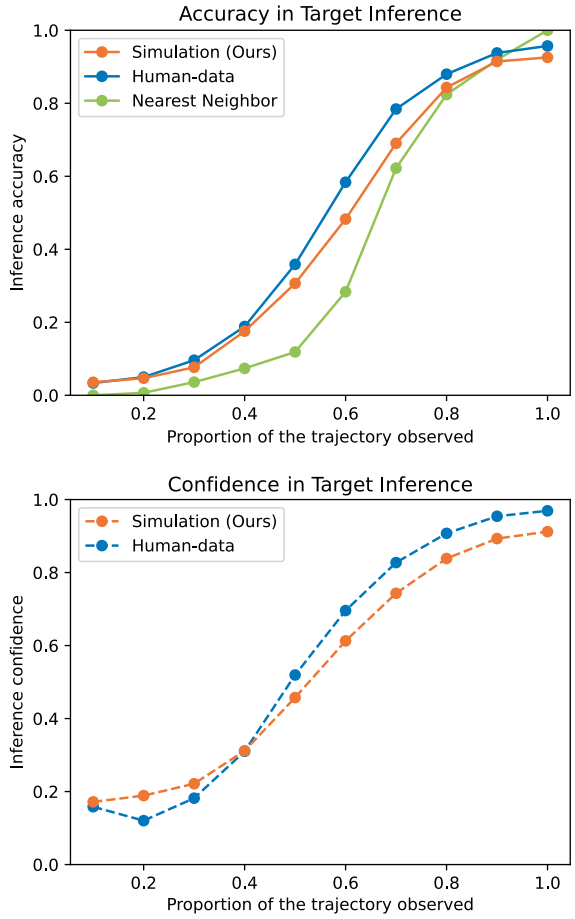


**Figure 13: Study 3: Mean accuracy and confidence in target inference during user-assisted target selection with each inference method.**

and shorter travel distances ($3.03 \pm 0.19$ m) compared to naive selection ($4.03 \pm 0.74$ submovements, $3.19 \pm 0.23$ m). However, there were no significant differences between the three methods, suggesting that the neural inference's selective visual suggestions mainly affected decision-making regarding click timing, rather than affecting cursor movement patterns.

*Inference accuracy with users assisted.* Having noted that the assistance influenced cursor movements, we examined its effect on the inference accuracy of each method. We evaluated each method's inference accuracy using trajectory data from trials with participants assisted by corresponding inference (Figure 13). Comparing with Study 2's results (Figure 8), which used naive selection trajectory data, we noted a consistent trend: Our method lagged behind human-data-based inference around the midpoint of the trajectory but narrowed the gap towards the trajectory's end, and ultimately showed comparable accuracy to the nearest-neighbor method. There was a general decline in inference accuracy compared to Study 2, because assistance reduced the time the cursor spent near the target, where inference accuracy is typically higher.

## 8 DISCUSSION AND CONCLUSION

This work introduces a novel simulation-based target inference method, leveraging biomechanical simulation. The three studies we conducted shed light on ways of applying this idea in HCI. We can sum up their findings as follows:

- In Study 1, our simulator replicated human performance measurements with high fidelity, falling within a one-standard-deviation range across various levels of task difficulty while also capturing motion variability.
- In Study 2, our inference model, trained exclusively on simulated data, achieved accuracy similar to the human-data-based approach's, with a short inference time: ~5 ms. The model usefully supplies a confidence level for its predictions.
- Study 2 showed also that data from at least seven participants were needed for exceeding the accuracy of our simulation-based inference model in our evaluation setting.
- In Study 3, a selection technique implemented using our inference method significantly improved speed and accuracy of users' target selection over naive selection, leading to fewer cursor submovements. Furthermore, the selective assistance using measured inference confidence led to higher accuracy in densely arranged target selection scenarios compared to pre-existing heuristics-based assistance.

Below, we discuss the implications of our findings and explore opportunities for further extensions.

*Biomechanics as a human-motion prior.* Our results illuminate the significant utility of human biomechanics as an essential prior in the study of humans' interactive motion. Traditionally, understanding such movements required resource-intensive data collection or heuristic programming, which may lack realism. We utilized our prior knowledge of biomechanical movement (limb kinematics, motor noise, and natural posture deviation) to generate realistic motion with variability. Study 1 showed that this approach faithfully captures motor variability, and Studies 2 and 3 provided evidence of the performance of the inference model trained with such simulation. This work showcases the potential of the biomechanics model as a powerful tool for replicating, analyzing, and understanding human motion.

*Utility of inference confidence.* In Study 3, we demonstrated that confidence levels from probabilistic inference function well as operational indicators within the system to prompt assistance. One factor contributing to our approach's enhanced end-user performance compared to the nearest-neighbor condition could be our selective suggestion of inferred targets, enabled by confidence levels. Also, confidence-driven auto-clicking further improved users' target-selection performance. These results suggest our probabilistic method effectively filters out unreliable inferences, a feat impossible with non-probabilistic methods. While we used a simplistic fixed threshold for confidence, future work should explore optimization techniques [54] or RL [97] for intelligently identifying optimal confidence thresholds or for adjusting to the desired balance between speed and accuracy.

*Intra- and inter-user variability.* Our simulation faithfully reproduces the intra-user variability. However, we observed that human

participants exhibited greater inter-user variability than the simulator, which may be attributed to factors not captured by our current user parameters. For instance, humans' internal reward functions can vary significantly. Also, users also complete selection with varying levels of attention, experience various levels of fatigue, and undergo unique learning processes, all contributing to inter-user variability. Further research is needed to capture these inter-user-level differences in motion generation. Although the greater inter-user variability in human data leads to slightly better inference accuracy, this difference does not necessarily translate to more effective assistance for target selection, as Study 3 attests, highlighting the efficacy of the simulation-based approach.

*Personalized simulation and inference.* The inclusion of user-specific parameters to account for motions' variability (Table 1) has demonstrated effectiveness in our simulation setting. Currently, we uniformly sample user parameters from a set range to reflect population-level variability. However, in scenarios requiring inference for a specific user or context, adjusting the user parameters' sampling distribution is a viable option for better representing the purpose at hand. Previous work has demonstrated the feasibility of inversely inferring user-specific parameters through neural density estimation techniques [63]. This opens opportunities for *personalized target inference*: A system can observe multiple target-selection trials from a user to infer that user's unique user parameters. The inferred parameters can then serve as the new prior for subsequent trials; thereby, the system can customize and enhance the system's target inference for this individual.

*Deployment in real-world application.* Our method can be applied outside of research settings with minimal alterations. The first challenge involves identifying the start of a user's aimed movement, which is non-trivial in real-world sequences. Techniques similar to those proposed by Chapuis et al. [14], which detect the start of movement based on the cursor's pause time and subsequent movement distance, provide a viable solution. The second challenge is the assumption that the target array is known in advance, crucial for generating appropriate simulated data and training the inference model. To adapt to real-world scenarios, the inference model requires pre-training on simulations that include diverse target configurations. This intensive pre-training enables the model to contextually infer target locations by processing the trajectory in conjunction with the specific target array presented, adapting its inference to the given situational context.

*Generalizability.* The proposed simulation-based target-inference approach has potential for application in target-selection techniques beyond raycasting, since none of the steps in our method are limited to certain interactions. Recent studies [13, 40] have expanded the repertoire of biomechanical models available, enhancing the versatility of our approach for replicating human motion in interactive tasks. Meanwhile, on the inference side, the flexibility inherent in the neural networks makes it suitable for handling a broader range of data channels or even longer trajectories [63]. Importantly, the fast inference (~5 ms) makes our approach compatible with systems for real-time selection assistance across various interfaces. Another advantage of our method is that training data can be generated through different means, provided that synthetic motion dynamics are available. This makes it possible to use optimal-control-based methods such as LQG [25] and MPC [46]. However, it is critical to remember that inference accuracy is contingent on the validity of the synthetic data.

*Limitations and future work.* Our research simultaneously has identified several challenges for further investigation, to broaden the area of application. Firstly, future research could focus on more realistic target-selection tasks; our validation was limited to simplified scenarios (fixed starting points, grid-based arrangements, uniform visual shapes, etc.). Secondly, simulations of human motion can be further enhanced via more realism by incorporating factors such as human-like perceptual processes (visual search), intermittency of motor control, and muscle actuation. Thirdly, more user data channels beyond just end-effector trajectory could be included; additional sensor data (hand position, eye-gaze, etc.) could enrich models and improve accuracy. Lastly, the field lacks a formal process to translate human movements into computational models; current methods need tuning of simulation parameters (user-specific variables, reward formulations, etc.) across applications, limiting efficient generalization. We hope our research serves as a pioneering example, inspiring future work in RL-driven biomechanical simulations and enabling cost-effective design evaluation, hypothesis testing, and study of complex interactions in HCI.

## ACKNOWLEDGMENTS

This work was supported by the Research Council of Finland (328400, 345604, 341763, 328813, 357578), the National Research Foundation of Korea (RS-2023-00237631, RS-2023-00223062), and the Institute of Information and Communications Technology Planning and Evaluation (2020-0-01361). We thank Helena Kirsta for the assistance in building the initial prototype of our target selection system.

## REFERENCES

[1] Bashar I Ahmad, Patrick M Langdon, Simon J Godsill, Richard Donkor, Rebecca Wilde, and Lee Skrypchuk. 2016. You do not have to touch to select: A study on predictive in-car touchscreen with mid-air selection. In *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. 113–120.

[2] Ferran Argelaguet and Carlos Andujar. 2013. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics* 37, 3 (2013), 121–136.

[3] Takeshi Asano, Ehud Sharlin, Yoshifumi Kitamura, Kazuki Takashima, and Fumio Kishino. 2005. Predictive interaction using the delphian desktop. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*. 133–141.

[4] Shiri Azenkot and Shumin Zhai. 2012. Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 251–260.

[5] Marc Baloup, Thomas Pietrzak, and Géry Casiez. 2019. Raycursor: A 3d pointing facilitation technique based on raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[6] Mark A Beaumont, Wenyang Zhang, and David J Balding. 2002. Approximate Bayesian computation in population genetics. *Genetics* 162, 4 (2002), 2025–2035.

[7] Xiaojun Bi, Yang Li, and Shumin Zhai. 2013. FFitts law: Modeling finger touch with fitts' law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1363–1372.

[8] Pradipta Biswas, Gokcen Aslan Aydemir, Pat Langdon, and Simon Godsill. 2013. Intent recognition using neural networks and Kalman filters. In *International Workshop on Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*. 112–123.

[9] Renaud Blanch, Yves Guiard, and Michel Beaudouin-Lafon. 2004. Semantic pointing: Improving target acquisition with control-display ratio adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 519–526.

[10] Michael Bohan, Alex Chaparro, and Deborah Scarlett. 1998. The effects of selection technique on target acquisition movements made with a mouse. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 42. 473–475.

[11] Gunnar A Borg. 1982. Psychophysical bases of perceived exertion. *Medicine and Science in Sports and Exercise* 14, 5 (1982), 377–381.

[12] James V Bradley. 1958. Complete counterbalancing of immediate sequential effects in a Latin square design. *Journal of the American Statistical Association* 53, 282 (1958), 525–528.

[13] Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. 2022. MyoSuite: A contact-rich simulation suite for musculoskeletal motor control. In *Learning for Dynamics and Control Conference*. 492–507.

[14] Olivier Chapuis, Renaud Blanch, and Michel Beaudouin-Lafon. 2007. Fitts' law in the wild: A field study of aimed movements. *LRI Technical Report* 1480 (2007).

[15] Olivier Chapuis, Jean-Baptiste Labrune, and Emmanuel Pietriga. 2009. DynaSpot: Speed-dependent area cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1391–1400.

[16] Noshaba Cheema, Laura A Frey-Law, Kourosh Naderi, Jaakko Lehtinen, Philipp Slusallek, and Perttu Hämäläinen. 2020. Predicting mid-air interaction movements and fatigue using deep reinforcement learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[17] Xiuli Chen, Gilles Bailly, Duncan P Brumby, Antti Oulasvirta, and Andrew Howes. 2015. The emergence of interactive behavior: A model of rational menu search. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 4217–4226.

[18] Aldrich Clarence, Jarrod Knibbe, Maxime Cordeil, and Michael Wybrow. 2021. Unscripted retargeting: Reach prediction for haptic retargeting in virtual reality. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. 150–159.

[19] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. 2020. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences* 117, 48 (2020), 30055–30062.

[20] Tor-Salve Dalsgaard, Jarrod Knibbe, and Joanna Bergström. 2021. Modeling pointing for 3D target selection in VR. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*. 1–10.

[21] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2017. Density estimation using Real NVP. In *International Conference on Learning Representations*.

[22] Seungwon Do, Minsuk Chang, and Byungjoo Lee. 2021. A simulation model of intermittently controlled point-and-click behaviour. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–17.

[23] Stephen R Ellis and Robert J Hitchcock. 1986. The emergence of Zipf's law: Spontaneous encoding optimization by users of a command language. *IEEE Transactions on Systems, Man, and Cybernetics* 16, 3 (1986), 423–427.

[24] 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), 1–15.

[25] 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 (2022), 1–70.

[26] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47, 6 (1954), 381.

[27] Tamar Flash and Neville Hogan. 1985. The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience* 5, 7 (1985), 1688–1703.

[28] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of language modeling and its personalization on touchscreen typing performance. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 649–658.

[29] Manuel Glöckler, Michael Deistler, and Jakob H Macke. 2022. Variational methods for simulation-based inference. In *International Conference on Learning Representations*.

[30] Joshua Goodman, Gina Venolia, Keith Steury, and Chauncey Parker. 2002. Language modeling for soft keyboards. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*. 194–195.

[31] Tovi Grossman and Ravin Balakrishnan. 2005. The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 281–290.

[32] Tovi Grossman and Ravin Balakrishnan. 2006. The design and evaluation of selection techniques for 3D volumetric displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*. 3–12.

[33] Maxime Guillon, François Leitner, and Laurence Nigay. 2015. Investigating visual feedforward for target expansion techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2777–2786.

[34] Michael U Gutmann and Jukka Corander. 2016. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research* 17, 125 (2016), 1–47.

[35] John Paulin Hansen, Vijay Rajanna, I Scott MacKenzie, and Per Bækgaard. 2018. A Fitts' law study of click and dwell interaction by gaze, head and mouse with a head-mounted display. In *Proceedings of the Workshop on Communication by Gaze Interaction*. 1–5.

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

[37] 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.

[38] Xueshi Hou, Jianzhong Zhang, Madhukar Budagavi, and Sujit Dey. 2019. Head and body motion prediction to enable mobile VR experiences with low latency. In *2019 IEEE Global Communications Conference (GLOBECOM)*. 1–7.

[39] Chien-Ming Huang, Sean Andrist, Allison Sauppé, and Bilge Mutlu. 2015. Using gaze patterns to predict task intent in collaboration. *Frontiers in Psychology* 6 (2015), 1049.

[40] 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*. 1–14.

[41] Robert JK Jacob. 1990. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 11–18.

[42] Yifeng Jiang, Tom Van Wouwe, Friedl De Groote, and C Karen Liu. 2019. Synthesis of biologically realistic human motion using joint torque actuation. *ACM Transactions On Graphics* 38, 4 (2019), 1–12.

[43] Jussi Jokinen, Aditya Acharya, Mohammad Uzair, Xinhui Jiang, and Antti Oulasvirta. 2021. Touchscreen typing as optimal supervisory control. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[44] Jussi PP Jokinen, Zhenxin Wang, Sayan Sarcar, Antti Oulasvirta, and Xiangshi Ren. 2020. Adaptive feature guidance: Modelling visual search with graphical layouts. *International Journal of Human–Computer Studies* 136 (2020), 102376.

[45] Antti Kangasrääsiö, Kumaripaba Athukorala, Andrew Howes, Jukka Corander, Samuel Kaski, and Antti Oulasvirta. 2017. Inferring cognitive models from data using approximate Bayesian computation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 1295–1306.

[46] 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 (2023), 1–50.

[47] Minhae Kwon, Saurabh Daptardar, Paul R Schrater, and Xaq Pitkow. 2020. Inverse rational control with partially observable continuous nonlinear dynamics. *Advances in Neural Information Processing Systems* 33 (2020), 7898–7909.

[48] Edward Lank, Yi-Chun Nikko Cheng, and Jaime Ruiz. 2007. Endpoint prediction using motion kinematics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 637–646.

[49] Huy Viet Le, Valentin Schwind, Philipp Göttlich, and Niels Henze. 2017. PredicTouch: A system to reduce touchscreen latency using neural networks and inertial measurement units. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. 230–239.

[50] Byungjoo Lee, Mathieu Nancel, Sunjun Kim, and Antti Oulasvirta. 2020. AutoGain: Gain function adaptation with submovement efficiency optimization. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[51] Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. 2019. Scalable muscle-actuated human simulation and control. *ACM Transactions On Graphics* 38, 4 (2019), 1–13.

[52] William H Levison, Sheldon Baron, and David L Kleinman. 1969. A model for human controller remnant. *IEEE Transactions on Man-Machine Systems* 10, 4 (1969), 101–108.

[53] Zhi Li, Maozheng Zhao, Dibyendu Das, Hang Zhao, Yan Ma, Wanyu Liu, Michel Beaudouin-Lafon, Fusheng Wang, Iv Ramakrishnan, and Xiaojun Bi. 2022. Select or suggest? Reinforcement learning-based method for high-accuracy target selection on touchscreens. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–15.

[54] Yi-Chi Liao, John J Dudley, George B Mo, Chun-Lien Cheng, Liwei Chan, Antti Oulasvirta, and Per Ola Kristensson. 2023. Interaction design with multi-objective Bayesian optimization. *IEEE Pervasive Computing* 22, 1 (2023), 29–38.

[55] Yiqin Lu, Chun Yu, and Yuanchun Shi. 2020. Investigating bubble mechanism for ray-casting to improve 3d target acquisition in virtual reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 35–43.

[56] Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast gaze typing with an adjustable dwell time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 357–360.

[57] Jennifer Mankoff, Scott E Hudson, and Gregory D Abowd. 2006. Interaction techniques for ambiguity resolution in recognition-based interfaces. In *ACM SIGGRAPH 2006 Courses*. 1–10.

[58] J Alberto Álvarez Martín, Henrik Gollee, Jörg Müller, and Roderick Murray-Smith. 2021. Intermittent control as a model of mouse movements. *ACM Transactions on Computer-Human Interaction* 28, 5 (2021), 1–46.

[59] Michael McGuffin and Ravin Balakrishnan. 2002. Acquisition of expanding targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 57–64.

[60] Michael J McGuffin and Ravin Balakrishnan. 2005. Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *ACM Transactions on Computer-Human Interaction* 12, 4 (2005), 388–422.

[61] Mark R Mine. 1995. Virtual environment interaction techniques. *UNC Chapel Hill Computer Science Technical Report TR95-018* (1995).

[62] Hee-Seung Moon, Seungwon Do, Wonjae Kim, Jiwon Seo, Minsuk Chang, and Byungjoo Lee. 2022. Speeding up inference with user simulators through policy modulation. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–21.

[63] 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*. 1–20.

[64] Hee-Seung Moon and Jiwon Seo. 2021. Optimal action-based or user prediction-based haptic guidance: Can you do even better?. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[65] Martez E Mott and Jacob O Wobbrock. 2014. Beating the bubble: Using kinematic triggering in the bubble lens for acquiring small, dense targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 733–742.

[66] Jörg Müller, Antti Oulasvirta, and Roderick Murray-Smith. 2017. Control theoretic models of pointing. *ACM Transactions on Computer-Human Interaction (TOCHI)* 24, 4 (2017), 1–36.

[67] Christian Müller-Tomfelde. 2007. Dwell-based pointing in applications of human computer interaction. In *Proceedings of the 11th IFIP TC 13 International Conference on Human-Computer Interaction*. 560–573.

[68] Aanand Nayyar, Utkarsh Dwivedi, Karan Ahuja, Nitendra Rajput, Seema Nagar, and Kuntal Dey. 2017. OptiDwell: Intelligent adjustment of dwell click time. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. 193–204.

[69] Antti Oulasvirta, Jussi PP Jokinen, and Andrew Howes. 2022. Computational rationality as a theory of interaction. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–14.

[70] George Papamakarios, Eric T Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. 2021. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research* 22, 57 (2021), 1–64.

[71] Eunji Park and Byungjoo Lee. 2020. An intermittent click planning model. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[72] J Karen Parker, Regan L Mandryk, and Kori M Inkpen. 2005. TractorBeam: Seamless integration of local and remote pointing for tabletop displays. In *Proceedings of Graphics Interface 2005*. 33–40.

[73] Andriy Pavlovych and Wolfgang Stuerzlinger. 2009. The tradeoff between spatial jitter and latency in pointing tasks. In *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. 187–196.

[74] Philip Quinn and Shumin Zhai. 2018. Modeling gesture-typing movements. *Human–Computer Interaction* 33, 3 (2018), 234–280.

[75] Stefan T Radev, Ulf K Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. 2020. BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[76] Danilo Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. In *International Conference on Machine Learning*. 1530–1538.

[77] 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.

[78] Richard A Schmidt, Howard Zelaznik, Brian Hawkins, James S Frank, and John T Quinn Jr. 1979. Motor-output variability: A theory for the accuracy of rapid motor acts. *Psychological Review* 86, 5 (1979), 415.

[79] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[80] Linda E Sibert and Robert JK Jacob. 2000. Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 281–288.

[81] Frank Steinicke, Timo Ropinski, and Klaus Hinrichs. 2006. Object selection in virtual environments using an improved virtual pointer metaphor. In *International Conference on Computer Vision and Graphics*. 320–326.

[82] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[83] Robert J Teather and Wolfgang Stuerzlinger. 2015. Factors affecting mouse-based 3d selection in desktop vr systems. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*. 10–19.

[84] 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*. 5026–5033.

[85] Emanuel Todorov and Michael I Jordan. 1998. Smoothness maximization along a predefined path accurately predicts the speed profiles of complex arm movements. *Journal of Neurophysiology* 80, 2 (1998), 696–714.

[86] Robert J Van Beers, Patrick Haggard, and Daniel M Wolpert. 2004. The role of execution noise in movement variability. *Journal of Neurophysiology* 91, 2 (2004), 1050–1063.

[87] Herman Van Der Kooij and Robert J Peterka. 2011. Non-linear stimulus-response behavior of the human stance control system is predicted by optimization of a system with sensory and motor noise. *Journal of Computational Neuroscience* 30 (2011), 759–778.

[88] Lode Vanacken, Chris Raymaekers, and Karin Coninx. 2006. Evaluating the influence of multimodal feedback on egocentric selection metaphors in virtual environments. In *Haptic and Audio Interaction Design: First International Workshop*. 12–23.

[89] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 659–668.

[90] Colin Ware and Ravin Balakrishnan. 1994. Reaching for objects in VR displays: Lag and frame rate. *ACM Transactions on Computer-Human Interaction (TOCHI)* 1, 4 (1994), 331–356.

[91] Chadwick A Wingrave, Doug A Bowman, and Naren Ramakrishnan. 2002. Towards preferences in virtual environment interfaces. In *Eurographics Workshop on Virtual Environments*. 63–72.

[92] Chadwick A Wingrave, Ryan Tintner, Bruce N Walker, Doug A Bowman, and Larry F Hodges. 2005. Exploring individual differences in raybased selection: Strategies and traits. In *Proceedings of the 2005 IEEE Conference on Virtual Reality*. 163–170.

[93] Aileen Worden, Nef Walker, Krishna Bharat, and Scott Hudson. 1997. Making computers easier for older adults to use: Area cursors and sticky icons. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. 266–271.

[94] Haijun Xia, Ricardo Jota, Benjamin McCanny, Zhe Yu, Clifton Forlines, Karan Singh, and Daniel Wigdor. 2014. Zero-latency tapping: Using hover information to predict touch locations and eliminate touchdown latency. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. 205–214.

[95] Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, and Efstathios Velenis. 2020. An ensemble deep learning approach for driver lane change intention inference. *Transportation Research Part C: Emerging Technologies* 115 (2020), 102615.

[96] Xin Yi, Chen Wang, Xiaojun Bi, and Yuanchun Shi. 2020. Palmboard: Leveraging implicit touch pressure in statistical decoding for indirect text entry. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[97] Difeng Yu, Ruta Desai, Ting Zhang, Hrvoje Benko, Tanya R Jonker, and Aakar Gupta. 2022. Optimizing the timing of intelligent suggestion in virtual reality. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–20.

[98] Difeng Yu, Qiushi Zhou, Joshua Newn, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2020. Fully-occluded target selection in virtual reality. *IEEE Transactions on Visualization and Computer Graphics* 26, 12 (2020), 3402–3413.

[99] Shumin Zhai, Jing Kong, and Xiangshi Ren. 2004. Speed–accuracy tradeoff in Fitts' law tasks—on the equivalency of actual and nominal pointing precision. *International Journal of Human-Computer Studies* 61, 6 (2004), 823–856.

[100] Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 246–253.

[101] Xiaolei Zhou and Xiangshi Ren. 2010. An investigation of subjective operational biases in steering tasks evaluation. *Behaviour & Information Technology* 29, 2 (2010), 125–135.

[102] Suwen Zhu, Yoonsang Kim, Jingjie Zheng, Jennifer Yi Luo, Ryan Qin, Liuping Wang, Xiangmin Fan, Feng Tian, and Xiaojun Bi. 2020. Using Bayes' theorem for command input: Principle, models, and applications. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–15.

[103] Brian Ziebart, Anind Dey, and J Andrew Bagnell. 2012. Probabilistic pointing target prediction via inverse optimal control. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*. 1–10.