Predicting Ray Pointer Landing Poses in VR Using Multimodal LSTM-Based Neural Networks

Wenxuan Xu
Guarini School of Graduate and Advanced Studies
Dartmouth College

Yushi Wei
Computational Media and Arts Thrust
The Hong Kong University of Science and Technology (Guangzhou)

Xuning Hu
School of Advanced Technology
Xi'an Jiaotong-Liverpool University

Wolfgang Stuerzlinger
School of Interactive Arts + Technology
Simon Fraser University

Yuntao Wang*
Department of Computer Science and Technology
Tsinghua University

Hai-Ning Liang†
Computational Media and Arts Thrust
The Hong Kong University of Science and Technology (Guangzhou)

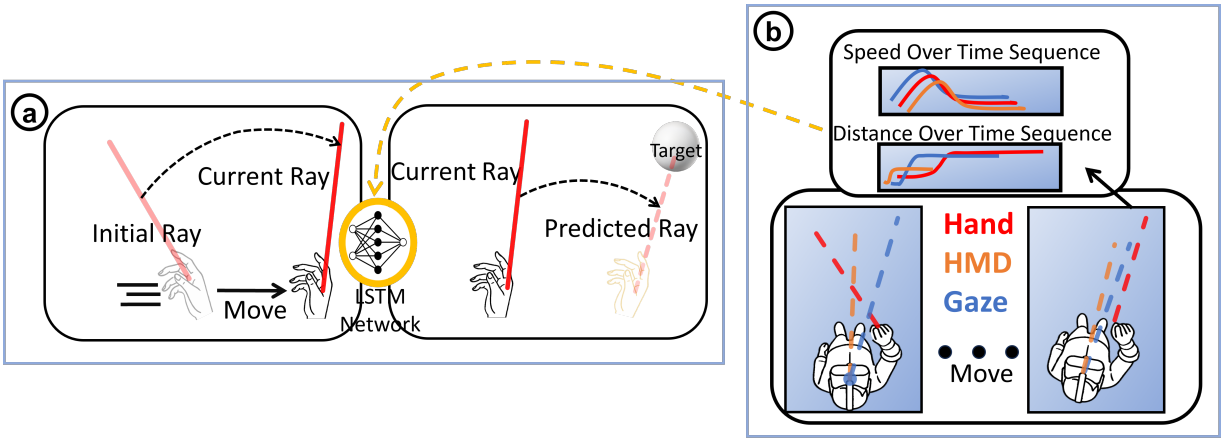

Figure 1: We present a novel LSTM-based approach for ray pointer landing Pose prediction. (a) A user's target selection is assisted by a novel inference network that proactively infers the user's future ray endpoint Pose from their prior and current movements. (b) Our network is trained using speed-and-distance over time features from three modalities: Hand, HMD, and Gaze.

Abstract

Target selection is one of the most fundamental tasks in VR interaction systems. Prediction heuristics can provide users with a smoother interaction experience in this process. Our work aims to predict the ray landing pose for hand-based raycasting selection in Virtual Reality (VR) using a Long Short-Term Memory (LSTM)-based neural network with time-series data input of speed and distance over time from three different pose channels: hand, Head-Mounted Display (HMD), and eye. We first conducted a study to collect motion data from these three input channels and analyzed these movement behaviors. Additionally, we evaluated which combination of input modalities yields the optimal result. A second study validates raycasting across a continuous range of distances, angles, and target sizes. On average, our technique's predictions were within 4.6° of the true landing Pose when 50% of the way through the movement. We compared our LSTM neural network model to a kinematic information model and further validated its generalizability in two ways: by training the model on one user's data and testing on other users (cross-user) and by training on a group of users and testing on entirely new users (unseen users). Compared to the baseline and a previous kinematic method, our model increased prediction accuracy by a factor of 3.5 and 1.9, respectively, when 40% of the way through the movement.

Index Terms: Virtual Reality, Pointing Selection, Modeling, Neural Networks

1 Introduction

Target selection is one of the most fundamental and common tasks in Virtual Environment (VE) interaction. Yet using controllers or bare hands for raycasting to interact with virtual objects remains a prevalent interaction method on modern VR headset[2], especially for objects beyond arm's length, including UI elements. Although raycasting is widely used, its effectiveness can suffer when selecting objects that appear small, are occluded, or within dense environments. Users might then need to exert more effort, making optimizing interaction methods for VEs a worthwhile research objective.

Various methods have been proposed to optimize raycasting, such as dual-stage mechanisms [26, 3] and heuristic approaches to aid selection [24]. Several studies have used prediction to enhance the user experience, such as forecasting short-term hand movement trajectories for pre-rendering or correcting hardware tracking errors, and modeling endpoint distributions combined with probabilistic inference for more accurate selection [25, 40, 61, 57]. Some of these techniques predict user intent to interact with objects during the user's selection process [12, 44, 8], but these approaches require prior knowledge of objects in the scene, essentially making it a classification problem limited to specific objects. In contrast, we consider that predicting the final landing position of the user's cursor during the selection process in *unconstrained* scenes, i.e.,

*e-mail: yuntaowang@tsinghua.edu.cn. Corresponding Author
†e-mail: hainingliang@hkust-gz.edu.cn. Corresponding Author

without knowledge of objects, has significant potential to assist selection and save user effort. There has been much work predicting the final click position of a mouse in 2D [56], and some have extended the Kinematic Template Matching (KTM) method to 3D, e.g., [31]. However, these approaches are currently limited by the need to collect personalized templates for each user and, compared to without predictions, achieve only low accuracy in the later stages of selection. Additionally, no other methods exist for predicting ray landing poses in VR.

As current VR devices easily acquire head pose data and increasingly incorporate eye-tracking, many researchers have started using multimodal data to optimize interaction experiences. For instance, Lystbaek et al. [42] combined eye raycasting with hand gestures for faster selection. Given the rich information in multimodal data, some studies have begun using machine learning models to identify user intent, e.g., with logistic regression or deep neural networks [15, 53, 62]. Considering the close relationship between pointing selection behavior and gaze [16], we believe that neural networks driven by the combination of gaze, head, and hand data can more effectively predict raycasting endpoints in VR.

In this paper, we propose a novel Long Short-Term Memory (LSTM)-based neural network model that leverages multimodal input from hand, head, and eye movements to predict ray pointer landing poses in VR environments. Our technique's predictions are within 4.6° of the true landing position at 50% of the movement, significantly outperforming traditional kinematic methods. Additionally, we built a user-independent model, demonstrating accurate predictions for new users and achieving better generalization compared to the KTM method, which relies on personalized templates for each user.

Two user studies were conducted in this work. The first was used to collect user data and analyze the behavior of the head, hand, and eyes during the selection process, to identify the best input modalities and to empirically analyze the value of gaze input, which had not previously been used in KTM methods. The second study tested the generalizability of our proposed model by collecting data across a range of angles and target sizes and comparing its performance with the current state-of-the-art model, the Head-Coupled Kinematic Template Matchin (HC-KTM). We further conducetd a cross-user validation and trained a general model to further our model's generalizability and reliability.

The key contributions of this work are (1) a neural network model based on multimodal input, outperforming previous work for predicting ray pointer landing poses in VR environments, (2) results and interpretations of eye and head behavior characteristics during target selection using bare hand raycasting, and (3) an open-source dataset collected from two empirical user studies, comprising 72,096 trials of bare hand raycasting selection.

2 Related Work

2.1 Raycasting Selection in VR

There are various techniques for selecting and manipulating objects in VR, with raycasting being one of the most prevalent and effective methods for 3D selection tasks, allowing users to select distant targets with minimal physical effort [2, 49, 27]. Numerous studies have explored factors affecting raycasting performance, such as rotational jitter [5], ray length [6], grip styles [7], and object-related factors [19, 61].

While raycasting is widely used due to its effectiveness, it becomes less efficient for selecting small objects that require high angular accuracy [11]. To address this challenge, various techniques have been developed, which Argelaguet and Andujar categorized into visualization, heuristic, and behavioral modifications [2]. For example, visualization techniques like the Ray Cursor [3] allow users to manipulate a cursor along the ray, while heuristic methods like redirecting rays [24] subtly redirect the ray for better precision. However, some advanced techniques can introduce learning costs and may perform poorly for selecting large objects, where selection times can even increase [60]. These methods primarily focus on enhancing the selection performance at the final phase of pointing. Our work aims to predict the landing pose for standard raycasting throughout the entire selection process, not just toward the end.

Our approach is most closely related to Henrikson et al. [31], who proposed an extension of the template-matching technique to predict the final landing pose for traditional raycasting in a 3D environment. This model uses kinematic methods to construct a template library from multimodal data (speed and distance over time sequence data from users' head and hand movements), as well as ranking and filtering to predict future selections. Yet, this approach is limited by the need for individual template libraries for each user. Our work improves the selection process by using multimodal data input combined with learned representations to predict the final landing pose without relying on personalized data collection.

2.2 Gaze-Head Input and Coordination

Gaze and head movements are naturally coordinated with hand movements, making them valuable input modalities for interactive systems [50]. They both can be used in explicit and implicit modes. Explicit gaze usage includes tasks like selecting virtual keyboard keys or navigating a UI [51, 35, 37, 48]. Implicit gaze usage reflects a user's cognitive state, with gaze dynamics correlating with visual attention and motor processes [47, 16]. Similarly, head movements can be used for tasks like head-based cursor control or as implicit input to enhance interaction [14, 41, 45, 10, 32].

Since gaze behavior inherently occurs together with eye and head movements, the coordination between these movements can significantly impact the use of HMDs, especially during search and selection tasks. Here, we review studies that investigate how the head and eyes coordinate during visual tasks. Gaze shifts are typically a multimodal input [55], generally involving both head and eye orientation [58, 22]. When the required gaze shift is relatively small (e.g., 20-30°, as seen in laboratory settings), eye movement alone may suffice [21]. However, due to neuromechanical constraints, the maximum comfortable eye movement usually does not exceed 40-45° [22, 9], and in practice, it is often less for a more comfortable experience [22, 38]. Thus, when the gaze shift exceeds approximately 20-25°, a person typically moves their head and eyes in the same direction. During this time, the gaze shift is assisted by head movement, although the timing of eye and head movements during shifts can vary depending on factors like tasks and orbital position [22]. For larger amplitude gaze shifts, head movements play an increasingly significant role, with their contributions varying based on the task [28, 52].

Previous prediction models only utilized head motion features [31], but since these works have highlighted the behavioral differences between gaze and head movements, incorporating gaze information into multimodal fusion may lead to better model performance. Moreover, due to the strong correlation between gaze behavior and manual input actions such as pointing [15], specifically, gaze typically precedes hand movements in similar tasks by a few milliseconds [23, 30], this pattern suggests that eye movement can serve as a reliable predictor of short-term future hand motion. Consequently, our work investigates the impact of incorporating gaze data into the prediction model for ray landing poses and also compares the performance differences between models using various input modalities.

2.3 Predictive Metaphor During Pointing

To enhance the user experience in pointing interactions, various models have been developed. For example, Yu et al. [61] proposed the ED model, which predicts the endpoint distribution for objects selected via raycasting in 3D Virtual Environments (VEs). Using

Bayesian methods to predict the user's intended target, Bi et al. [8] treated object selection as a probabilistic inference process. Wei et al. [57] combined these approaches in a probabilistic model for gaze-based selection in Augmented Reality (AR), improving target selection accuracy. These techniques effectively reduce selection errors in dense scenes or with compact UI elements.

Similarly, studies have focused on continuously predicting hand movements in 3D VEs. Casallas et al. [12] predicted moving object selection by analyzing head and hand movements, while Gamage et al. [25] demonstrated a regression model with kinematic methods. Clarence et al. [13] utilized an LSTM model that combines hand trajectories with gaze data to forecast future interaction targets. More recently, biomechanical simulations have been used to generate trajectory data for training neural networks in object selection [44]. However, these methods often require prior knowledge of scene objects, turning it into a discrete classification problem.

Given the rich information in hand and gaze dynamics, many studies focus on multimodal fusion to enhance interactive experiences. For instance, Wolf et al. [59] analyzed real-time hand-eye coordination to predict potential hand-action errors during target selection. Hallgarten et al. [29] recently proposed the GEARS framework, using self-supervised learning to integrate hand and gaze data into multi-purpose embeddings, simplifying temporal data processing for model development.

Our research focuses on predicting the final landing pose of a ray in VR. The closest related work is by Henrikson et al. [31], who extended Kinematic Template Matching (KTM) [46] to Head-Coupled Kinematic Template Matching (HC-KTM) in 3D. However, their approach relies on user-dependent templates, requiring extensive behavioral data for template construction.

In the context of predicting 2D mouse click positions, Wei et al. [56] used LSTM neural networks with a hybrid loss function to outperform kinematic methods such as Kinematic Template Matching [46] and Kinematic Endpoint Prediction (KEP) [39]. We aim to explore whether this performance improvement can be replicated in 3D environments. Our work incorporates additional 3D modality data into a neural network model that is target-agnostic and uses raw time-series input, enabling a user-independent approach that eliminates the need for individual templates.

3 Proposed Model and Study Design

3.1 Prediction Model

This study aims to develop a learning-based model to predict the ray landing pose during point-and-select interactions in VR environments using ray-casting techniques (either controller- or barehand-based). To predict the final ray landing pose, which consists of two components—position (the starting point of the ray) and orientation (the direction in which the ray is pointing), we need two sets of output values: the origin (3D coordinates) and the direction of the ray (yaw axis and pitch axis). To address this challenge, Henrikson et al. [31] extended 2D Kinematic Template Matching (KTM) to 3D environments, resulting in the Head-Coupled Kinematic Template Matching (HC-KTM) method. Their model outputs scalar values for the linear and angular movement amplitude of the ray. The linear movement direction is then calculated based on the current coordinates of the input device relative to the initial position. The rotation axis is determined by the cross-product of the current and initial orientations. With these directional and scalar values, the final position and orientation are then predicted separately.

3.1.1 Data Preparation

To train and test our model, we collected motion data from the HMD, gaze, and hand during a barehand pointing and selection task with raycasting. This was a reciprocal three-dimensional pointing task.(see 4). For our analysis, we derived five velocity profiles: (1) hand position, which measures the change in the hand's origin over time; (2) HMD position, capturing the change in the HMD's origin over time; (3) hand angle, reflecting the change in the angle of the hand's forward-facing vector over time; (4) HMD angle, indicating the change in the angle of the HMD's forward-facing vector over time; and (5) gaze angle, representing the angular displacement between consecutive gaze samples divided by the time interval. Abnormal gaze velocity values exceeding 800°/s were removed [17], and missing values were linearly interpolated. These velocity sequences, along with the distance traveled from the starting point over time, served as model inputs.

Following previous work, we applied Gaussian smoothing to reduce noise [31, 25]. Each user's features were normalized using Min-Max scaling within participants, and the total distance traveled was used as the label for each sequence. The aforementioned prediction methods generate two sets of output values. To provide such ground truth for each user's selection process, our model calculated the magnitude of the hand's linear movement distance and the rotational angle from the start of the selection behavior to the moment of target selection for each task trial.

Since the completion time for each selection task varies, we employed post-padding with a value of -10 to fill features beyond the completion time of each trial, to match the duration of the longest trial. A masking layer in the first layer of our network ignores these values [18, 56]. To enable our network to recognize partial (incomplete) sequences, we segment the data to produce partial sequences for data augmentation [13, 56]. We record data at 90 Hz, i.e., every 11 ms. For prediction, our minimum *unit time step* is 55 ms, with increments equal to this step. For each trial, we create multiple instances (partial sequences) by starting from the initial time step in increments of *unit time steps* until the entire duration of the trial is covered.

Due to the strong capabilities of neural networks for feature modeling and generalization and following previous work [62, 56, 13], using raw features can often yield good results. Previous kinematics-based models have demonstrated strong performance using speed and distance over time as input features [25, 31, 39]. We thus investigate whether learning methods can outperform kinematics-based methods using similar inputs. The selection of input features for these methods builds upon common patterns observed in human experimental pointing tasks. For example, cursor movement during selection typically comprises a ballistic phase (high speed) and a corrective movement phase (low speed) [43]. Additionally, the minimum jerk law [20] suggests that humans aim to minimize jerk (the derivative of acceleration) to produce smooth movements. These findings suggest predictable patterns in pointing tasks, observable in various parameters such as velocity and peak velocity. Therefore, rather than relying on engineered features, our work uses raw velocity sequences and the distance traveled from the starting point as inputs.

3.1.2 Model Inputs and Data Integration

At each time step t, we record a set of features for each modality, including rotation angles (accumulated from the starting point up to time t) and angular velocities. For the HMD and the hand, we also include linear velocities and the distance traveled from the starting point. For the gaze modality, we discarded the positional velocity because the changes in the 3D coordinates of the eye's origin were nearly identical to those of the HMD, leading to redundant performance. All of these signals are concatenated into a single feature vector: $\mathbf{x}_t \in \mathbb{R}^d$, where d is the total number of features. The resulting sequence of feature vectors is then fed into the LSTM model as: $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$.

3.1.3 Model Structure and Training

We aim to develop a model that predicts the final landing pose of a user's ray based on multimodal velocity and motion distance time-

series data. We frame this as a time-series regression problem for our proposed prediction method. Prior work has shown that LSTM neural networks can achieve good performance in learning spatiotemporal patterns from long-term series data [13, 46]. LSTMs are capable of handling motion sequences with variable lengths and can learn the implicit mapping between velocity, distance sequences, and the final movement distance. Therefore, we adopted an LSTM-based model for our prediction task.

We used two types of modeling approaches. First, we constructed a within-participants model. We stratified all of each user's trials by the task's θ value (the angular distance between the start and end targets, see Section 4.3) into a 4:1 ratio, with 80% used for training and the remaining 20% for testing. Additionally, 20% of the training set was used as a validation set to monitor loss changes during training and ensure our model does not over- nor underfit. However, with this approach, each user had their own model, and thus these models were not participant-independent. Consequently, we also trained a second set of models in a participant-independent manner. Here, we used data from some participants to construct the training set and data from other participants for the testing set, while ensuring that no participant belonged to both sets.

We implemented a four-layer stacked LSTM using TensorFlow's Keras API to process fixed-length sequences of duration T (padded if shorter). The input shape is (T, d), where d is the feature count, and a `Masking` layer (`mask_value=-10`) discards padding. The LSTM stack, with hidden sizes of 128, 64, 32, and 16, processes inputs sequentially, with the first three layers returning intermediate outputs and the final layer returning the last hidden state. A `Dropout` layer (`rate=0.4`) follows, then a `Dense` layer (32 units, `sigmoid` activation), another `Dropout` layer (`rate=0.2`), and a final `Dense` layer (`activation='linear'`) outputs the motion range. The network is trained with Mean Squared Error loss and the Adam optimizer [36] for 30 epochs (batch size = 32), using validation loss to select the best-performing epoch for testing [13].

3.1.4 Evaluation Metrics

We used the following metrics to compare our model with the baselines. *Angular Distance* measures the accuracy of each prediction. Angular Distance refers to the angle between the direction of the output predicted ray and the Perfect Ray. The direction of the Perfect Ray is defined as the ray originating from *the predicted origin* and pointing toward the center of the target for the current trial. This concept is similar to the pixel error used in 2D prediction models [46]. *Target Hit Rate* measures how often predictions fell within the bounds of the goal target at a specific point in time. At different points in time during the pointing motion, we check if the predicted pointing direction would intersect with the target area.

So far, HC-KTM is the only technique we know that predicts the ray landing pose during the selection process. We treat this technique as a non-naïve baseline. Additionally, we also use a naïve baseline, which does not use prediction (i.e., it uses the current actual ray pose), following previous work [25]. In this case, the angular distance is calculated using the current hand position and orientation information and the Perfect Ray.

3.2 Study Outline

We conducted a first user study to build the model and identify the best input modality. In the first study, we collected data and analyzed the behavior patterns of the head, hand, and eye modalities during selection tasks. Specifically, we examined how factors such as movement distance and object size influence the motion distances of the head, hand, and eyes. We also compared the differences between gaze shifts and HMD rotational movements. Using the collected data, we preliminarily identified the most suitable input modalities for our features. Additionally, we integrated gaze shift information into the well-established HC-KTM model to investigate whether incorporating this modality would enhance that model's performance. Finally, we compared the performance of our model with that of kinematic ones.

We evaluated and validated our model in a second study. In this study, we investigated more general and more continuous task parameters (specifically θ) to further assess our model's predictive performance on continuous values. In addition to evaluating the prediction accuracy for each person using their own model, we performed two additional experiments to test the model's generalizability to other users. We first used a cross-user model trained on one user's data to predict the behavior of other users. Then we trained a larger, generalized model using input data from multiple users to determine if it could accurately predict the behavior of new users.

4 User Study 1: Data Collection and Preliminary Evaluation

4.1 Participants, Apparatus, and Materials

We recruited sixteen participants (6 females, 10 males), all 20 to 25 years old (*mean* = 22, *SD* = 1.3) from a local university. All participants had normal or corrected-to-normal vision and reported clear visibility of objects in the scene. Additionally, all participants identified as right-handed. On average, participants self-rated their familiarity with the VR system as 4 (*SD* = 3.2) on a 7-point Likert scale. To immerse the users into the 3D VE, we used a Meta Quest Pro VR headset, with 106° horizontal and 96° vertical FoV (field-of-view) and a resolution of 1800 × 1920 per eye. The experimental program, developed using C#.NET, the Oculus integration, and the Unity3D game engine, was run on a PC with an Intel Core i7-8850H CPU and an NVIDIA GeForce RTX 2080 graphics card. The PC and headset were connected via a three-meter USB-C cable. Throughout the study, the HMD display refresh rate was set to 90 FPS (frames per second) to ensure smooth screen rendering and rendering consistency. The positions and orientations of the three tracked modalities were recorded at the same, fixed rate of 90 Hz.

4.2 Experimental Task and Procedure

Each trial presented two objects within the scene: a start and an end spherical target. Participants were tasked with alternately selecting these two objects, repeatedly moving the ray between the two targets. At the beginning of each trial, the start sphere changed color to yellow, signaling it as the target candidate for selection, while the opposing sphere turned grey to indicate the position of the subsequent target. When the cursor successfully intersected the target, that sphere turned green, at which point the user could perform a selection operation. Selections made without the ray intersecting the target were recorded as errors. Only when the current target was correctly selected would the trial switch to the next one, where the two spheres in the scene swapped colors (the opposing sphere became the target sphere for the current trial, turning yellow). The time taken to move from the initial position to the target for selection is defined as the movement time (MT). After a predefined number of reciprocal selections, the targets would shift to another pair of opposite positions.

Participants used their dominant hand for selection, employing a bare-hand-based ray-pointing mechanism. The direction and orientation of the rays were controlled by the participant's hand movements. A pinch pose between the index finger and thumb served as the trigger for selection actions, similar to the standard trigger function of a VR controller.

Previous studies recommended maintaining a 4% error rate in the selection task [31]. However, to better simulate a real application scenario, we did not require participants to sustain a specific error rate [61, 57]. During the main task, no other object was rendered in the scene except for the two spheres. The participants were asked to use both accuracy and speed in completing the task.

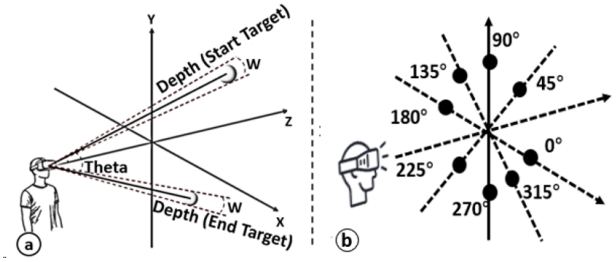

Figure 2: Experimental setup illustrating the manipulated variables and target positions. **(a)** The 3D view illustrates the *Depth, θ, Position*, and *Width* of the targets. (in this example, the ϕ values are 90° and 270°.) **(b)** This illustrates all target positions (ϕ) around the user from 0° to 315° in increments of 45°.

4.3 Design and Procedure

Our study used a $3_D \times 6_\theta \times 8_\phi \times 2_W$ within-subjects design. Like previous work [31], Our experiment manipulated four independent variables characterizing each target's position and visual size: **Depth** (D), including $Depth_{start}$ (D_e: 3m, 6m, 9m) and $Depth_{end}$ (D_s: 3m, 6m, 9m); **Theta** (θ): 10°, 15°, 20°, 25°, 50°, and 75°; **Position** (ϕ): ranging from 0° to 315° in increments of 45°; and **Width** (W): 4.5° and 9°.

In this study, we used angular dimensions instead of Euclidean radii to ensure consistent visual size and selection difficulty across different target distances from the user, as appropriate for raycasting [4]. In our design, we considered the user's eyes as the reference point (origin). Then, *Depth* is the distance of the target center from the origin and *theta* quantifies the angular distance between the start and end targets.

Previous work presented similar experiments [31, 25], yet the θ values in these studies did not balance head and eye coordination. For θ values greater than 30°, both head and eye movements significantly contribute to gaze shifts, while for values less than or equal to 25°, head movements contribute only about 10% to gaze shifts [55]. This imbalance could affect the weighting of different modalities in the prediction algorithm, as larger θ values might increase the influence of HMD data. In our study, we thus also included three θ values smaller than 25° to observe whether the HMD modality remains a good indicator for prediction in scenarios requiring minimal head movement.

Target Width (W) specifies the angular size of the target sphere, ensuring that visual size and selection difficulty remain constant, independent of the target's distance from the user. *Position (ϕ)* is the azimuth of the target, ranging from 0° to 315° in 45° increments. In our reciprocal task, the start and end positions are opposite.

The experiment was organized into $3_{D_e} \times 3_{D_s} \times 6_\theta \times 2_W = 108$ blocks, each representing one of the 108 combinations of the variables D (D_e and D_s), θ, and W, presented in random order. Within each block, participants completed four sets of trials, each involving a pair of opposing ϕ angles (e.g., (0°, 180°)), necessitating a total of eight selection operations. In each block, the start and end targets were positioned oppositely in ϕ, sharing the same W and θ values, though their D values could differ. Each participant completed 3456 trials during the session (54 blocks $\times$ 4 repetitions $\times$ 8ϕ). In total, the study collected data for $(3D_e \times 3D_s =)\ 9D \times 6\theta \times 2W \times 8\phi \times$ 4 repetitions $\times$ 16 participants = 55,296 trials.

Eye-tracking data was acquired through the Oculus Integration Movement SDK. Additionally, upon the completion of each trial, we recorded the linear and angular movement amplitude of the three tracked modalities (hand, head, eye) used during the trial, the duration of the trial, and whether the trial was completed with errors.

The entire duration of the study was approximately 60 minutes per participant. Before commencing the experiment, participants provided demographic information via a questionnaire. The experimenter then introduced the VR headset and explained the tasks. Participants were fitted with the Head-Mounted Display (HMD) and underwent the Oculus's default eye calibration process. To reduce unfamiliarity bias, an unlimited training session preceded the formal trials, allowing participants to familiarize themselves and practice with the device and controls for at least 5 minutes. To mitigate potential biases from mental and physical fatigue, the process was segmented into several blocks (see Section 4.3). Participants were permitted to take off the HMD and take breaks between blocks and were required to use the VR headset's built-in recentering mechanism upon resuming to recalibrate their position.

4.4 Data Pre-processing

As is common practice, outlier trials where the movement time exceeded two standard deviations beyond the mean movement time for the same θ and W combination were removed, accounting for approximately 3.01% of the total number of trials. Additionally, for trials marked as errors, which comprised about 9.05% of the total, only data collected before the user's pinch action was analyzed. For gaze vector data, we used the eye-in-world movement measure. Additionally, since we are primarily concerned with movement velocity and angular rotation, which are nearly identical between the left and right eyes, all subsequent calculations were based only on the gaze data collected from the left eye.

We analyzed a total of 53,632 data points. These data were then processed according to the methods described in 3.1, including the calculation of velocity profiles, generation of incomplete feature sequences, and the division into training and testing sets.

4.5 Results and Analysis

4.5.1 Movement Time

We conducted a repeated-measures ANOVA for each variable potentially influencing MT, applying the Greenhouse-Geisser correction to adjust the degrees of freedom when the assumption of sphericity was violated. The analysis revealed that both θ (θ, $F_{2.125,31.871} = 263.344, p < .0001$) and width ($W$, $F_{1,15} = 1044.967, p < .0001$) significantly affect MT. No significant main effects were observed for depth (D) on MT ($F_{2,30} = .345, ns$). The interaction between θ and W was also significant for MT ($F_{3.253,48.793} = 14.517, p < .0001$).

Our empirical study confirms that both target width (W) and amplitude (θ) significantly influence MT, both individually and in combination. However, depth (D) did not have a significant effect. To further substantiate these results, we conducted a Fitts' law [19] analysis using our data, which included eleven different indices of difficulty (we had 2 W values and 6 θ values, but for W=4.5 and θ=10 and W=9 and θ=20, the index of difficulty overlapped, yielding only 11 unique indices). The analysis indicated that our data closely align with the angular derivation of Fitts' law, with an R^2 value of 0.9783, see Figure 3.

$$MT = a + b \log_2 \left(\frac{\theta}{W} + 1 \right) \quad (1)$$

Although we used barehand ray-casting, instead of a controller, and mid-air pinch for confirmation, we can confirm that the angular deviation of Fitts' law applies to ray pointing in VR environments, consistent with previous work [31, 6, 4].

4.5.2 Head and Hand Angular Movements, Gaze Shifts

To gain a better understanding of user behaviors within the context of raycasting and barehand-based tasks in 3D VEs, we calculated the cumulative angular distance, which serves as an indicator of the distance traveled for each selection, providing insights into the conjunction between different modalities (see Table 3). This was

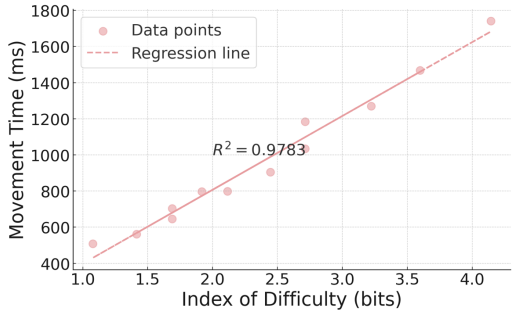

Figure 3: The data for Movement Time shows a high fit with the index of difficulty computed by the angular derivation of Fitts' Law.

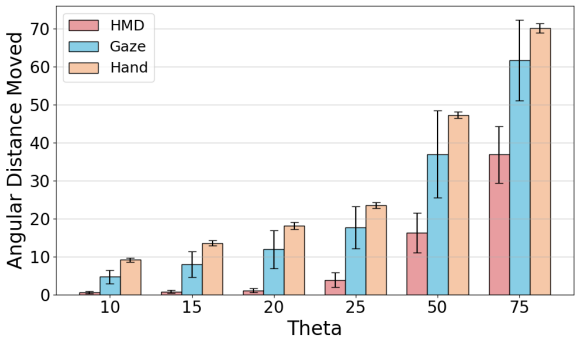

Figure 4: The Hand, HMD, and Eye angular movements for the Angular Distance to the target. Error bars represent standard deviation.

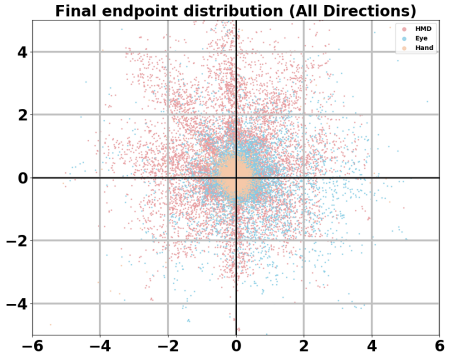

Figure 5: Final endpoint distribution of Hand (orange), HMD (red), and Eye (blue) during target selection confirmation. The origin represents the center of the target.

calculated by measuring the angle between the forward vectors of the modality at two points in time. The results of RM-ANOVAs revealed that both the eye and head were significantly affected by θ, W, and ϕ. Additionally, the traveled distance of the hand was influenced by θ, W, and D. Interaction effects were observed across all three modalities – eye, head, and hand. Specifically, $\theta \times W \times \phi$ affected all three modalities. Furthermore, the factor $D \times \theta \times W$ had a statistically significant impact on gaze. Other interaction effects were identified within individual modalities, such as $\theta \times W$ and $\theta \times \phi$ on the head, and $D \times \phi$ on the hand.

In comparing angular movements among the eye, head, and hand, the head exhibited only a fraction of the angular movement observed in the hand (see Figure 4). However, the movement distances of the eye and hand were relatively similar. This is not surprising, as during selection, the hand's angular movement is crucial for aligning with the target's boundaries, while the eyes move to focus on the selected object. In contrast, the head only needs to move enough to ensure the target falls within the user's field of view. It is worth mentioning that at θ values of 50° and 75°, the head's rotation became notably more pronounced, whereas, for movements less than 25°, the head's contribution was minimal. This observation aligns with the findings of Land and Tatler [38], who noted that head movement is often necessary when the target's distance from the center of the head exceeds 30°, providing additional support to the eyes. Thus, at θ values of 50° and 75°, head movement in conjunction with eye movement becomes a frequent requirement for target selection. Additionally, larger θ values mean that head rotation contributes more to gaze shifts [22]. Lastly, we observed a large standard deviation in gaze shift and HMD rotation, as the motion of the eyes relative to the head depends on user preferences. Some users prefer to move their heads more, while others prefer to move their eyes more [55].

These relationships are further illustrated in Figure 5 through scatter plots of the final endpoints of the eye, head, and hand when targets were acquired. Nearly all of the hand's endpoints are clustered near the target's center. Although the eye's endpoints also tend to converge towards the center, they are not as tightly grouped as the hand's endpoints and some points are farther from the target center. This pattern likely stems from multiple factors, including eye-tracking noise and the nature of the fovea, the central part of the retina responsible for sharp, detailed vision. Unlike a precise point of contact for the hand, the fovea is a small region rather than a single point. This means that the eye can achieve high-acuity vision of a target even when the gaze is not exactly centered on it, resulting in a somewhat wider distribution of endpoints compared to the hand. In contrast, the endpoints of the head exhibited significant variation between each trial, with the head moving relatively less.

4.5.3 Velocity Profiles

Velocity profiles are crucial inputs for our model, so it is important to observe how the velocities of different modalities change over time with varying angular movements (see Figure 6). To produce the velocity profiles in Figure 6, we first computed instantaneous velocities through the angular change between consecutive forward vectors, divided by the sampling interval. After applying Gaussian smoothing for denoising, we resampled each trial's velocity signal at 20 Hz. Trials were then grouped by their angular distance θ, and we computed an average movement time for each group across all participants. We aligned the velocity trajectories within the same θ group to this average timescale, and finally took the mean velocity at each time point across all trials to form the aggregated profiles.

The velocity profiles show distinct patterns across different angular movements. For θ values greater than 25°, the HMD shows significant speed differentiation, while for smaller angles, the curves tend to overlap. In contrast to previous work,we added three smaller θ values, we still observed that the divergence in the hand speed is initially only very slight [33].On the other hand, gaze shows some differentiation in the beginning, but this difference becomes even more pronounced after the ballistic phase.

Consistent patterns were identified across all modalities, with higher θ values associated with increased speed. The velocity curves aligned with the two-stage movement model [43]: a high-speed ballistic phase followed by a slower correction phase.

Notably, gaze demonstrates promise as an earlier predictor, reaching peak velocity at 40.17% of movement duration (SD = 21.44%), compared to the HMD at 43.66% (SD = 27.40%) and hand at 44.81% (SD = 15.19%). Using a dual-threshold velocity

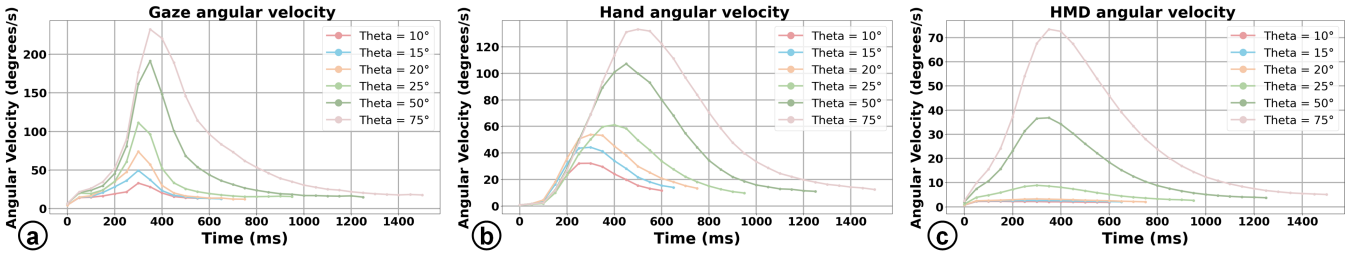

Figure 6: Velocity-time curves of the three modalities across different angles.

heuristic [1], the final saccade ends at 46.37% (SD = 15.02%) [30], confirming gaze as an effective early indicator of landing positions.

4.5.4 Multimodal Comparison

To better understand how each input modality aids our model in making predictions, we trained seven different models with various inputs. These included using the velocity of the Gaze, Head, or Hand individually, all combinations of two features, and finally, a model where all features were used as inputs.

We aim to achieve better performance in early-stage prediction, so we chose to use the angular error at 40% of the selection movement as the primary metric for performance comparison. For single modalities, eye movement consistently performed better before reaching 40% of the movement. At 40%, the angular error for the hand was 9.29°, for the HMD was 9.16°, and for the eye was 8.23° (See Figure 7).As time progresses, the hand modality gradually becomes the most effective, likely because hand movements need to persist until the end of the selection process.

Incorporating multimodal data improved early-stage prediction accuracy across the board. The least effective multimodal input, combining HMD and eye data, had an angular error of 8.30° at 40%, almost equivalent to that of the single HMD input model. We attribute this to the absence of hand data, as the differences between head and eye movements become less significant after 50% of the movement. The likely reason is that while the hand continues to move, the head and gaze rotations may have already stopped, with the user having already brought the target into their field of view.

The models combining either HMD or eye with the hand had angular errors of 7.54° and 7.42°, respectively. The tri-modal model, incorporating hand, HMD, and eye data, had the smallest angular error of 7.33°, albeit by a slight margin. This marginal advantage could be due to using gaze shift information instead of rotational information relative to the head, leading to some redundancy as eye movement speed data overlaps with HMD movement information.

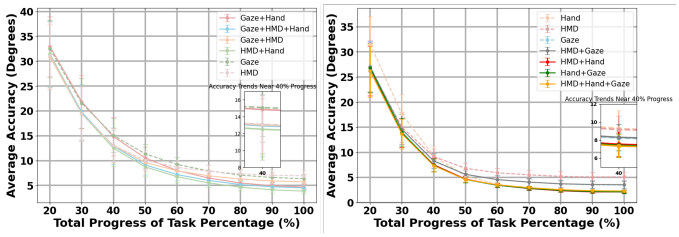

Figure 7: The prediction accuracy for all input models at different stages of the selection with the LSTM-based approach and KTM Method.

In addition to selecting the best input modalities for our model, we also investigated if incorporating gaze data would improve the predictive performance of the template matching technique (HC-KTM, see Figure 7). During the selection of input modalities for our LSTM-based model, we found that early in the movement, the gaze-plus-hand model outperformed the HMD-plus-hand model. This led us to explore whether this effect would also occur in kinematics-based models or a three-modality Template Matching approach. To this end, we implemented Henrikson et al.'s algorithm [31], which is the only model that uses kinematics for endpoint prediction in 3D environments.

Specifically, we manipulated a total of six parameters. Firstly, we selected the *top-n* templates for different values of n and then compared the cumulative angular error to make a selection. Coincidentally, we ended up with the same value of 7 for n as the original study. Other variables that needed fine-tuning were the weights of different modal scores in the score function, which required us to consider each modality's input into the KTM method independently. We then compared their performance at 40% of the selection process and used interpolation to obtain the final values. The final parameter list for the HMD's positional velocity, angular velocity, Hand's positional velocity, angular velocity, and gaze's angular velocity was $0.94, 1, 0.55, 0.5$, and 0.8. We discarded the gaze's positional velocity because the changes in the 3D coordinates of the eye's origin were nearly identical to those of the HMD, leading to redundant performance.

Previous research has already demonstrated the utility of modality comparisons. Therefore, we only present the accuracy curves of the untested Gaze input over time, along with HMD Angular as a single-modality comparison, and dual-modality and tri-modality inputs. We found that models related to Gaze information (Gaze-only and Gaze combined with Hand) did not perform as well as expected. In the early stage (10%-40%), their performance was even lower than that of the model with HMD as the sole input modality. This could be due to the high-speed nature of the Gaze, where the velocity differences across angles are minimal during the initial ballistic phase. Since the KTM method uses an accumulative approach, the algorithm can only select templates with similar movement amplitudes during the late correction phase. The most effective model was still the HC-KTM model (using HMD and Hand information). Adding Gaze information to this model did not improve accuracy and even caused some performance degradation. Similarly, the dual-modality input of Gaze and HMD did not enhance accuracy in the early stage and only improved during the latter correction phase. We believe this is because Gaze shifts are partly coupled with head movements, leading to redundant information between these modalities. Due to this redundancy, the kinematic-based velocity inputs might not capture effective features.

Our initial comparison between the LSTM-based model and the KTM models showed that at the 40% mark, the LSTM model's worst-performing input (9.29°) significantly exceeded the best-performing input of the KTM model (12.49°). Also, unlike the HC-KTM method, incorporating gaze velocity as input can achieve higher accuracy, indicating that the LSTM-based network approach can better identify implicit mappings between velocity time series data and hand movement magnitude. Since we prioritize early pre-

diction, in the 20% to 40% movement stages, the input from all three modalities proved to be superior. Therefore, in our subsequent validation, we opted to use the tri-modal input configuration for our model.

5 User Study2: Model Validation

This study aimed to test the robustness and reliability of an LSTM-based prediction model by increasing the complexity of the pointing selection task in VR environments. Given that the experimental parameters D, θ, ϕ, and W used in 4 were discrete, this study uses continuously changing values of these parameters to more dynamically specify the target position. Furthermore, since the truth for the model is determined by the angular and linear movement distances of a user's hand movement during a trial, both of which are largely influenced by the selection of θ (although not entirely equal to θ) we expanded the task parameters in Study 2 to test our model's robustness in facing non-fixed θ scenarios, ensuring the model could predict selections under various movement distances (θs). The data collected were the same as in Study 1.

5.1 Participants, Apparatus, and Materials

We recruited 8 new participants from the same pool as Study 1 for this study. The apparatus and materials were the same as in Study 1.

5.2 Experimental Task and Measurements

We used the same task as in Study 1, with a repeated-measures within-participant design. As with the first experiment, the task was a reciprocal three-dimensional pointing task with no distractors. We controlled θ continuously to encompass all integer values from 15° to 84° (inclusive). These 70 distinct values were presented five times each throughout the experiment, with the presentation order randomized across the sets of trials. The 350 resulting trial combinations were arranged as 50 blocks, each with 7 distinct target pairs. Each trial combination then involved 6 reciprocal selections. All other variables were set randomly, with the upper and lower limits the same as in Study 1. The experiment was conducted in a single session lasting around sixty minutes. To prevent fatigue and provide opportunities for breaks, the session was divided into 50 equal blocks. Participants could rest between blocks and, upon returning, had to recenter the HMD for recalibration. This setup resulted in a total of 2100 trials per participant (50 blocks x 7 pairs x 6 selections). Before the main session, participants completed practice trials to ensure they were familiar with the task.

5.3 Results

We processed all the collected trajectory data for analysis, excluding those with movement times greater than two standard deviations above the mean for the same theta and width settings.

5.3.1 Comparison with Other Prediction Models

After identifying the best input modality combinations, we aimed to compare the accuracy of our model with the current state-of-the-art model on the dataset collected in this study. We again used the HC-KTM algorithm with the same parameter settings as described in Section 4.5.4. As the baseline, we used the angular discrepancy between the actual projected ray and the target object without prediction.

The performance analysis of the LSTM model discussed here uses the data from Study 2. Our model's angular accuracy was superior to KTM from the very start of the movement, as shown in Figure 8. As in previous work [31], we used the 40% mark as a baseline for comparison, which is still in the early stages of selection. At this stage, our method exhibited an angular error of 7.06°, compared to 13.28° for HC-KTM-7(Head Coupled Kinematic Template Matching with Top-n templates parameter 7), and 24.77° for the naive baseline with no prediction, showing an approximate 1.9x improvement over HC-KTM and 3.5x over the baseline. For the objects with the minimum angular width of 4.5° that we used here, this is an encouraging result. Similarly, regarding hit rate (see Table 1), our method consistently outperformed HC-KTM between 50% and 90% of the movement. Also, unlike previous methods that were less accurate than the baseline at the end of the movement [63], our method demonstrated higher accuracy than the baseline at that time, too.

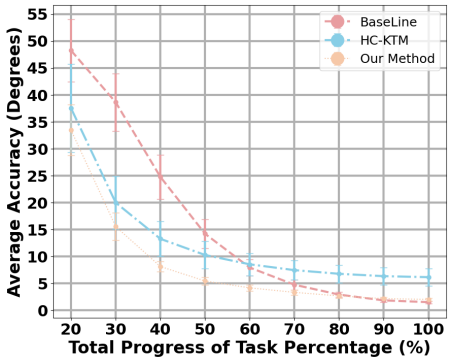

Figure 8: The prediction accuracy for the LSTM network, KTM method, and baseline at 10% increments of the movement.

Distance Travelled (%)	Our Method	HC-KTM
50%	34.6%	20.8%
60%	44.4%	26.8%
70%	58.9%	32.9%
80%	76.0%	37.3%
90%	88.0%	40.0%

Table 1: The percentage of predictions where the predicted ray hits the intended target for our method and the HC-KTM method.

5.3.2 Prediction performance on new users

Previous template matching methods and the models we trained are highly personalized, meaning one would need to collect data for each user to train a model specifically for them. We thus test the generalizability of our model by training it on data from multiple users to create a more general model. This approach allows new users to use our model without us needing to collect data for training, thus enhancing usability and making adoption easier.

Toward this goal, we selected data from six random participants from Study 2 as the training set, i.e., 12,450 trajectories, and used the data from the remaining two participants as the test set. The results showed that at 40% of the movement process, the angular accuracy reached 9.97° (see details in Table 2). This is nearly identical to the average accuracy of 9.50° when each user's performance was predicted using their own template. This indicates that our method supports a "plug-and-play" scenario, where a prediction model can be pre-trained on the data of other users and then applied to predict the behavior of new users *without additional training*.

We also conducted a cross-participant experiment, where models trained on each participant's data were used to predict the data of other participants (see Figure 9). Generally, the model achieved the best result for each user with models trained on their own data. Some users, due to unique behavioral patterns, performed poorly when using models trained on others' data, such as Participant 7. Interestingly, Participant 8's model performed better on other users' data than on their own. In real-world scenarios, we thus, overall, still recommend using models trained with participant-split data.

Distance Travelled (%)	Accuracy	Target Hit Rate
40%	9.97°	15.4%
50%	6.18°	26.0%
60%	4.64°	38.2%
70%	3.93°	50.2%
80%	3.29°	60.8%
90%	2.96°	67.8%

Table 2: Prediction performance with participant-independent model.

Data Source \ Model Source	P1	P2	P3	P4	P5	P6	P7	P8
P1	7.26	11.16	9.11	9.26	8.29	10.45	13.11	7.36
P2	9.64	8.72	11.14	12.75	11.17	12.08	11.83	9.72
P3	8.93	12.67	8.36	8.38	9.13	9.45	15.79	8.94
P4	9.02	13.97	9.02	6.52	8.46	7.31	15.57	8.19
P5	9.51	13.97	10.61	9.76	9.02	7.66	13.89	9.34
P6	10.53	15.01	11.05	8.28	9.64	10.03	14.42	9.29
P7	11.66	11.65	12.54	14.48	12.15	15.45	9.49	11.30
P8	13.79	15.07	15.23	14.32	14.03	14.62	19.90	12.53

Figure 9: Accuracies (at 40%) for each participant's test sets (rows) when using another user's model (columns).

6 Discussion

Our work leverages an LSTM-based neural network with velocity and motion distance sequence data to predict ray landing poses in raycasting-based VR selection tasks. This approach demonstrates a 1.9x improvement in angular accuracy over kinematic methods and doubles hit rate metrics, achieving participant-independent results with robust generalization.

The improvements stem from using an LSTM-based Neural Network (NN) and incorporating gaze-related time-series data. The NN approach offers significant advantages as it allows us to focus solely on data inputs without the need to manually adjust the weights of the modality when introducing new inputs, as required in KTM. NN models can also easily integrate Boolean-type features such as saccades or fixations. Further, KTM typically requires collecting personalized templates for each user to achieve optimal performance, whereas our participant-independent NN model maintains high effectiveness without needing individualized templates. The effectiveness of gaze data can be attributed to the behavioral dynamics of eye movements in target selection tasks. Gaze movements typically precede head and hand movements [30, 23, 15]. Specifically, gaze velocity reaches its peak earlier than that of the head and the hand (see Section 4.5.3), and the final saccade ends at about 46% of the total movement time, which is consistent with previous studies [30]. This evidence further indicates that gaze position can serve as an early indicator of the landing position in target selection behaviors, thus enhancing prediction accuracy.

To integrate our system into real-world applications, there are two primary considerations. The first is detecting the onset of selection behaviors. In practical VR settings, there is no explicit "start" signal for selection gestures, unlike our controlled tasks that had clear start- and endpoints. A promising solution is to use multimodal data—such as gaze, head, and hand trajectories—to detect when a user transitions into a targeting state [62]. Once targeting is recognized, partial trajectories can be processed by or model for early endpoint prediction. A second consideration is the timing of predictions. Making predictions too early risks lower reliability, while waiting too long negates the benefit of anticipating endpoints. Our observations suggest that detecting the user's gaze peak velocity—around 40% of the movement (40.17% in our data)—provides a good balance between accuracy and timeliness, allowing the system to deliver reliable results before the movement concludes.

Our endpoint prediction system enables several practical applications that can enhance VR interaction. Here we discuss key use cases that demonstrate its potential benefits. A primary application of our system complements existing selection facilitation techniques. By providing early predictions of intended targets, our approach can be integrated with various selection assistance methods [2] to enhance interaction accuracy and efficiency. For example, our system enables dynamic adjustment of the control-display (CD) ratio based on prediction confidence. When the system can make a high-confidence prediction (e.g., after identifying gaze peak velocity), it could automatically decrease the cursor speed near the predicted endpoint region, facilitating precise target refinement [43]. Building on Shi et al.'s work [54] who showed that target expansion can reduce selection time even at 90% movement progress, our early prediction capability allows for proactive expansion at as early as 40% of movement progress, potentially further improving selection efficiency. Our system can also be integrated with endpoint distribution models [61, 57] to enhance selection accuracy. When the prediction confidence reaches a threshold, these probabilistic models can help correct initial prediction errors by identifying the most likely intended target, even before the ray physically intersects with it. Beyond selection assistance, our prediction system can help reduce system latency by enabling preemptive resource allocation. In VR environments where heavy computation can impact responsiveness [34], early predictions allow for anticipatory optimization, such as initiating foveated rendering or preloading resources in predicted interaction areas. This preemptive approach can significantly improve system responsiveness in complex VR applications like design tools or simulations.

Our controlled experimental design with reciprocal tasks effectively supported precise data collection but may have induced anticipatory behaviors. Future work could explore more naturalistic scenarios with randomized targets or scenarios requiring full-body movement [55] to validate the model's robustness in real-world applications. The impact of model architecture choice appears more significant than the inclusion of additional modalities. While switching from KTM to LSTM yielded a 25.58% improvement, incorporating gaze data only improved accuracy by 1.59%. This suggests potential for exploring more sophisticated feature engineering approaches and alternative architectures. Future studies could investigate unsupervised models like GEARS [29] for feature extraction or advanced architectures such as TCNs [15]. Additionally, extending the model to predict target depth [4] could enhance its practical applications.

7 Conclusion

In this paper, we introduced an LSTM-based model that predicts the ray pointer's landing pose in virtual reality by utilizing multimodal input from the user's hand, HMD, and gaze data. Additionally, our work examined the impact of incorporating gaze information into an existing kinematic model. Whereas the kinematic method failed to capture features related to gaze velocity, our results show that adding gaze input data substantially enhances the performance of our LSTM-based model. Our new model not only surpasses traditional kinematic methods in prediction accuracy but also demonstrates good cross-user generalization without requiring extensive individual data collection. Our work opens new avenues for predicting users' interaction intentions along the selection path in VR, providing new options for future user interface design.

8 Acknowledgment

We thank our participants for their time and Pin Qian for some helpful early discussions. This work is supported in part by the National Key R&D Program of China under Grant No.2024YFB2808803.

References

[1] R. Andersson, L. Larsson, K. Holmqvist, M. Stridh, and M. Nyström. One algorithm to rule them all? an evaluation and discussion of ten eye

movement event-detection algorithms. *Behavior research methods*, 49:616–637, 2017. 7

[2] F. Argelaguet and C. Andujar. A survey of 3d object selection techniques for virtual environments. *Computers & Graphics*, 37(3):121–136, 2013. 1, 2, 9

[3] M. Baloup, T. Pietrzak, and G. Casiez. Raycursor: A 3d pointing facilitation technique based on raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2019. 1, 2

[4] A. U. Batmaz, M. Hudhud Mughrabi, M. D. Barrera Machuca, and W. Stuerzlinger. Effect of stereo deficiencies on virtual distal pointing. In *28th Symposium on Virtual Reality Software and Technology*, VRST '22, Nov 2022. 5, 9

[5] A. U. Batmaz and W. Stuerzlinger. The effect of rotational jitter on 3d pointing tasks. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, p. 1–6. Association for Computing Machinery, New York, NY, USA, 2019. 2

[6] A. U. Batmaz and W. Stuerzlinger. Effect of fixed and infinite ray length on distal 3d pointing in virtual reality. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, p. 1–10. Association for Computing Machinery, New York, NY, USA, 2020. 2, 5

[7] A. U. Batmaz, R. Turkmen, M. Sarac, M. D. B. Machuca, and W. Stuerzlinger. Effect of grip style on peripersonal target pointing in vr head mounted displays. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 425–433, 2023. 2

[8] X. Bi and S. Zhai. Bayesian touch: a statistical criterion of target selection with finger touch. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pp. 51–60, 2013. 1, 3

[9] E. Bizzi. *Eye-Head Coordination*, pp. 1321–1336. John Wiley & Sons, Ltd, 2011. 2

[10] R. Bovo, D. Giunchi, L. Sidenmark, H. Gellersen, E. Costanza, and T. Heinis. Real-time head-based deep-learning model for gaze probability regions in collaborative vr. In *2022 Symposium on Eye Tracking Research and Applications*, ETRA '22. Association for Computing Machinery, New York, NY, USA, 2022. 2

[11] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., USA, 2004. 2

[12] J. S. Casallas, J. H. Oliver, J. W. Kelly, F. Merienne, and S. Garbaya. Using relative head and hand-target features to predict intention in 3d moving-target selection. In *2014 IEEE Virtual Reality (VR)*, pp. 51–56, 2014. 1, 3

[13] A. Clarence, J. Knibbe, M. Cordeil, and M. Wybrow. Unscripted retargeting: Reach prediction for haptic retargeting in virtual reality. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pp. 150–159, 2021. 3, 4

[14] R. M. Clifford, N. M. B. Tuanquin, and R. W. Lindeman. Jedi forceextension: Telekinesis as a virtual reality interaction metaphor. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 239–240, 2017. 2

[15] B. David-John, C. Peacock, T. Zhang, T. S. Murdison, H. Benko, and T. R. Jonker. Towards gaze-based prediction of the intent to interact in virtual reality. In *ACM symposium on eye tracking research and applications*, pp. 1–7, 2021. 2, 9

[16] S. De Vries, R. Huys, and P. Zanone. Keeping your eye on the target: eye–hand coordination in a repetitive fitts' task. *Experimental Brain Research*, 236(12):3181–3190, 2018. 2

[17] S. Dowiasch, S. Marx, W. Einhäuser, and F. Bremmer. Effects of aging on eye movements in the real world. *Frontiers in Human Neuroscience*, 9:46, 2015. 3

[18] M. Dwarampudi and N. V. S. Reddy. Effects of padding on lstms and cnns, 2019. 3

[19] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954. 2, 5

[20] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5(7):1688–1703, 1985. 3

[21] J. M. Franchak, B. McGee, and G. Blanch. Adapting the coordination of eyes and head to differences in task and environment during fully-mobile visual exploration. *PLoS ONE*, 16(8):e0256463, Aug. 2021. 2

[22] E. G. Freedman. Coordination of the Eyes and Head during Visual Orienting. *Experimental brain research. Experimentelle Hirnforschung. Experimentation cerebrale*, 190(4):369–387, Oct. 2008. 2, 6

[23] M. Furtner and P. Sachse. The psychology of eye-hand coordination in human computer interaction. *Proceedings of the 3rd IASTED International Conference on Human-Computer Interaction, HCI 2008*, pp. 144–149, 01 2008. 2, 9

[24] J. Gabel, S. Schmidt, O. Ariza, and F. Steinicke. Redirecting rays: Evaluation of assistive raycasting techniques in virtual reality. In *Proceedings of the 29th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–11, 2023. 1, 2

[25] N. M. Gamage, D. Ishtaweera, M. Weigel, and A. Withana. So predictable! continuous 3d hand trajectory prediction in virtual reality. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, p. 332–343. Association for Computing Machinery, New York, NY, USA, 2021. 1, 3, 4, 5

[26] T. Grossman and R. Balakrishnan. 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*, UIST '06, p. 3–12. Association for Computing Machinery, New York, NY, USA, 2006. 1

[27] T. Grossman and R. Balakrishnan. 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*, UIST '06, p. 3–12. Association for Computing Machinery, New York, NY, USA, 2006. 2

[28] D. Guitton and M. Volle. Gaze control in humans: eye-head coordination during orienting movements to targets within and beyond the oculomotor range. *Journal of Neurophysiology*, 58(3):427–459, 1987. PMID: 3655876. 2

[29] P. Hallgarten, N. Sendhilnathan, T. Zhang, E. Sood, and T. R. Jonker. Gears: Generalizable multi-purpose embeddings for gaze and hand data in vr interactions. In *Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pp. 279–289, 2024. 3, 9

[30] W. F. Helsen, J. L. Starkes, and M. J. Buekers. Effects of target eccentricity on temporal costs of point of gaze and the hand in aiming. *Motor Control*, 1(2):161–177, 1997. 2, 7, 9

[31] R. Henrikson, T. Grossman, S. Trowbridge, D. Wigdor, and H. Benko. Head-coupled kinematic template matching: A prediction model for ray pointing in vr. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–14. Association for Computing Machinery, New York, NY, USA, 2020. 2, 3, 4, 5, 7, 8

[32] Z. Hu, S. Li, C. Zhang, K. Yi, G. Wang, and D. Manocha. Dgaze: Cnn-based gaze prediction in dynamic scenes. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):1902–1911, 2020. 2

[33] F. Hwang, P. Langdon, S. Keates, P. J. Clarkson, and P. Robinson. Cursor characterisation and haptic interfaces for motion-impaired users. In S. Keates, P. Langdon, P. J. Clarkson, and P. Robinson, eds., *Universal Access and Assistive Technology*, pp. 87–96. Springer London, London, 2002. 6

[34] R. Jota, A. Ng, P. Dietz, and D. Wigdor. How fast is fast enough? a study of the effects of latency in direct-touch pointing tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, p. 2291–2300. Association for Computing Machinery, New York, NY, USA, 2013. 9

[35] M. Khamis, C. Oechsner, F. Alt, and A. Bulling. Vrpursuits: interaction in virtual reality using smooth pursuit eye movements. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, AVI '18. Association for Computing Machinery, New York, NY, USA, 2018. 2

[36] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. 4

[37] M. Kumar, T. Winograd, and A. Paepcke. Gaze-enhanced scrolling

techniques. In *CHI'07 Extended Abstracts on Human Factors in Computing Systems*, pp. 2531–2536, 2007. 2

[38] M. Land and B. Tatler. *Looking and Acting: Vision and eye movements in natural behaviour*. Oxford University Press, 07 2009. 2, 6

[39] E. Lank, Y.-C. N. Cheng, and J. Ruiz. Endpoint prediction using motion kinematics. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 637–646, 2007. 3

[40] F. Lebrun, S. Haliyo, and G. Bailly. A trajectory model for desktop-scale hand redirection in virtual reality. In *IFIP Conference on Human-Computer Interaction*, pp. 105–124. Springer, 2021. 1

[41] X. Lu, D. Yu, H.-N. Liang, W. Xu, Y. Chen, X. Li, and K. Hasan. Exploration of hands-free text entry techniques for virtual reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 344–349, 2020. 2

[42] M. N. Lystbæk, P. Rosenberg, K. Pfeuffer, J. E. Grønbæk, and H. Gellersen. Gaze-hand alignment: Combining eye gaze and mid-air pointing for interacting with menus in augmented reality. *Proceedings of the ACM on Human-Computer Interaction*, 6(ETRA):1–18, 2022. 2

[43] D. E. Meyer, R. A. Abrams, S. Kornblum, C. E. Wright, and J. Keith Smith. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological review*, 95(3):340, 1988. 3, 6, 9

[44] H.-S. Moon, Y.-C. Liao, C. Li, B. Lee, and A. Oulasvirta. Real-time 3d target inference via biomechanical simulation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–18, 2024. 1, 3

[45] K. Nickel and R. Stiefelhagen. Pointing gesture recognition based on 3d-tracking of face, hands and head orientation. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, ICMI '03, p. 140–146. Association for Computing Machinery, New York, NY, USA, 2003. 2

[46] P. T. Pasqual and J. O. Wobbrock. Mouse pointing endpoint prediction using kinematic template matching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, p. 743–752. Association for Computing Machinery, New York, NY, USA, 2014. 3, 4

[47] A. Pastukhov and J. Braun. Rare but precious: microsaccades are highly informative about attentional allocation. *Vision research*, 50(12):1173–1184, 2010. 2

[48] K. Pfeuffer, J. Alexander, M. K. Chong, Y. Zhang, and H. Gellersen. Gaze-shifting: Direct-indirect input with pen and touch modulated by gaze. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 373–383, 2015. 2

[49] I. Pouypyrev and T. Ichikawa. Manipulating objects in virtual worlds: Categorization and empirical evaluation of interaction techniques. *Journal of Visual Languages & Computing*, 10(1):19–35, 1999. 2

[50] P. Qvarfordt. *Gaze-informed multimodal interaction*, p. 365–402. Association for Computing Machinery and Morgan & Claypool, 2017. 2

[51] V. Rajanna and J. P. Hansen. Gaze typing in virtual reality: impact of keyboard design, selection method, and motion. In *Proceedings of the 2018 ACM symposium on eye tracking research & applications*, pp. 1–10, 2018. 2

[52] S. Saeb, C. Weber, and J. Triesch. Learning the optimal control of coordinated eye and head movements. *PLOS Computational Biology*, 7(11):1–12, 11 2011. 2

[53] N. Sendhilnathan, T. Zhang, B. Lafreniere, T. Grossman, and T. R. Jonker. Detecting input recognition errors and user errors using gaze dynamics in virtual reality. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–19, 2022. 2

[54] R. Shi, Y. Wei, Y. Li, L. Yu, and H.-N. Liang. Expanding targets in virtual reality environments: A fitts' law study, 2023. 9

[55] L. Sidenmark and H. Gellersen. Eye, head and torso coordination during gaze shifts in virtual reality. *ACM Trans. Comput.-Hum. Interact.*, 27(1), dec 2019. 2, 5, 6, 9

[56] D. Wei, C. Yang, X. L. Zhang, and X. Yuan. Predicting mouse click position using long short-term memory model trained by joint loss function. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21. Association for Computing Machinery, New York, NY, USA, 2021. 2, 3

[57] Y. Wei, R. Shi, D. Yu, Y. Wang, Y. Li, L. Yu, and H.-N. Liang. Predicting gaze-based target selection in augmented reality headsets based on eye and head endpoint distributions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23. Association for Computing Machinery, New York, NY, USA, 2023. 1, 3, 4, 9

[58] H. R. Wilson, F. Wilkinson, L.-M. Lin, and M. Castillo. Perception of head orientation. *Vision Research*, 40(5):459–472, 2000. 2

[59] J. Wolf, Q. Lohmeyer, C. Holz, and M. Meboldt. Gaze comes in handy: Predicting and preventing erroneous hand actions in ar-supported manual tasks. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 166–175, 2021. 3

[60] D. Yu, H.-N. Liang, F. Lu, V. Nanjappan, K. Papangelis, W. Wang, et al. Target selection in head-mounted display virtual reality environments. *J. Univers. Comput. Sci.*, 24(9):1217–1243, 2018. 2

[61] D. Yu, H.-N. Liang, X. Lu, K. Fan, and B. Ens. Modeling endpoint distribution of pointing selection tasks in virtual reality environments. *ACM Trans. Graph.*, 38(6), nov 2019. 1, 2, 4, 9

[62] T. Zhang, Z. Hu, A. Gupta, C.-H. Wu, H. Benko, and T. R. Jonker. Rids: Implicit detection of a selection gesture using hand motion dynamics during freehand pointing in virtual reality. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22. Association for Computing Machinery, New York, NY, USA, 2022. 2, 3, 9

[63] B. Ziebart, A. Dey, and J. A. Bagnell. Probabilistic pointing target prediction via inverse optimal control. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*, IUI '12, p. 1–10. Association for Computing Machinery, New York, NY, USA, 2012. 8

A Significant results of RM-ANOVAs for eye, head, and hand angular movements

Table 3: Significant results of RM-ANOVAs for eye, head, and hand angular movements (α = .05).

Factor	Modality	df_{effect}	df_{error}	F	p	η_p^2
θ	Eye	1.689	25	232.825	**.000**	.939
W	Eye	1	15	161.498	**.000**	.915
ϕ	Eye	2.416	36.241	4.503	**.013**	.231
$D \times \theta \times W$	Eye	4.138	62.070	4.674	**.002**	.238
θ	Head	1.186	17.791	29675.680	**.000**	.999
W	Head	1	15	9.836	**.007**	.396
ϕ	Head	1.504	22.556	18.370	**.000**	.550
$\phi \times W$	Head	5	75	30.612	**.000**	.671
$\theta \times \phi$	Head	2.447	36.708	16.077	**.000**	.517
D	Hand	2	30	5.618	**.008**	.272
θ	Hand	2.264	33.958	29675.680	**.000**	.999
W	Hand	1	15	9.836	**.007**	.396
$D \times \phi$	Hand	5.087	76.300	70.013	**.000**	.824