# Modeling Temporal Target Selection: A Perspective from Its Spatial Correspondence

**Difeng Yu**
The University of Melbourne
Melbourne, VIC, Australia
difengy@student.unimelb.edu.au

**Brandon Victor Syiem**
The University of Melbourne
Melbourne, VIC, Australia
bsyiem@student.unimelb.edu.au

**Andrew Irlitti**
The University of Melbourne
Melbourne, VIC, Australia
andrew.irlitti@unimelb.edu.au

**Tilman Dingler**
The University of Melbourne
Melbourne, VIC, Australia
tilman.dingler@unimelb.edu.au

**Eduardo Velloso**
The University of Melbourne
Melbourne, VIC, Australia
eduardo.velloso@unimelb.edu.au

**Jorge Goncalves**
The University of Melbourne
Melbourne, VIC, Australia
jorge.goncalves@unimelb.edu.au

Figure 1: Temporal target selection examples. A player is about to (1) shoot an enemy spaceship in Demon Attack, (2) fire an arrow towards a balloon in a VR Archery game in The Lab, (3) control the jump of a character onto a waterwheel in Moss, (4) attack a minion with a finishing shot to get the most gold while minions are hitting each other in Dota 2, and (5) dodge an enemy's attack when it is close so that the character can immediately fight back in Elden Ring. The player needs to wait and hit the input trigger within a limited time window to complete the task successfully. The player estimates how long the bullet/arrow/jump/attack/dodge takes based on their previous experiences with the game.

## ABSTRACT

Temporal target selection requires users to wait and trigger the selection input within a bounded time window, with a selection cursor that is expected to be delayed. This task conceptualizes, for example, a variety of game scenarios such as determining the timing of shooting a projectile towards a moving object. In this work, we explore models that predict "when" users typically perform a selection (i.e., user selection distribution) and their selection error rates in such tasks. We hypothesize that users react to temporal factors including "distance", "width", and "delay" as how they treat the corresponding variables in spatial target selection. The derived models are evaluated in a controlled experiment and an MTurk-based online study. Our research contributes new knowledge on user behavior in temporal target selection tasks and its potential connection with its spatial correspondence. Our models and conclusions can benefit both users and designers of relevant interactive applications.

## CCS CONCEPTS

• **Human-centered computing → User models**; **Empirical studies in HCI**.

## KEYWORDS

Timing, object selection, moving target, modeling, game.

## 1 INTRODUCTION

Explainable user models can enhance our understanding of human-computer interaction behaviors and inform the design of relevant

applications. This research concerns user behavioral modeling for a task scenario called temporal target selection. In a temporal target selection task, a user must wait for a time $D_t$ (temporal distance) for a target to become selectable and trigger the input within a limited time window $W_t$ (temporal width) to select the target successfully. The selection event happens $R_t$ in time after the user input, which is expected by the user (expected delay).

This task is a conceptualization of many application scenarios. For example, imagine a player in a first-person archery game aiming an arrow and firing it when a target is about to reach the crosshair. The player must anticipate when the target will reach the crosshair based on its movement ($D_t$) and trigger the selection within a limited time period ($W_t$) so that the arrow can hit the target. The arrow needs to travel for a certain time period to reach the target, so the player must estimate the cursor travel time ($R_t$) based on previous experiences and take this "delay" into account to trigger the selection earlier. The temporal factors ($D_t$, $W_t$, and $R_t$) can even be encoded in more abstract forms/animations (more in Figure 1). Note that the task only concerns users' temporal precision (to hit the target at the correct timing), while the spatial movement of the cursor is negligible. Despite being prevalent in games, models that can explain and predict user behaviors in such tasks are still under-explored.

To address this challenge, we propose temporal target selection models that predict user selection distribution (encapsulates "when" users typically perform a selection) and selection error rate. Our initial model is based on the hypothesis that the temporal factors are treated as individual cues for users to decide "*when*" to execute the input, as the corresponding spatial factors are used to decide "*where*" to execute the input. We conduct our first study in a controlled VR experiment where we evaluate different model variants and compare the impact of the corresponding factors in temporal and spatial target selection. We demonstrate that the proposed models provide accurate prediction results and are robust under cross-validation tests. We further conduct a second, MTurk-based online study to explore the generalizability of the models and conclusions. We show that our models can still provide helpful estimations in scenarios with more complex visual encoding, larger parameter ranges, and less-controlled environments. Additionally, we discover how temporal factors typically influence user behavior. For example, we found whether the expected delay will push the selection distribution forward or backward in time depending on the value of temporal distance.

Our primary contributions include:

- Models for predicting user selection distribution and error rate in temporal target selection.
- Findings and implications based on two user studies regarding user behavior in temporal target selection.
- Open-source datasets on three different temporal target selection applications collected from our studies.

Our models and conclusions bring new knowledge on human behavior in temporal target selection tasks and can benefit both designers and users of relevant applications. For example, without extensive user testing, game designers can be more confident in determining the appropriate difficulty levels with the estimated user selection errors. Players can also better approach challenging
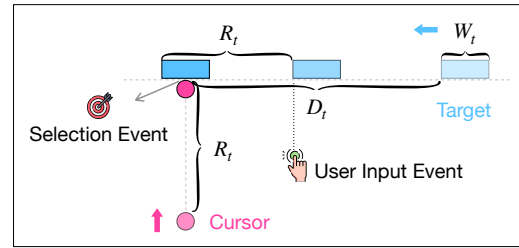


**Figure 2: Spatial demonstration of a temporal target selection task. The blue rectangle is a target, and the pink circle represents a cursor. The target takes $D_t$ (temporal distance) to become selectable, and $W_t$ (temporal width) to pass the selectable region. The cursor takes $R_t$ (expected delay) to trigger a selection event after user input. The selection is only successful if the cursor "hits" the target in the selectable region (i.e., input is triggered within $[D_t - R_t, D_t - R_t + W_t]$).**

game scenarios by understanding how temporal factors typically influence selection.

## 2 PROBLEM FORMULATION

To formalize the problem, we consider an example scenario illustrated in Figure 2. A *target* (blue rectangle) appears and travels on a straight line. To select the target, the user must trigger a *user input event* which shoots forward a *selection cursor* (pink circle). The user's goal is to time the user input event so that the selection cursor "collides" with the target, which triggers a successful *selection event*. We define the *selectable region* to be the interval of time within which a successful selection event can be detected. In addition, we define the *temporal distance* $D_t$ to be the time it takes for the target to reach the selectable region and the *temporal width* $W_t$ to be the time the target remains selectable. The *expected delay* $R_t$ is the amount of time it takes for the cursor to reach the target's path, and the user should have an estimation of $R_t$ based on existing experience. To successfully select the target, users must trigger the user input event within the time interval of $[D_t - R_t, D_t - R_t + W_t]$.

Note for each user input event occurring at $t_{\text{input}}$, the corresponding selection event (an individual data point of a selection distribution) is calculated as $t_{\text{input}} + R_t - D_t$. The selection distribution is thus relative to the selectable region (by treating the time that the target first becomes selectable as the origin) rather than the usual time scale used by the input event that treats the onset timing of the target as the origin.

In this problem definition, we only consider the case where the user must decide *when* to trigger the selection, as opposed to deciding the spatial position of the selection. In other words, unlike in a spatial target selection task, the user cannot aim the cursor with the input device (e.g., a controller) but can only control the cursor's onset timing through, for example, a button press. The upper bound of the input time window ($D_t - R_t + W_t$) must be larger than human reaction time (~0.25s), otherwise typical users would not be able to select the temporal target successfully without prediction.

## 3 RELATED WORK

Our temporal target selection task concerns how temporal factors like "distance", "width", and "delay" may influence "when" users are likely to perform a selection (i.e., user selection distribution). This section reviews relevant but different temporal tasks explored in the literature, considering *when* users typically launch a selection input. We then summarize spatial target selection models that focus on predicting *where* users typically select a given spatial target with a certain "distance" and "width", which inspired our temporal models. After that, we scrutinize the factor of "delay" in our temporal target selection task.

### 3.1 Temporal target selection

In our temporal target selection task, user selection distribution describes *when* users typically select a given temporal target; it summarizes the spread of selection events recorded by a computer in the time domain. Existing literature has studied several relevant but different temporal tasks, including reaction, coincidence-anticipation, temporal pointing, and interception.

*Reaction tasks* require users to respond to a stimulus as rapidly as possible [39]. In these tasks, the onset of the stimulus is unpredictable. The user reacts to the stimulus once it appears ($D_t = 0$), and the time window for selection is minimal because it must be executed as fast as possible ($W_t = 0$).

*Coincidence anticipation tasks* require users to respond when a moving stimulus coincides with a fixed object [2, 30]. These tasks expect users to respond at the exact instant of the coincidence and do not measure the success or failure of a response based on whether the response is provided within a pre-determined time window ($W_t = 0$).

*Temporal pointing tasks* require the selection of a target within a limited time window with the following prerequisites: (1) the target appears repetitively in time so that a user forms expectations on how often the selection should be repeated, and (2) the visual stimulus is smaller than human reaction time ($\sim 0.25s$) so that the user does not simply react to it [23–26]. While a temporal pointing task and our task both require some repetitions of the selection action for users to form the mental model ($R_t$ in our case), the former expects users to synchronize with some repeating targets, while our task does not assume this. Furthermore, the expected delay $R_t$ was not considered in temporal pointing tasks.

*Interception tasks* require a user to capture a moving object with an intercepting effector (e.g., a hand) [17, 40]. In such tasks, the effector and the target must coincide at the exact location (spatial) and time (temporal) for an interception to occur. However, our task focuses on scenarios where the spatial movements of the effector are negligible. The additional $R_t$ factor was also not acknowledged in previous interception tasks.

In sum, the temporal target selection task we aim to model has several unique properties as compared to the temporal tasks studied by previous research. In our task, (1) a temporal target is needed ($W_t > 0$); (2) the actual selection event is delayed after user input ($R_t > 0$), and the user has an estimation of the delay; (3) the minimal time for the user to make a successful selection based on the visual stimulus ($D_t + W_t - R_t$) needs to be larger than human reaction time ($\sim 0.25s$), as typical users will not be able to make a

successful selection otherwise; and (4) the spatial task requirement (e.g., moving an effector spatially) is negligible.

### 3.2 Spatial target selection

Spatial target selection requires users to hit a target (width $W_s$ and distance $D_s$) in the spatial domain (e.g., a button on a screen) with a movable cursor. Numerous models have been proposed to understand user behaviors in such tasks. For example, Fitts's law and its descendants [13, 21, 28, 36] are widely applied in HCI research to predict cursor movement time for selecting a given target.

More relevant to our research are models that predict *where* users typically select a given spatial target. This is typically termed user selection distribution or endpoint distribution, which represents the spread of cursor positions (or orientations with 3D Raycasting) when the selection is triggered. Early endeavors on selection distribution and speed-accuracy tradeoff found that the selection endpoints can be approximated by Gaussian (1D) or bi-variate Gaussian (2D) distributions [33, 46].

Later works have investigated how different spatial factors like $W_s$ and $D_s$ may affect user selection distribution of spatial targets with various input devices/modalities. Grossman and Balakrishnan [14] and Grossman et al. [15] found that the spread of hit (distribution standard deviation) increases with $D_s$ by a constant factor for puck-based input. Bi and colleagues [3–5] demonstrated a strong linear relationship between the distribution variance and $W_s^2$ for finger touch input on touchscreens. Yu et al. [50] explored pointing selection distribution in VR and found both $W_s$ and $D_s$ could play a role in the mean and variance of the distribution depending on the input modality (head/hand pointing). Yamanaka and Usuba [48] later found that there was no apparent benefit of integrating $D_s$ for predicting distribution variance in an on-screen-start pointing task with touchscreens. Huang and colleagues [18, 19, 52] found that only $W_s$ and target speed would affect the selection distribution in their mouse input-based moving target selection tasks. These findings on how spatial factors (i.e., $W_s$ and $D_s$) may influence user selection distribution inspired our initial hypothesis for temporal target selection.

### 3.3 Latency and Expected Delay

Various channels can defer the actual selection event (as determined by the computing system) after the user forms a selection intention. Users' motor delays caused by, for example, neuromuscular transmission lags, lie between the selection intention and the user click action [31, 45]. In addition, the user click action, which is the rapid movement of the user's finger pressing the trigger, also defers the actual input event. Previous research related to spatial pointing, such as the application of Fitts's law in HCI [36], usually assume that the effects of such delays are negligible as compared to the spatial pointing task itself and do not take these factors into account.

Another source of the delay comes from end-to-end latency, which is normally referred to as the unintended total time elapsed between a user's initiation act (e.g., a button press) and the system's responses [26]. Multiple factors can contribute to end-to-end latency in an interactive system, including device delays, network delays, and processing delays [7, 8, 26, 32, 47]. When end-to-end

latency reaches a problematic level (i.e., significant latency that is perceptible by users), it can shift user's input distribution and affect user performance [1, 20].

In contrast to previous research, our work focuses on *expected delay* ($R_t$), which represents the user-anticipated delay between the selection action and the actual selection event. Unlike end-to-end latency, which is usually unexpected or not intended, expected delays are anticipated by users according to their previous experiences. For example, players have a rough estimate of the bullet travel speed because they have played with the weapon a few times.

## 4 HYPOTHESES AND MODELS

### 4.1 Hypotheses

To investigate how users react to temporal factors including "distance" ($D_t$), "width" ($W_t$), and "delay" ($R_t$) in temporal target selection, we first took inspiration from relevant findings in spatial target selection. Specifically, we assume that users' responses (i.e., user selection distribution) are similarly affected by corresponding variables in spatial and temporal target selection tasks. In this case, $D_t$ and $D_s$ both represent the "distance" before a successful hit. $W_t$ and $W_s$ stand for the "width" for a successful hit. We interpret $R_t$ and $R_s$, as will be illustrated in Section 4.2, as the estimated "distance" between a nominal target and an actual (but invisible) target. We treat all temporal factors ($D_t$, $W_t$, and $R_t$) as individual cues for users to decide "*when*" to execute the input, as spatial factors ($D_s$, $W_s$, and $R_s$) are used to decide "*where*" to execute the input.

The assumption is motivated by Walsh's *A Theory Of Magnitude* (ATOM). ATOM hypothesizes that space and time information are linked by a common metric for action [43]. A spatial or temporal event input (e.g., distance and duration) is handled by a shared analogue magnitude system to produce motor output (an estimation of how fast and how long) [27, 43]. Based on ATOM, a potential inference is that the corresponding spatial and temporal information, as processed by a common mechanism, can similarly influence perception and response. Evidence of this includes previous studies that have shown, for example, that discriminating temporal and spatial magnitude both follow Weber–Fechner's law, which states that the just noticeable difference in a stimulus is a constant ratio of the original stimulus [10, 37, 38].

Based on these hypotheses, we built an initial model that predicts user selection distribution $\mathcal{N}(\mu, \sigma^2)$ and error rate $E$ in temporal target selection based on established literature in spatial target selection. The model takes the temporal width $W_t$ (time that a target remains selectable), temporal distance $D_t$ (time for the target to reach the selectable region after its appearance), and expected delay $R_t$ (user-anticipated time for the cursor to reach the selectable region) as input and produces estimated mean $\mu$ and standard deviation $\sigma$ of the selection distribution, which is assumed to be Gaussian, and with error rate $E$. We detail our modeling process in the following.

### 4.2 Variable mapping

To establish an initial temporal model based on our hypotheses, we first examine how spatial and temporal variables correspond to each other by mapping a 1D spatial target selection task to our temporal target selection task (see Figure 3). In the temporal task, a user
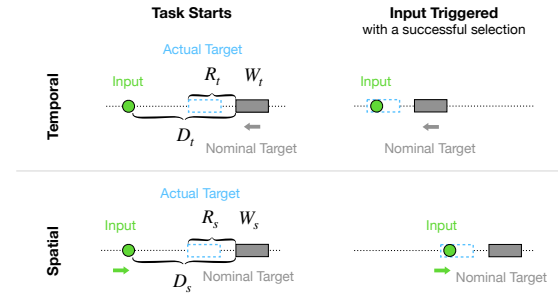


**Figure 3: An analogy of a temporal target selection task (top) and a 1D spatial target selection task (bottom). In the temporal task, the user waits and triggers the input $R_t$ before the arrival of a nominal target to make a successful selection. In the spatial task, the user moves an input cursor towards a nominal target and triggers the input $R_s$ in front of the nominal target to make a successful selection. In both cases, the user in fact needs to trigger the input to hit an actual (but invisible) target that is $R_t$ or $R_s$ ahead of the nominal target.**

triggers the onset of a selection cursor, which is delayed by $R_t$, to hit a moving target with temporal width $W_t$ and temporal distance $D_t$. We name the visible moving target as the *nominal target*. To select the nominal target successfully, the user must trigger the input $R_t$ earlier in time, so that the delayed selection cursor can "collide" with the target. This is equivalent to triggering the input within a bounded time window $W_t$ that is always $R_t$ ahead of the selectable region. We name this input time window as the *actual (but invisible) target* (see Figure 3 top).

Correspondingly, in the 1D spatial selection task, the user moves an input cursor to hit an actual (but invisible) target, given a nominal target with spatial width $W_s$ and spatial distance $D_s$. The actual target has width $W_s$ and stays $R_s$ ahead of the nominal target (see Figure 3 bottom). With such a formulation, the temporal distance $D_t$ and temporal width $W_t$ are directly mapped onto the spatial distance $D_s$ and spatial width $W_s$. The expected delay $R_t$ is reconstructed as the "distance in time" between the actual and the nominal target and is mapped onto the "distance in space" $R_s$. In both cases, the user must estimate this "distance" to hit the actual (but invisible) target successfully. That is, the temporal task is successful only if the input is triggered within $[D_t - R_t, D_t - R_t + W_t]$, which has the same successful criterion as the spatial selection task— the cursor movement amplitude (towards the target) is between $[D_s - R_s, D_s - R_s + W_s]$. The user should have a prior estimation of $R_t$ and $R_s$ based on previous experiences.

Admittedly, we define the abstract spatial selection task mainly to demonstrate how temporal and spatial factors correspond to each other, so that we can infer the effect of temporal factors on selection distributions based on empirical results collected from spatial target selection research. However, such a spatial selection task also has practical applications. Existing research has suggested that the effective input region (the actual but invisible target) can be shifted from the perceived visual target because of interface designs [41, 42] or through the use of predictive systems [12, 49, 50].

## 4.3 Selection distribution mapping

Based on the assumption that the user responds to the corresponding variables in spatial and temporal target selection tasks in a similar manner, we explore how user selection distribution may change according to $D_s$, $W_s$, and $R_s$ in the 1D spatial target selection task, which provides us insights for deriving a model in the temporal domain.

Previous literature suggests that the selection distribution of spatial target selection tasks can be approximated by a Gaussian distribution $\mathcal{N}(\mu_s, \sigma_s^2)$ [3, 14, 29, 34, 44, 50]. The center of the distribution $\mu_s$ is affected by both target width and movement amplitude, presumably in a linear relationship [50]. Intuitively, a larger and distant object can shift the center of the selection distribution closer to the edge of the object. The magnitude of the effect can depend on the input modality and the variable range tested in the experiments. For example, existing research has shown that it is useful to add movement amplitude to predict the center of the distribution with head-based input (a sloppier pointer), but not hand-based input [48, 50]. To account for all potential effects, it is reasonable to assume a linear relationship of $\mu_s = p_0 + p_1(D_s - R_s) + p_2 W_s$, where the movement amplitude of the cursor to select the actual target is represented by $(D_s - R_s)$.

Previous findings have also suggested that the standard deviation of the distribution $\sigma_s$ can relate linearly to both target width and movement amplitude [4, 5, 14, 50]. Larger target width enables a wider possible area for selection (e.g., $\sigma^2 \propto W^2$ in Bi et al. [4]), and larger movement amplitude causes larger variances in ballistic selection movements (e.g., $\sigma \propto D$ in Grossman et al. [14]). Thus, according to the literature, another linear relationship that can be established based on the actual movement amplitude $(D_s - R_s)$ and the target width $(W_s)$ is $\sigma_s = q_0 + q_1(D_s - R_s) + q_2 W_s$.

Since there is no existing research on how the distance between the nominal and the actual target $(R_s)$ affects the selection distribution, so we assume $R_s$ has a linear effect on both $\mu_s$ and $\sigma_s$. Intuitively, we hypothesize that a larger $R_s$ can further shift the whole distribution $(\mu_s)$ and increase the uncertainty in the distribution $(\sigma_s)$. Therefore, we conclude that the selection distribution of the 1D spatial target selection task can be approximated by $\mathcal{N}(\mu_s, \sigma_s^2)$, where $\mu_s$ and $\sigma_s$ can be calculated via Equation 1 and 2. The intercept of both equations ($p_0$ and $q_0$) aggregates imprecision and noise from the input device and the internal human motor control system.

$$\mu_s = p_0 + p_1(D_s - R_s) + p_2 W_s + p_3 R_s \tag{1}$$

$$\sigma_s = q_0 + q_1(D_s - R_s) + q_2 W_s + q_3 R_s \tag{2}$$

In a temporal target selection task, we assume all temporal factors ($D_t$, $W_t$, and $R_t$) are treated as individual cues for users to decide "when" to execute the input, similarly to how spatial factors ($D_s$, $W_s$, and $R_s$) are used to decided "where" to execute the input. Therefore, to formulate our hypothesized model, we derived two linear relations (Equation 3 and 4) for the temporal selection distribution $\mathcal{N}(\mu, \sigma^2)$ based on the spatial correspondence. Coefficients including $a_0$, $a_1$, $a_2$, $a_3$, $b_0$, $b_1$, $b_2$, and $b_3$ must then be empirically determined.

$$\mu = a_0 + a_1(D_t - R_t) + a_2 W_t + a_3 R_t \tag{3}$$

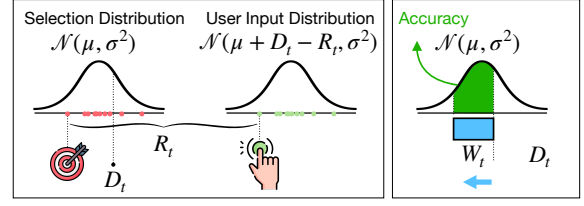$$\sigma = b_0 + b_1(D_t - R_t) + b_2 W_t + b_3 R_t \tag{4}$$



**Figure 4: The relationship between selection distribution and user input distribution (left). Selection accuracy can be calculated by integrating a Gaussian distribution (right).**

## 4.4 Selection error rate prediction

Similarly to previous research (e.g., [5]), by plugging $\mu$ and $\sigma$ into the probability density function of Gaussian distribution (Equation 5) and taking an integral on the selectable area (Equation 6, Figure 4 right), we can compute the selection error rate.

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(t-\mu)^2}{2\sigma^2}} \tag{5}$$

$$E = 1 - \int_0^{W_t} f(t)dt = 1 - \frac{1}{2}\left[ erf\left(\frac{W_t - \mu}{\sigma\sqrt{2}}\right) + erf\left(\frac{\mu}{\sigma\sqrt{2}}\right)\right] \tag{6}$$

where $erf(z)$ is commonly used in Gaussian distribution integration.

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt \tag{7}$$

## 5 STUDY 1: CONTROLLED EVALUATION

In this study, our goal was to evaluate the hypothesized model and analyze how temporal factors ($D_t$, $W_t$, and $R_t$) affect user selection distribution and error rate as compared to their spatial correspondence. Therefore, we conducted a controlled temporal target selection experiment in VR. We chose VR because it allowed us to immerse participants in the experimental environment, and we could potentially compare our results with a recent study that investigated head- and controller-based spatial target selection distribution in VR [50].

## 5.1 Method

*5.1.1 Participants.* We recruited 16 participants (8 women, 8 men) with diverse educational backgrounds from a local university. Their average age was 23.5 (SD = 2.4). Their self-rated familiarity score with VR systems was 4.6 on average (SD = 1.0) on a 7-point Likert scale. All of them had normal or corrected-to-normal vision.

*5.1.2 Apparatus and materials.* The application was developed with Unity3D and ran on an Intel Core i7 processor laptop with a dedicated NVIDIA RTX 2070 graphics card. Participants performed the experiment with an Oculus Quest 2 headset (featuring 1832 × 1920 pixel resolution per eye) and a right-hand Oculus Touch controller.
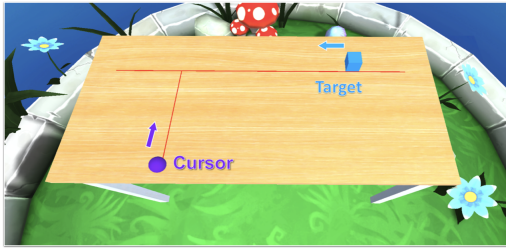
**Figure 5: A demonstration of the experimental environment in Study 1.**

*5.1.3 Design.* The experiment employed a $3 \times 2 \times 4$ within-subjects design with three independent variables: Temporal Distance, Temporal Width, and Expected Delay. The levels were chosen as follows, leading to 24 experimental conditions:

- Temporal Distance ($D_t$): 0.4s, 0.5s, and 0.6s
- Temporal Width ($W_t$): 0.1s and 0.2s
- Expected Delay ($R_t$): 0.05s, 0.1s, 0.15s, and 0.2s

These parameters were determined from a pilot study with the aim of not making the task too easy (information-saturated) or too difficult (faster than reaction time $\sim 0.25s$), while still covering a variety of difficulty levels. In the most difficult condition, the response time for a successful selection ($D_t + W_t - R_t$) was 0.3s. In the experiment, the order of $R_t$ was counterbalanced across participants. Within each $R_t$, each $W_t \times D_t$ experimental condition was generated 20 times, and all these generated trials appeared in a randomized order. This simulated the condition when participants had a constant estimate of $R_t$ in each block and responded to various targets with different properties (as per $W_t$ and $D_t$).

*5.1.4 Task.* Participants were asked to hit a blue target moving leftward along a horizontal line with a sphere (selection cursor) moving forward. We varied the sphere's color to distinguish between different $R_t$, with colors chosen from a colorblind-friendly palette. Figure 5 illustrates the task environment.

Targets always appeared at the same starting position in the virtual space $\vec{p_t} = (0.3, -0.5, 1.75)$. The target speed was adjusted according to $D_t$ (the time for the target to reach the hittable area), and the target width was modified based on $W_t$ (the time that the target remained hittable). Similarly, the cursor started at the same position $\vec{p_b} = (-0.7, -0.5, 1.05)$, and the its speed changed according to $R_t$. A successful hit was determined when the cursor front reached the horizontal line and the cursor coincided with the object. Programmatically, the hit was pre-calculated based on when participants pressed the selection trigger to avoid latency caused by collision detection. After the cursor reached the horizontal line, the animation paused for 0.1s, and quick sound feedback was given to reflect the correctness of the hit. The subsequent trial was then randomly generated within the next (0.5s, 1.0s) interval.

*5.1.5 Procedure.* The whole study lasted approximately 25 minutes for each participant. The study was designed to be short to avoid participant disengagement [51]. Before coming to the lab, participants filled in a questionnaire to collect their demographic information. Upon arrival, they were introduced to the experiment

and signed a consent form. Next, they were invited to wear a VR headset and were given around 1 minute to familiarize themselves with the device where they could look around the experimental environment in VR. They then adjusted their sitting position to make sure they could see both the cursor and the target. We instructed participants to perform the task as accurately as possible. Each session (4 sessions as per $R_t$) consisted of a warm-up period and a formal experiment period. During the warm-up, we asked participants to become familiar with the time for the cursor to reach the selectable region ($R_t$). After at least 5 practice trials, they could proceed to the next stage by pressing a controller button if they could confidently make a selection. The first 10 starting trials of the formal experiment period were sampled from the $D_t \times W_t$ conditions uniformly and at random and discarded as practice trials. Participants took a short break after completing each block, where they could decide how long to rest and whether to take off the headset. Participants received a $10 gift voucher after completing the experiment.

## 5.2 Results

In total, we collected 7680 data points (16 participants $\times 3\ D_t \times 2\ W_t \times 4\ R_t \times 20$ repetitions) from the experiment. We first removed 40 trials (0.52%) of outliers in which the user input time was above or below three standard deviations from the mean ($mean \pm 3std.$) in each condition. These outliers could be induced by confusion/mind-wandering of the participants. In the next sections, we present the results of normality testing, statistically effects of independent variables, model fitting, and information criterion measures.

*5.2.1 Normality of the response distribution.* We verified whether the selection distribution obtained from the experiment followed a Gaussian distribution as expected. We tested the normality of each task condition per user (16 participants $\times 24$ conditions) using Shapiro-Wilk tests with significance level $\alpha = .05$. We found that 84.4% of the 384 distributions obtained were normally distributed as in these cases the null hypothesis that the sample came from a normally distributed population could not be rejected. For the individual who met the normality distribution at the lowest rate, 70.8% of the conditions were found to be normal. We thus included all participants' data for analysis.

*5.2.2 Effects of independent variables.* We first evaluated the normality assumption of parametric analysis with Q-Q plots, which suggested strong fits of normality for all conditions (Q-Q plots are attached in the supplementary material [1]). We thus employed repeated-measures ANOVA (RM-ANOVA, $\alpha = 0.05$) and generalized eta-squared ($\eta_G^2$, an effect size measure) to explore the effects of temporal factors ($D_t$, $W_t$, and $R_t$) on $\mu$ and $\sigma$ of the selection distributions. We applied Greenhouse-Geisser correction when the sphericity assumption was violated, as indicated by Mauchly's test for sphericity.

Table 1 summarizes the results from the statistical tests regarding $\mu$ and $\sigma$. Overall, the RM-ANOVA indicated that $D_t$, $W_t$, and $R_t$ all had statistically significant main effects on $\mu$. It also revealed interaction effects between $D_t \times R_t$ and $W_t \times R_t$. Another RM-ANOVA showed that $D_t$, $W_t$, and $D_t \times R_t$ had significant main effects on

---

[1]https://github.com/Davin-Yu/TemporalSelection-CHI2023

**Table 1: Statistical effects of factors on $\mu$ (left) and $\sigma$ (right).**

| Factor | $df_{\text{effect}}$ | $df_{\text{error}}$ | $F$ | $p$ | $\eta_G^2$ | Sig? | Factor | $df_{\text{effect}}$ | $df_{\text{error}}$ | $F$ | $p$ | $\eta_G^2$ | Sig? |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $D_t$ | 1.299 | 19.485 | 371.835 | .000 | **.802** | yes | $D_t$ | 2 | 30 | 293.637 | .000 | **.527** | yes |
| $W_t$ | 1 | 15 | 48.422 | .000 | .116 | yes | $W_t$ | 1 | 15 | 11.979 | .003 | .027 | yes |
| $R_t$ | 3 | 45 | 15.993 | .000 | **.162** | yes | $R_t$ | 1.786 | 26.790 | 2.378 | .082 | .021 | no |
| $D_t \times W_t$ | 2 | 30 | 2.804 | .076 | .009 | no | $D_t \times W_t$ | 1.266 | 18.993 | 1.394 | .264 | .008 | no |
| $D_t \times R_t$ | 2.888 | 43.316 | 81.547 | .000 | **.431** | yes | $D_t \times R_t$ | 6 | 90 | 4.564 | .000 | .046 | yes |
| $W_t \times R_t$ | 3 | 45 | 4.282 | .009 | .014 | yes | $W_t \times R_t$ | 3 | 45 | 0.206 | .892 | .002 | no |
| $D_t \times W_t \times R_t$ | 6 | 90 | 0.719 | .635 | .005 | no | $D_t \times W_t \times R_t$ | 6 | 90 | 1.539 | .175 | .015 | no |

$\sigma$. Pairwise comparison results can be found in the supplementary material.

The 3D scatter plots in Figure 6 demonstrate the interaction relationship among $D_t$ and $R_t$ with regard to $\mu$ and $\sigma$. Generally, when $D_t$ increased, $\mu$ decreased (selections were performed earlier in the selection region), while $\sigma$ increased (wider spread of selections). The speed of the increases/decreases depended on $R_t$—the impact of $D_t$ on both $\mu$ and $\sigma$ become larger as $R_t$ increased.

It is shown from the effect size measures that $D_t$ and $R_t$ were predominant cues for determining the center of the distribution $\mu$, and $D_t$ had a large effect on determining the standard deviation of the distribution $\sigma$ (all $\eta_G^2 > 0.14$ as a rule of thumb for a large effect size [9]).

*5.2.3 Model Candidates.* In addition to the hypothesized model introduced in Section 4 (which we call the *hyp model*), we also included two model variants derived from our user data for comparison. As discussed, the effect magnitude of each temporal factor on the selection distribution ($\mu$ and $\sigma$) may vary depending on the input modality and the tested variable range in the experiment. Therefore, a simplified model that only incorporates main/simple effects with a large effect size may provide enough explanation power for a given task scenario. Thus, we deemed that a temporal factor should be incorporated into our first comparison model—the *simple model*—if it has a large effect size on $\mu$ or $\sigma$ based on the collected data ($\eta_G^2 > 0.14$). In this case, $D_t$ and $R_t$ were used for predicting $\mu$, and $D_t$ alone was used for estimating $\sigma$.

While we initially assumed an additive relationship of the temporal factors for determining the selection distribution, we found a strong interaction effect of $D_t \times R_t$ on $\mu$, with a large effect size being identified from the data ($\eta_G^2 = 0.431$). Therefore, we also included another model that considered $D_t \times R_t$ interaction for $\mu$ prediction on top of the simple model, which we call the *interact model*.

*5.2.4 Model fitting.* We fit the models with the `fitlm` function available in `MATLAB`. The fitting was performed at the population-level by averaging all participant data for each condition (24 conditions in total). The regression function produced the following results.

$$simple\ model \begin{cases} \mu = 0.300 - 0.489D_t + 0.147R_t \\ \sigma = -0.025 + 0.172D_t \end{cases} \tag{8}$$

$$interact\ model \begin{cases} \mu = 0.063 - 0.014D_t + 2.046R_t - 3.798D_t \cdot R_t \\ \sigma = -0.025 + 0.172D_t \end{cases}$$
$$\tag{9}$$

$$hyp\ model \begin{cases} \mu = 0.278 - 0.489(D_t - R_t) + 0.145W_t - 0.342R_t \\ \sigma = -0.035 + 0.172(D_t - R_t) + 0.047W_t + 0.198R_t \end{cases}$$
$$\tag{10}$$

*5.2.5 Model Performance.* We evaluated model performance with standard metrics including $R^2$ (coefficient of determination), *MAE* (mean absolute error), and two additional information criterion measures (*AIC* and *BIC*). Information criteria such as AIC (Akaike's information criterion) and BIC (Bayesian information criterion) evaluate the model fit by applying a penalty to the model complexity (i.e., the number of parameters). Generally, a lower information criterion value indicates a better model.

Table 2 summarizes the model performance of the three model variants. Overall, all models achieved good fit for $\mu$ and $\sigma$ estimation (all $R^2 > 0.80$). With the *simple model* as the baseline, the results suggested after incorporating the interaction of $D_t \times R_t$, as in the *interact model*, the prediction performance of $\mu$ increased significantly. Including $W_t$ for $\mu$ estimation, as in the *hyp model*, only provided limited benefits. For $\sigma$ estimation, adding $W_t$ and $R_t$ only offered marginal improvement of the fitting performance. Both information criteria favored the *interact model* for $\mu$ estimation and the *hyp model* for $\sigma$ estimation. These models struck a better balance between goodness-of-fit and parsimony (saving free parameters).

Error rate estimates were calculated based on the predicted distribution $\mathcal{N}(\mu_s, \sigma_s^2)$ with the three model variants. The *interact model* led to the most accurate prediction, while the *simple model* and the *hyp model* produced comparable performance.

*5.2.6 Cross-Validation.* We performed two cross-validation tests to verify the generalizability of the three models. In the first analysis, we obtained the model coefficients from 18 randomly chosen experimental conditions (levels) and tested the model fitting on the rest of 6 conditions for over 100 iterations. In the second analysis, we obtained the model coefficients from 12 randomly chosen participants and tested the model fitting on the remaining 4 participants for over 100 iterations.

Table 3 summarizes the performance results. Overall, our results demonstrated that all the models achieved accurate predictions in the cross-validation analysis, which are also similar to the original estimates. This indicates that the models could predict unseen
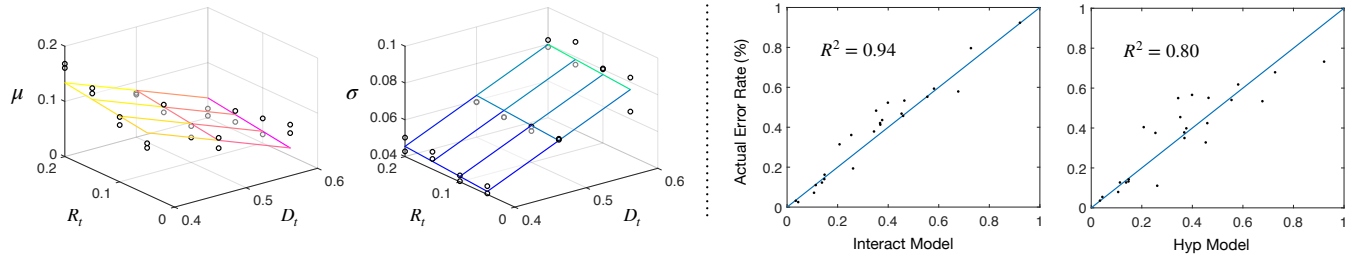
Difeng Yu, Brandon Victor Syiem, Andrew Irlitti, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves



**Figure 6: Left: The averaged predictions (surface) and the observations (dot) of $\mu$ and $\sigma$ using the hyp model. Right: Correlation plots between predicted versus observed error rates regarding the interact model and the hyp model.**

**Table 2: Goodness-of-fit and information criteria comparison among the simple model, the interact model, and the hyp model.**

| Model | Selection Mean ($\mu$) | | | | Selection Std. ($\sigma$) | | | | Error Rate | |
| | $R^2$ | MAE | AIC | BIC | $R^2$ | MAE | AIC | BIC | $R^2$ | MAE |
|---|---|---|---|---|---|---|---|---|---|---|
| Simple model | 0.81 | 0.017 | -113.64 | -110.11 | 0.90 | 0.003 | -186.43 | -184.08 | 0.77 | 8.20% |
| Interact model | 0.95 | 0.008 | -144.95 | -140.23 | 0.90 | 0.003 | -186.43 | -184.08 | 0.94 | 4.57% |
| Hyp model | 0.83 | 0.015 | -115.01 | -110.30 | 0.94 | 0.003 | -193.46 | -188.74 | 0.80 | 7.32% |

**Table 3: Cross-validation (LCO: leave-condition-out; LPO: leave-participant-out) performance results.**

| | Model | Selection Mean ($\mu$) | | Selection Std. ($\sigma$) | | Error Rate | |
| | | $R^2$ | MAE | $R^2$ | MAE | $R^2$ | MAE |
|---|---|---|---|---|---|---|---|
| LCO | Simple model | 0.78 (0.17) | 0.020 (0.005) | 0.90 (0.10) | 0.004 (0.001) | 0.72 (0.20) | 9.18% (3.39%) |
| | Interact model | 0.91 (0.10) | 0.011 (0.003) | 0.90 (0.10) | 0.004 (0.001) | 0.93 (0.06) | 5.33% (1.70%) |
| | Hyp model | 0.78 (0.17) | 0.019 (0.006) | 0.91 (0.07) | 0.004 (0.001) | 0.73 (0.20) | 8.64% (3.27%) |
| LPO | Simple model | 0.79 (0.06) | 0.018 (0.002) | 0.81 (0.06) | 0.006 (0.001) | 0.73 (0.05) | 9.58% (1.17%) |
| | Interact model | 0.92 (0.03) | 0.012 (0.002) | 0.81 (0.06) | 0.006 (0.001) | 0.89 (0.03) | 6.42% (1.18%) |
| | Hyp model | 0.82 (0.05) | 0.017 (0.002) | 0.82 (0.05) | 0.006 (0.001) | 0.76 (0.05) | 8.87% (1.08%) |
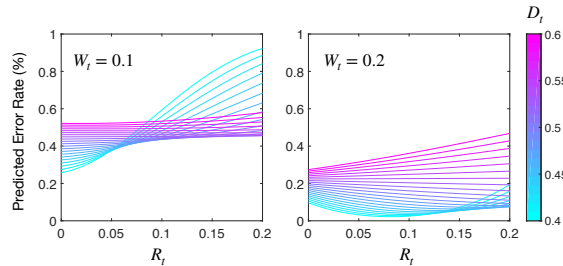


**Figure 7: Simulated effect of temporal factors on error rate.**

experimental conditions with high accuracy and could be safely generalized to unseen populations.

*5.2.7 Simulated effect of temporal factors on error rate.* We further explored how different levels of temporal factors typically influence selection error rates through simulations. We used the *interact model* for both $\mu$ and $\sigma$ estimation and calculated the corresponding predicted error rates, as it achieved the best performance. The empirical coefficients we used were $W_t = 0.1$ or $W_t = 0.2$, $D_t \in [0.4, 0.6]$ with step = 0.01, $R_t \in [0, 0.2]$ with step = 0.01.

The simulation results are presented in Figure 7. When $W_t = 0.1$ and $D_t$ were relatively small ($\sim [0.4, 0.5]$), the increase of $R_t$ would bring up the selection error rate significantly. In contrast, when $D_t$ was relatively large, the error rate increased steadily with $R_t$. When $W_t = 0.2$ and $D_t$ was $\sim [0.4, 0.45]$, the error rate went down when $R_t$ increased from 0 to 0.1 and went up again when $R_t$ increased from 0.1 to 0.2. The error rate decreased steadily with $R_t$ when $D_t$ was $\sim [0.45, 0.53]$ and increased steadily with $R_t$ when $D_t$ was $\sim [0.53, 0.6]$.

## 5.3 Discussion

Our models produced promising results in both selection distribution and error rate estimation. The performance of the *hyp model* was close to the *simple model*, which suggested that certain temporal factors did not play a crucial role in $\mu$ and $\sigma$ determination under our task setting. The *interact model* further boosted the performance, which indicated that it was helpful to consider the potential interaction between $D_t$ and $R_t$. Our leave-condition-out and leave-participant-out cross-validation showed that our models were robust to predict unseen experimental conditions and populations. In the following, we examine closer the space and time relationship and scrutinize the effect of the expected delay factor.

*5.3.1 Space and time relationship.* We developed the *hyp model* based on the hypotheses that the selection distribution is similarly affected by corresponding variables in spatial ($D_s$, $W_s$, and $R_s$) and temporal ($D_t$, $W_t$, and $R_t$) target selection tasks. The fitting results of the model allow us to examine closer this potential space and time relationship.

First, our results showed that user selection distributions in temporal target selection is also Gaussian, as in the spatial target selection. For determining $\mu$, the coefficient of temporal distance ($a_1$) is smaller than 0, which suggests that increasing target distance shifts the selection distribution forward in time. Similar evidence can be found with head-based pointing in spatial target selection, where a larger $D_s$ brought the selection distribution closer to the starting point [50]. Meanwhile, the coefficient of temporal width ($a_2$) is larger than 0, which indicates that a larger $W_t$ "pushes back" the center of the distribution. While we could not find a spatial correspondence of this finding, our result seems to suggest that a larger $W_t$ enabled a larger possible space for selection. Additionally, we noticed that when users are selecting a zero width object $W_t = 0$ with zero delay ($R_t = 0$) at zero distance away ($D_t - R_t = 0$), $\mu = a_0 = 0.278s$ which is close to typical human reaction time (~0.25s). The value probably encodes the effects caused by the input device and the internal human motor control system.

For determining $\sigma$, temporal distance and temporal width are positively correlated to the standard deviation of the distribution (as $b_1 > 0$ and $b_2 > 0$). This matches previous findings in spatial target selection, where the standard variance of the selection distribution can increase with $D_s$ and $W_s$ by a constant factor [14, 50].

*5.3.2 The effect of expected delay.* We first assumed a simple linear effect of expected delay ($R_t$) on both $\mu$ and $\sigma$, as there was no existing research on the corresponding spatial factor. However, we found a significant interaction of $D_t \times R_t$ on $\mu$, which complicates the impact of $R_t$ on the distribution.

Overall, because of the positive coefficients of $R_t$ for both $\mu$ (coefficient: 0.147) and $\sigma$ (coefficient: 0.198) in the *hyp model* (which only considers main effects), increasing $R_t$ generally pushes back the distribution in time and enlarges the variance of the distribution. However, the effect of $R_t$ on $\mu$ depends on $D_t$ according to the *interact model*—the coefficient of $R_t$ becomes $2.046 - 3.798D_t$. When $D_t$ is small (< 0.54), $R_t$ delays the distribution. This might be because when $D_t$ is small, the accumulated evidence (within $[D_t - R_t, D_t - R_t + W_t]$) for users to determine a proper input time is limited while they still need more time to average out uncorrelated noises [6]. This can induce a large number of errors in selecting small targets (i.e., $W_t$ is small). When $D_t$ is large ($\geq 0.54$), however, $R_t$ moves the distribution earlier in time. It could mean that when $D_t$ is large, users have more time to "wait" for the selection (as least $D_t - R_t$) and might trigger the input slightly earlier (as they do not want to wait any longer). This is a possible explanation for why selection errors increase when $D_t$ is large.

## 6 STUDY 2: GENERALIZATION

In this study, our goal was to assess the generalizability of our models and conclusions. Specifically, we were interested in verifying that we can generalize our results to (1) more complex visual encoding of temporal factors of $D_t$, $W_t$, and $R_t$, (2) larger parameter ranges of temporal factors, and (3) less controlled experiment environments other than immersive VR. To resolve these questions, we built two web-based game applications on temporal target selection (Space Shooter and Jump! Jump!) and used them in an online study on MTurk.

### 6.1 Space Shooter

*Space Shooter* is a simplified space shooter game in WebGL (Figure 8 left). The goal is to shoot down the enemy spaceship, which moves horizontally at a fixed speed, with the laser from the player's spaceship. Players cannot control the movement of their spaceship but shoot the laser at a specific time by pressing the spacebar. An explosion animation of the enemy spaceship is played if the laser hits it successfully.

We fixed the position of the player's spaceship at the center of the screen and its initial horizontal distance to the enemy spaceship as 6 units. We also set the movement speed of the laser to 80 unit/s. We then modified the following three in-game variables to construct different game difficulties.

- TARGET SPEED (horizontal movement speed of the enemy spaceship): 6 units/s, 8 units/s, and 12 units/s
- TARGET SIZE (size of the enemy spaceship): 1 unit, 1.5 units
- HEIGHT (vertical distance between the enemy's and the player's spaceship): 8 units, 12 units, and 16 units

These variables resulted in 6 $\{D_t, W_t\}$ combinations ({0.500s, 0.083s}, {0.500s, 0.125s}, {0.750s, 0.125s}, {0.750s, 0.188s}, {1.000s, 0.167s}, {1.000s, 0.250s}) and 3 $R_t$ values (0.100s, 0.150s, 0.200s), leading to 18 experimental conditions in total. The parameter ranges were $D_t \in [0.5s, 1.0s]$, $W_t \in [0.083s, 0.25s]$, and $R_t \in [0.1s, 0.2s]$. Notably, the values of $D_t$ were slightly higher than in the first controlled study ($D_t \in [0.4s, 0.6s]$). We thought it was likely that participants' response time would be longer for a less-controlled environment [22, 35], so that we lowered the difficulty a bit to avoid receiving too many error trials. The visual encoding of the temporal factors was more complicated with realistic textures and complex shapes.

### 6.2 Jump! Jump!

*Jump! Jump!* is a WebGL game that requires players to control the character to get the gold coin across the river by jumping onto a moving block (Figure 8 right). It mimics a game scenario in Moss where the players jump the character onto a waterwheel within a limited time window (Figure 1 middle). Players control the jump of the character by pressing the spacebar on their keyboard. An animation of the character getting the gold coin is played if the character passes the river successfully through the moving block, otherwise, the character will drop into the water.

We fixed the starting position of the moving block and the character with an initial distance of 8 units. We then adjusted the following three in-game variables to construct different game difficulties.

- BLOCK SPEED (movement speed of the block along the river): 6 units/s, 8 units/s
- BLOCK WIDTH (width of the moving block): 1.6 units, 2.4 units
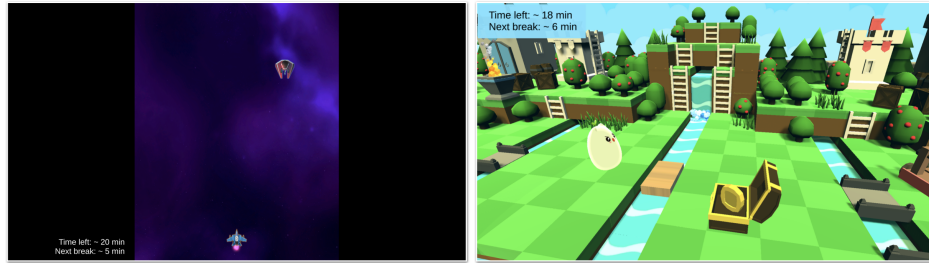- JUMP TIME (time it takes the character to jump onto the block): 0.4s, 0.6s, and 0.8s

Difeng Yu, Brandon Victor Syiem, Andrew Irlitti, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves



**Figure 8: Demonstration of the two game applications in Study 2. Left: Space Shooter, Right: Jump! Jump!**

These variables led to 4 $\{D_t, W_t\}$ combinations ($\{1.000s, 0.200s\}$, $\{1.000s, 0.300s\}$, $\{1.333s, 0.267s\}$, $\{1.333s, 0.400s\}$) and 3 $R_t$ values (which is the same as Jump Time 0.400s, 0.600s, and 0.800s), leading to 12 experimental conditions in total. Notably, the parameters were set to be larger than the previous settings to explore the model generalizability in a wider parameter range ($D_t \in [1.0s, 1.33s]$, $W_t \in [0.2s, 0.4s]$, $R_t \in [0.4s, 0.8s]$). Meanwhile, we still ensured that $D_t - R_t + W_t$ was larger than human reaction time ($\sim 0.3s$). Our pilot study suggested that some conditions could be easier than the previous scenarios, probably due to the larger $W_t$. The character's jump also represented non-linear movements (height calculated based on sine waves), which allowed us to assess more complex visual encoding of the temporal factors.

## 6.3 Method

We recruited participants from MTurk to perform the tasks. This allowed us to test our conclusions in less controlled experiment environments than immersive VR.

### 6.3.1 Participants, apparatus, and materials.
We received valid responses from 48 MTurk workers (24 for each scenario). The sample consists of 20 women and 28 men with an average age of 38.3 (SD = 10.5). The game applications were developed with Unity3D and deployed through WebGL. Participants performed the experiment with their own PC.

### 6.3.2 Procedure.
The average time for completing the whole study was around 60 minutes. After accepting the task on MTurk, they gave their informed consent, read the instructions of the experiment, filled in their demographic information, and were then directed to the web application.

Similarly to the first study, participants had a warm-up session to get familiar with $R_t$—they practiced for at least 5 trials and could practice more for as long as they liked. The first 10 starting trials of each block were randomly generated and discarded for participants to (re-)adapt to the formal experiment. The formal session lasted approximately 20 minutes for both applications. Breaks were given around every 5 minutes in Space Shooter, and when switching to a new character in Jump! Jump! (around 6 minutes). They were instructed to perform the task as accurately as possible and were informed that bonus payment would be given to highly accurate performers. The interval between the two trials was randomized

within [1.5s, 2s], and the applications would replay a trial if participants did not respond. After completing the experiment, participants submitted the redemption code to the MTurk form, and we compensated them with $15 after verifying their data.

## 6.4 Results - Space Shooter

### 6.4.1 Pre-processing and normality tests.
We collected 8640 data points (24 participants × 6 $\{D_t, W_t\}$ combinations × 3 $R_t$ × 20 repetitions) in the experiment. First, we removed the data of 3 participants whose overall error rate was abnormally high (> 70%). Second, we removed 93 outliers (1.08%) where user input time was outside of $mean \pm 3std$. in each condition. We were left with 7467 trials of data after the pre-processing. Among the 432 distributions, 78.0% of them were statistically normally distributed as shown by Shapiro-Wilk tests.

### 6.4.2 Model fitting and performance.
We fit the models with the `fitlm` function as in the first study. Table 4 summarizes the performance results, and the following equations demonstrate model fitting results.

$$\text{simple model} \begin{cases} \mu = 0.116 - 0.094D_t + 0.227R_t \\ \sigma = 0.038 + 0.066D_t \end{cases} \quad (11)$$

$$\text{interact model} \begin{cases} \mu = -0.076 + 0.161D_t + 1.505R_t - 1.704D_t \cdot R_t \\ \sigma = 0.038 + 0.066D_t \end{cases}$$
$$(12)$$

$$\text{hyp model} \begin{cases} \mu = 0.116 - 0.194(D_t - R_t) + 0.480W_t + 0.033R_t \\ \sigma = -0.002 + 0.070(D_t - R_t) - 0.017W_t + 0.333R_t \end{cases}$$
$$(13)$$

Overall, all the existing models could not properly explain the variations in $\mu$ (highest $R^2 = 0.53$). Incorporating $D_t \times R_t$ interaction as in the *interact model* was not helpful and produced worse results when compared to the *hyp model*. For $\sigma$, the *hyp model* achieved reasonable estimation with $R^2 = 0.71$ and lower information criterion measures, which indicated that adjusting $R_t$ and $W_t$ could also affect the variance of the selection distribution, as compared to the other two models which only considered $D_t$ when predicting $\sigma$. While the $R^2$ values for $\mu$ and $\sigma$ estimates were relatively low, the absolute prediction error (MAE) of the models was generally small. This resulted in relatively accurate estimates in user selection error rate (e.g., the *hyp model* achieved $R^2 = 0.90$ and $MAE = 6.88\%$).

### 6.4.3 Further analysis on $\mu$ fitting results.
The $R^2$ results of $\mu$ for all existing models were unexpectedly low (highest $R^2 = 0.53$),

**Table 4: Goodness-of-fit and information criteria comparison among the models for the two scenarios.**

| Scenario | Model | Selection Mean ($\mu$) | | | | Selection Std. ($\sigma$) | | | | Error Rate | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $R^2$ | MAE | AIC | BIC | $R^2$ | MAE | AIC | BIC | $R^2$ | MAE |
| Space Shooter | Simple model | 0.34 | 0.024 | -70.13 | -67.46 | 0.43 | 0.012 | -94.96 | -93.18 | 0.78 | 9.16% |
| | Interact model | 0.50 | 0.021 | -72.99 | -69.43 | 0.43 | 0.012 | -94.96 | -93.18 | 0.89 | 7.69% |
| | Hyp model | 0.53 | 0.021 | -74.14 | -70.58 | 0.71 | 0.009 | -102.89 | -99.33 | 0.90 | 6.88% |
| | LfDistance model | 0.97 | 0.006 | -119.47 | -115.02 | 0.71 | 0.009 | -102.89 | -99.33 | 0.96 | 5.40% |
| Jump! Jump! | Simple model | 0.71 | 0.033 | -38.81 | -37.35 | 0.64 | 0.011 | -61.29 | -60.32 | 0.52 | 11.06% |
| | Interact model | 0.96 | 0.013 | -60.40 | -58.46 | 0.64 | 0.011 | -61.29 | -60.32 | 0.92 | 4.59% |
| | Hyp model | 0.74 | 0.031 | -38.16 | -36.22 | 0.79 | 0.010 | -63.86 | -61.92 | 0.69 | 8.49% |

which motivated us to explore the potential issue. One difference between this scenario and the previous controlled experiment was that $D_t$ was extended from $[0.4s, 0.6s]$ to $[0.5s, 1.0s]$, because we wanted to make the online experiment "easier" as participants might have longer response time due to the lack of rigorous, lab-based experimental control. However, our analysis showed that the impact of $D_t - R_t$ (i.e., waiting time before an user input) on $\mu$ was no longer linear as in the *hyp model*: $\mu$ decreased with $D_t - R_t$ when $D_t - R_t < 0.55s$, and stayed more or less stable when $D_t - R_t > 0.55s$. We hypothesized that the impact of the ($D_t - R_t$) factor on $\mu$ saturated after passing a certain threshold (i.e., when the waiting time was long enough, it would no longer shift the center of the user input distribution). To model this effect, we used a logistic function $f(D_t - R_t) = a_1/(1 + e^{-k(D_t - R_t)})$ to transform the original linear term of $f(D_t - R_t) = a_1(D_t - R_t)$ in the *hyp model*. The added coefficient $k$ represents the steepness of the logistic function. We fit the model with fitnlm in Matlab and derived the following model (*LfDistance model*). The variation of $\mu$ was better captured by this model ($R^2 = 0.97$). The corresponding *AIC* and *BIC* values were also much lower, which illustrated the benefit of incorporating the logistic function and the extra coefficient $k$.

*LfDistance model*

$$\begin{cases} \mu = -0.02 - 2.198/(1 + e^{8.896(D_t - R_t)}) + 0.484W_t - 0.080R_t \\ \sigma = -0.002 + 0.070(D_t - R_t) - 0.017W_t + 0.333R_t \end{cases}$$

(14)

## 6.5 Results - Jump! Jump!

*6.5.1 Pre-processing and normality tests.* We collected 5760 data points (24 participants × 4 {$D_t$, $W_t$} combinations × 3 $R_t$ × 20 repetitions) in the experiment. We removed the data of 3 participants whose overall error rate was > 70% and 42 outliers (0.73%). We were left with 4998 trials of data after the pre-processing. Shapiro-Wilk tests showed that 81.3% were statistically normally distributed among the 288 distributions.

*6.5.2 Model fitting and performance.* After fitting the models with fitlm, we summarize the performance results in Table 4 and the model fitting results with the following equations.

*simple model* $\begin{cases} \mu = 0.530 - 0.346D_t + 0.033R_t \\ \sigma = -0.041 + 0.127D_t \end{cases}$ (15)

*interact model* $\begin{cases} \mu = -0.363 + 0.419D_t + 1.520R_t - 1.275D_t \cdot R_t \\ \sigma = -0.041 + 0.127D_t \end{cases}$

(16)

*hyp model* $\begin{cases} \mu = 0.530 - 0.398(D_t - R_t) + 0.207W_t - 0.365R_t \\ \sigma = -0.079 + 0.121(D_t - R_t) + 0.025W_t + 0.183R_t \end{cases}$

(17)

Overall, the *interact model* offered the best performance in $\mu$ ($R^2 = 0.96$ and 0.013) and error rate estimates ($R^2 = 0.92$ and $MAE = 4.59\%$). While the *hyp model* produced the highest $R^2$ (0.79) for $\sigma$ estimation, the *MAE*, *AIC*, and *BIC* values suggested that the additional parameters ($W_t$ and $R_t$) may not be major sources that altered the distribution variance. Moreover, *AIC* and *BIC* values were lower for the *simple model* as compared to the *hyp model*, indicating the additional $W_t$ factor might not be useful in this case. Noticeably, the coefficients of the *hyp model* in this scenario were rather close to the ones in the first study.

## 6.6 Discussion

We explored the generalizability of our models in two interactive game applications (Space Shooter and Jump! Jump!) through an online study. In Space Shooter, although the models produced reasonably accurate estimates for error rates (e.g., *hyp model* achieved $R^2 = 0.90$ and $MAE = 6.88\%$), all existing models failed to explain the variations in $\mu$ (highest $R^2 = 0.53$). A logistic function-based model variation was found to be more accurate ($R^2 = 0.97$) and lead to lower (better) information criterion measures. In Jump! Jump!, the *interact model* was found to be the most accurate on error rate prediction ($R^2 = 0.92$ and $MAE = 4.59\%$). Overall, the study results demonstrated that our models can still be helpful for conditions with more complex visual encoding, extended parameter ranges, and less-controlled environments. In the following, we discuss why we may need a logistic-like function to transform the impact of temporal factors on $\mu$ and the feasibility of generalizing the models and conclusions to larger $D_t$, $W_t$, and $R_t$ values.

*6.6.1 The logistic-like impact of temporal factors on user selection distributions.* In Space Shooter, we replaced the linear term of $a_1(D_t - R_t)$ with a logistic function-transformed term $a_1/(1 + e^{-k(D_t - R_t)})$. The logistic function was used to describe the effect of $D_t - R_t$ (i.e., waiting time before a user input) becoming saturated on $\mu$. Intuitively, while larger $D_t - R_t$ tended to shift the center of the distribution closer to the edge of the target, it would

not deviate the distribution center away from the target boundary as if modeled linearly. In other words, increasing $D_t - R_t$ would have a smaller and smaller effect on the distribution center as the center approaches the target edge. The fitting results ($R^2 = 0.97$ and $MAE = 0.006$) demonstrated the necessity of integrating the logistic term.

An analogy can be found in spatial target selection: when facing a relatively easy task (large $D_s$ and $W_s$), users will shift their distributions more towards the edge of the object [14, 50]. However, continuously increasing $D_s$ and $W_s$ will not move the distribution center infinitely far away from the target center, and the impact of $D_s$ and $W_s$ will get smaller as the distribution center is very close to the target edge [36].

As compared to Study 1, the additional logistic-like effect was likely caused by the fact that we prolonged the range of $D_t$ from [0.4s, 0.6s] to [0.5s, 1.0s] to compensate for the potential longer response time in a less-controlled experiment environment, which resulted in an easier task condition when $D_t$ was long. This means that logistic terms could be better suited for all of the temporal factors ($D_t$, $W_t$, and $R_t$) regarding their effects on the distribution (both $\mu$ and $\sigma$). However, we did not introduce the logistic terms to every temporal factor because it might bring too many additional coefficients. We expected that linear terms would have enough fitting power when a task condition was not too easy.

*6.6.2 The impact of larger $D_t$, $W_t$, and $R_t$.* We cross-compared our results from Jump! Jump! and the first study, and identified several similarities between them despite having extended the range of $D_t$, $W_t$, and $R_t$. All the coefficients of the *hyp model* in these two studies were rather close even though the experimental settings were completely different. Moreover, we found, as in the first study, $W_t$ had a marginal impact on the distribution center, and both $W_t$ and $R_t$ were not the main sources that changed the distribution variance. However, the $R^2$ values were not as high as in the first study for the *simple model* and the *hyp model*. We identified two potential reasons. First, the data were noisier in a less-controlled study. Second, the impact of $D_t$, $W_t$, and $R_t$ could be saturating with larger values as discussed in the last section.

The *interact model* achieved high performance in predicting the center of the distribution ($R^2 = 0.96$ and $MAE = 0.013$) and the selection error rates ($R^2 = 0.92$ and $MAE = 4.59\%$). Under this model, the coefficient of $R_t$ became ($2.046 - 3.798D_t$), which suggested $R_t$ delayed the distribution when $D_t < 1.192$ and pushed the distribution forward in time when $D_t > 1.192$. This threshold of $D_t$ could be dependent on $[D_t - R_t, D_t - R_t + W_t]$ (i.e., input interval for a successful selection) as discussed in the first study. In Jump! Jump!, the averaged input interval was [0.4, 0.7] when $R_t$ pushed the distribution backward and was [0.733, 1.033] when $R_t$ pushed the distribution forward. These intervals were slightly larger than the averaged intervals in Study 1 ([0.325, 0.475] and [0.475, 0.625]), which could be due to the differences in experimental settings.

## 7 IMPLICATIONS AND APPLICATIONS

Through the course of two user studies, we gathered a better understanding of how temporal factors including "distance" ($D_t$), "width" ($W_t$), and "delay" ($R_t$) may influence user selection distribution and selection error rates. We envision our models and conclusions to apply to various interactive applications that involve the selection of temporal targets.

Our models can enable automated playtesting [25] in various games that contain temporal target selection (e.g., shooting a moving target, jumping onto a moving block, hitting a minion within a fixed time window). For example, by estimating the user selection distribution and the selection error rate, a game designer no longer needs to rely on intuition to set the game difficulty but can determine the end-effect of adjusting game variables related to $D_t$, $W_t$, and $R_t$ computationally. Previous research suggested that predicting game difficulty is an uneasy task for game designers [16]. With a generalizable model, the designer can better predict how users will behave when adjusting the parameters to unseen conditions without costly user tests. Further, the parameters can even be adjusted adaptively in-game with our models for players having different skills, enabling more personalized game experiences. For example, the same game can be adjusted, based on the subtle changes in game parameters like in Study 2, to be more challenging for expert players while easier for novice players. According to the flow theory [11], balancing the level of player skill and game challenge leads to better game experiences. Based on our studies, we derived the following implications for modeling user selection behavior in temporal target selection.

**I1.** A good start of modeling a new temporal target selection scenario is to use the model based on our initial hypotheses of a mapping between temporal and spatial target selection (i.e., the *hyp model*). Its prediction performance in terms of user selection distribution and selection error rate was relatively robust and its prediction error (i.e., *MAE*) was acceptable even compared to a more complex model.

**I2.** The *hyp model* may be simplified to a simpler model variant that only considers a subset of temporal factors that have a significant impact on the selection distribution (i.e., the *simple model*). According to our results, $D_t$ seemed to be the primary source that altered the center and the variance of the selection distribution. Only considering a subset of temporal factors may achieve reasonable even comparable performance as a more complicated model.

**I3.** Consider the potential interaction effect between $D_t$ and $R_t$, as in the *interact model*. Our results suggested that when the input interval of a successful selection ($[D_t - R_t, D_t - R_t + W_t]$) allowed only a short reaction time (i.e., the bounding values were small), larger $R_t$ would delay the selection distribution. On the other hand, when the reaction time was relatively long (i.e., bounding values were large), larger $R_t$ could push the selection distribution forward in time. Leveraging the interaction term ($D_t \times R_t$) to describe this user behavior can lead to higher prediction accuracy.

**I4.** Consider the potential saturation effect of temporal factors, as in the *LfDistance model*. Our results showed that the impact of a temporal factor on selection distribution was not always linear—its influence may become saturated when the value was large. For example, if the temporal distance was too long, it would not always drag the distribution center linearly but have a smaller and smaller effect on it as the center approaches the target edge. Modeling the saturated term with a logistic

function can boost prediction performance when some task conditions seem to be a bit easy.

Additionally, our models and conclusions may benefit users of temporal target selection applications. By understating how temporal factors typically influence selection error rates as predicted by the model, a player can better counter-play challenging game scenarios. For example, based on our findings that a longer expected delay could push the distribution forward in time considerably when the temporal distance was large, we know that players typically rush their input when hitting a slowly approaching target with a slow bullet. In such cases, a player should try to execute the selection input a bit late to improve accuracy.

## 8 CONCLUSION AND FUTURE WORK

We have presented models to predict user selection distribution and error rate in temporal target selection tasks. We started the modeling process with a simple analogy between temporal and spatial target selection: we hypothesized that users react to temporal factors, including distance, width, and delay as how they treat the corresponding variables in spatial target selection. Through the course of two user studies (one controlled experiment in VR and one MTurk-based online experiment), we found similarities between spatial and temporal target selection. For example, increasing temporal distance ($D_t$) or width ($W_t$) shifted the center of the selection distribution and increased the distribution variance in a similar way as increasing the spatial distance ($D_s$) or width ($W_s$) do in a spatial task. Moreover, the impact of a factor may become saturated after crossing a threshold in both domains. While less explored in spatial target selection, we also discovered an interplay between the expected delay ($R_t$) and $D_t$ on the resulting distribution center.

We envision our models and conclusions will help designers and users when encountering relevant scenarios. Future research can extend our model to combine it with other spatial models to predict interaction tasks involving both spatial and temporal target selection. Future work can also explore more complex cursor and target movements (e.g., acceleration≠0 like an enemy target switches from walking to running). While our tasks did not limit the types of target movements, we tested our models mostly in scenarios with constant speeds of cursors and targets (expect for *Jump!Jump!*). Less-predictable movements may affect a user's estimation of temporal factors. Additionally, while our research is more empirically-driven, we hope our discussions and interpretations could inspire future theories that explain target selection behavior.

## OPEN-SOURCED DATASETS

We open-sourced our datasets on https://github.com/Davin-Yu/TemporalSelection-CHI2023 for replication and future research.

## REFERENCES

[1] Axel Antoine, Sylvain Malacria, and Géry Casiez. 2018. *Using High Frequency Accelerometer and Mouse to Compensate for End-to-End Latency in Indirect Interaction.* Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3173574.3174183

[2] James J Belisle. 1963. Accuracy, reliability, and refractoriness in a coincidence-anticipation task. *Research Quarterly. American Association for Health, Physical Education and Recreation* 34, 3 (1963), 271–281. https://doi.org/10.1080/10671188.1963.10613234

[3] 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. https://doi.org/10.1145/2470654.2466180

[4] Xiaojun Bi and Shumin Zhai. 2013. 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* (St. Andrews, Scotland, United Kingdom) *(UIST '13).* ACM, New York, NY, USA, 51–60. https://doi.org/10.1145/2501988.2502058

[5] Xiaojun Bi and Shumin Zhai. 2016. Predicting Finger-Touch Accuracy Based on the Dual Gaussian Distribution Model. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16).* ACM, New York, NY, USA, 313–319. https://doi.org/10.1145/2984511.2984546

[6] Rafal Bogacz, Eric Brown, Jeff Moehlis, Philip Holmes, and Jonathan D Cohen. 2006. The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological review* 113, 4 (2006), 700. https://doi.org/10.1037/0033-295X.113.4.700

[7] Mark Claypool. 2018. Game Input with Delay—Moving Target Selection with a Game Controller Thumbstick. *ACM Trans. Multimedia Comput. Commun. Appl.* 14, 3s, Article 57 (June 2018), 22 pages. https://doi.org/10.1145/3187288

[8] Mark Claypool, Ragnhild Eg, and Kjetil Raaen. 2017. Modeling user performance for moving target selection with a delayed mouse. In *International Conference on Multimedia Modeling.* Springer, 226–237. https://doi.org/10.1007/978-3-319-51811-4_19

[9] Jacob Cohen. 1988. *Statistical power analysis for the behavioral sciences.* Routledge. https://doi.org/10.4324/9780203771587

[10] Virginie Crollen, Stéphane Grade, Mauro Pesenti, and Valérie Dormal. 2013. A common metric magnitude system for the perception and production of numerosity, length, and duration. *Frontiers in psychology* 4 (2013), 449. https://doi.org/10.3389/fpsyg.2013.00449

[11] Mihaly Csikszentmihalyi and Mihaly Csikszentmihaly. 1990. *Flow: The psychology of optimal experience.* Vol. 1990. Harper & Row New York.

[12] 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* (Osaka, Japan) *(VRST '21).* Association for Computing Machinery, New York, NY, USA, Article 42, 10 pages. https://doi.org/10.1145/3489849.3489853

[13] 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.

[14] Tovi Grossman and Ravin Balakrishnan. 2005. A probabilistic approach to modeling two-dimensional pointing. *ACM Transactions on Computer-Human Interaction (TOCHI)* 12, 3 (2005), 435–459. https://doi.org/10.1145/1096737.1096741

[15] Tovi Grossman, Nicholas Kong, and Ravin Balakrishnan. 2007. *Modeling Pointing at Targets of Arbitrary Shapes.* Association for Computing Machinery, New York, NY, USA, 463–472. https://doi.org/10.1145/1240624.1240700

[16] Christoffer Holmgård, Michael Cerny Green, Antonios Liapis, and Julian Togelius. 2018. Automated playtesting with procedural personas through MCTS with evolved heuristics. *IEEE Transactions on Games* 11, 4 (2018), 352–362. https://doi.org/10.1109/TG.2018.2808198

[17] Jin Huang and Byungjoo Lee. 2019. Modeling Error Rates in Spatiotemporal Moving Target Selection. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19).* Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3290607.3313077

[18] Jin Huang, Feng Tian, Xiangmin Fan, Xiaolong (Luke) Zhang, and Shumin Zhai. 2018. Understanding the Uncertainty in 1D Unidirectional Moving Target Selection. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18).* Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173811

[19] Jin Huang, Feng Tian, Nianlong Li, and Xiangmin Fan. 2019. Modeling the Uncertainty in 2D Moving Target Selection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19).* Association for Computing Machinery, New York, NY, USA, 1031–1043. https://doi.org/10.1145/3332165.3347880

[20] Ricardo Jota, Albert Ng, Paul Dietz, and Daniel Wigdor. 2013. *How Fast is Fast Enough? A Study of the Effects of Latency in Direct-Touch Pointing Tasks.* Association for Computing Machinery, New York, NY, USA, 2291–2300. https://doi.org/10.1145/2470654.2481317

[21] Yu-Jung Ko, Hang Zhao, Yoonsang Kim, IV Ramakrishnan, Shumin Zhai, and Xiaojun Bi. 2020. Modeling Two Dimensional Touch Pointing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* 858–868. https://doi.org/10.1145/3379337.3415871

[22] Steven Komarov, Katharina Reinecke, and Krzysztof Z. Gajos. 2013. Crowdsourcing Performance Evaluations of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13).* Association for Computing Machinery, New York, NY, USA, 207–216. https://doi.org/10.1145/2470654.2470684

[23] Byungjoo Lee, Sunjun Kim, Antti Oulasvirta, Jong-In Lee, and Eunji Park. 2018. Moving Target Selection: A Cue Integration Model. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada)

*(CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article 230, 12 pages. https://doi.org/10.1145/3173574.3173804

[24] Byungjoo Lee and Antti Oulasvirta. 2016. Modelling Error Rates in Temporal Pointing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1857–1868. https://doi.org/10.1145/2858036.2858143

[25] Injung Lee, Hyunchul Kim, and Byungjoo Lee. 2021. Automated Playtesting with a Cognitive Model of Sensorimotor Coordination. In *Proceedings of the 29th ACM International Conference on Multimedia* (Virtual Event, China) *(MM '21)*. Association for Computing Machinery, New York, NY, USA, 4920–4929. https://doi.org/10.1145/3474085.3475429

[26] Injung Lee, Sunjun Kim, and Byungjoo Lee. 2019. Geometrically Compensating Effect of End-to-End Latency in Moving-Target Selection Games. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300790

[27] Jonna Loeffler, Rouwen Cañal-Bruland, Anna Schroeger, J Walter Tolentino-Castro, and Markus Raab. 2018. Interrelations between temporal and spatial cognition: The role of modality-specific processing. *Frontiers in psychology* 9 (2018), 2609. https://doi.org/10.3389/fpsyg.2018.02609

[28] I Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction* 7, 1 (1992), 91–139. https://doi.org/10.1207/s15327051hci0701_3

[29] Atsuo Murata. 1999. Extending effective target width in Fitts' law to a two-dimensional pointing task. *International journal of human-computer interaction* 11, 2 (1999), 137–152. https://doi.org/10.1207/S153275901102_4

[30] Hiroki Nakamoto and Shiro Mori. 2012. Experts in fast-ball sports reduce anticipation timing cost by developing inhibitory control. *Brain and cognition* 80, 1 (2012), 23–32. https://doi.org/10.1016/j.bandc.2012.04.004

[31] Eunji Park and Byungjoo Lee. 2020. An Intermittent Click Planning Model. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376725

[32] Andriy Pavlovych and Carl Gutwin. 2012. Assessing Target Acquisition and Tracking Performance for Complex Moving Targets in the Presence of Latency and Jitter. In *Proceedings of Graphics Interface 2012* (Toronto, Ontario, Canada) *(GI '12)*. Canadian Information Processing Society, CAN, 109–116.

[33] Réjean Plamondon and Adel M Alimi. 1997. Speed/accuracy trade-offs in target-directed movements. *Behavioral and brain sciences* 20, 2 (1997), 279–303. https://doi.org/10.1017/S0140525X97001441

[34] 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.

[35] Kilian Semmelmann and Sarah Weigelt. 2017. Online psychophysics: Reaction time effects in cognitive experiments. *Behavior Research Methods* 49, 4 (2017), 1241–1260. https://doi.org/10.3758/s13428-016-0783-4

[36] R William Soukoreff and I Scott MacKenzie. 2004. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *International journal of human-computer studies* 61, 6 (2004), 751–789. https://doi.org/10.1016/j.ijhcs.2004.09.001

[37] SS Stevens and Hilda B Greenbaum. 1966. Regression effect in psychophysical judgment. *Perception & Psychophysics* 1, 5 (1966), 439–446. https://doi.org/10.3758/BF03207424

[38] Robert Teghtsoonian and Martha Teghtsoonian. 1978. Range and regression effects in magnitude scaling. *Perception & Psychophysics* 24, 4 (1978), 305–314. https://doi.org/10.3758/BF03204247

[39] Warren H Teichner. 1954. Recent studies of simple reaction time. *Psychological Bulletin* 51, 2 (1954), 128. https://doi.org/10.1037/h0060900

[40] James R Tresilian. 2005. Hitting a moving target: perception and action in the timing of rapid interceptions. *Perception & Psychophysics* 67, 1 (2005), 129–149. https://doi.org/10.3758/BF03195017

[41] Hiroki Usuba, Shota Yamanaka, and Homei Miyashita. 2018. User Performance by the Difference between Motor and Visual Widths for Small Target Pointing. In *Proceedings of the 10th Nordic Conference on Human-Computer Interaction* (Oslo, Norway) *(NordiCHI '18)*. Association for Computing Machinery, New York, NY, USA, 161–169. https://doi.org/10.1145/3240167.3240171

[42] Hiroki Usuba, Shota Yamanaka, and Homei Miyashita. 2019. Touch Pointing Performance for Uncertain Touchable Sizes of 1D Targets. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services* (Taipei, Taiwan) *(MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA, Article 20, 8 pages. https://doi.org/10.1145/3338286.3340131

[43] Vincent Walsh. 2003. A theory of magnitude: common cortical metrics of time, space and quantity. *Trends in cognitive sciences* 7, 11 (2003), 483–488. https://doi.org/0.1016/j.tics.2003.09.002

[44] Alan Traviss Welford. 1968. Fundamentals of skill. (1968).

[45] Alan M Wing and Alfred B Kristofferson. 1973. Response delays and the timing of discrete motor responses. *Perception & Psychophysics* 14, 1 (1973), 5–12. https://doi.org/10.3758/BF03198607

[46] Jacob O. Wobbrock, Edward Cutrell, Susumu Harada, and I. Scott MacKenzie. 2008. An Error Model for Pointing Based on Fitts' Law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 1613–1622. https://doi.org/10.1145/1357054.1357306

[47] David L Woods, John M Wyma, E William Yund, Timothy J Herron, and Bruce Reed. 2015. Factors influencing the latency of simple reaction time. *Frontiers in human neuroscience* 9 (2015), 131. https://doi.org/10.3389/fnhum.2015.00131

[48] Shota Yamanaka and Hiroki Usuba. 2020. Rethinking the Dual Gaussian Distribution Model for Predicting Touch Accuracy in On-screen-start Pointing Tasks. *Proceedings of the ACM on Human-Computer Interaction* 4, ISS (2020), 1–20. https://doi.org/10.1145/3427333

[49] 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* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 6, 20 pages. https://doi.org/10.1145/3526113.3545632

[50] Difeng Yu, Hai-Ning Liang, Xueshi Lu, Kaixuan Fan, and Barrett Ens. 2019. Modeling Endpoint Distribution of Pointing Selection Tasks in Virtual Reality Environments. *ACM Trans. Graph.* 38, 6, Article 218 (Nov. 2019), 13 pages. https://doi.org/10.1145/3355089.3356544

[51] Difeng Yu, Qiushi Zhou, Benjamin Tag, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2020. Engaging Participants during Selection Studies in Virtual Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE. https://doi.org/10.1109/VR46266.2020.00071

[52] Ziyue Zhang, Jin Huang, and Feng Tian. 2020. Modeling the Uncertainty in Pointing of Moving Targets with Arbitrary Shapes. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. 1–7. https://doi.org/10.1145/3334480.3382875