Modeling Visually-Guided Aim-and-Shoot Behavior in First-Person Shooters

June-Seop Yoon[a], Hee-Seung Moon[b], Ben Boudaoud[c], Josef Spjut[c], Iuri Frosio[c], Byungjoo Lee[a,*], Joohwan Kim[c,*]

[a] *Yonsei University, Republic of Korea*
[b] *Chung-Ang University, Republic of Korea*
[c] *NVIDIA, USA*

Abstract

In first-person shooters, players aim by aligning the crosshair onto a target and shoot at the optimal moment. Since winning a match is largely determined by such aim-and-shoot skills, players demand quantitative evaluation of the skill and analysis of hidden factors in performance. In response, we build a simulation model of the cognitive mechanisms underlying aim-and-shoot behavior based on the computational rationality framework. Unlike typical aimed movements in HCI, such as pointing, the aim-and-shoot offers a unique task scenario: as players move the mouse with their hand, the first-person view camera rotates, which in turn directly affects the target's visible position on the screen. That is, during the aim-and-shoot process, players experience a stronger coupling between perception and motor processes. To realistically simulate such complex mechanisms, we model players' perceptual, decision-making, and motor processes more sophisticatedly than any existing model. Model fitting based on amortized inference showed that our model could successfully reproduce the behavior of 20 FPS players (10 professionals) on several key measures, outperforming a baseline. Additionally, model fit parameters revealed that professionals had distinct cognitive or motivational characteristics.

Keywords: Human-Computer Interaction, Computational Rationality, Reinforcement Learning, Esports, First-person Shooters

*Corresponding authors
Email addresses: jsyoon.k5@yonsei.ac.kr (June-Seop Yoon), hsmoon@cau.ac.kr (Hee-Seung Moon), bboudaoud@nvidia.com (Ben Boudaoud), jspjut@nvidia.com (Josef Spjut), ifrosio@nvidia.com (Iuri Frosio), byungjoo.lee@yonsei.ac.kr (Byungjoo Lee), sckim@nvidia.com (Joohwan Kim)

1. Introduction

First-person shooter (FPS) is a video game genre played by nearly a billion gamers over the past few decades (Newzoo, 2022). In an FPS game, a player controls a character from a first-person perspective to shoot and destroy enemies. There is usually a weapon's crosshair in the center of the first-person view, and the player can rotate the view camera to aim at the enemy. In the competitive session, slight skill differences can decide victory or defeat among similarly ranked players. This extreme competition has led players to discuss necessary skills for winning and methods to improve them (Lamers James and O'Connor, 2023; Park et al., 2021).

The *aim-and-shoot* skill, i.e., the ability to quickly align the crosshair to an enemy and fire the weapon with the correct timing, is likely the most basic and essential among skills in FPS games (Park et al., 2021; Warburton et al., 2023; Rogers et al., 2024). Even in professional matches, a player's excellent aim-and-shoot skill often significantly impacts the game's outcome. For instance, Polish `CS:GO` (Counter-Strike: Global Offensive) professional player *Filip "Neo" Kubski* showcased his superior aim-and-shoot skill by eliminating all opponents in a 1 versus 5 situation on match point of the semi-final[1], leading his team to the finals. For this reason, millions of FPS players dedicate a significant amount of hours to aim training with dedicated software[2] (Roldan and Prasetyo, 2021; Meta, 2018; Rogers et al., 2024). Esports teams often hire professional coaches to help athletes with aim training and evaluation. Local esports academies are also being established to offer classes centered on aim-and-shoot skills.

Despite its high practical interest and importance, *scientific explanations* of the aim-and-shoot behavior in FPS are rare in online communities and academia (Park et al., 2021). Players today are not given a scientific theory to explain the cognitive mechanisms behind aim-and-shoot behavior, nor a technology to analyze the *latent* factors that determine aim-and-shoot performance. `AimLab`, an aim-and-shoot training application with millions of active users, only provides descriptive statistics on visible behavioral differences between players (e.g., shot accuracy, completion time) (Statespace, 2018; Roldan and Prasetyo, 2021). This lack of solid analytical foundation has led amateur players to train by simply mimicking the in-game behavior of professional players or relying on untested guidelines floating around in the community (Park et al., 2021). Also, in professional players' training and coaching process, scientific evidence is used to a limited extent (Horst et al., 2021; Rerick and Moritz, 2023; Koposov et al., 2020).

The ultimate goal of our study is to build a scientific model that can explain the cognitive mechanisms underlying aim-and-shoot behavior, enabling deeper analysis and evaluation of a player's aim-and-shoot skills. Our model-

[1] DreamHack ZOWIE Open Bucharest 2016 Semi Final #2, *Virtus.pro* vs. *dignitas*, Set 3 Round 29/30 (*Virtus.pro* up 15-13) (DreamHack, 2016)

[2] Aim Lab (Statespace, 2018), Aim Hero (ProGames Studio, 2016), 3D Aim Trainer (Steelseries, 2021)

ing is grounded in the computational rationality (CR) framework (Gershman et al., 2015; Lewis et al., 2014; Oulasvirta et al., 2022): we assume that human behavior is the result of bounded rationality that emerges under external (i.e., task environment), internal (i.e., cognitive capacities), and utility (or reward) constraints. Accordingly, we implement an artificial aim-and-shoot agent with human-like perception, decision-making, and motor capabilities, which can be adjusted by model parameters. We use deep reinforcement learning (RL) to learn the agent's optimal aim-and-shoot policy to maximize the collection of rewards that are presumably in common with those optimized by human players (e.g., shorter execution time, higher hit rate). Eventually, the trained agent can realistically predict how a player with specific perceptual and motor characteristics and specific motivational arousal will behave in a given aim-and-shoot task scenario. By fitting a model to an actual player's input behavioral data, we can also infer the player's cognitive and motivational characteristics.

Unlike typical aimed movements in HCI, such as pointing, the aim-and-shoot offers players a unique task scenario: hand (or mouse) movement during the aim-and-shoot process directly affects the visible position of the target on the screen, as it causes rotation of the first-person view. That is, during the aim-and-shoot process, players experience a strong *coupling* between perception (i.e., estimation of the target's current and future position) and motor processes (i.e., planning and execution of hand movements). This can make aim-and-shoot movements qualitatively distinct from typical aimed movements and provide additional challenges to players' perceptual and motor processes. For example, when players want to estimate the future position of a target to build a new motor plan, they must take into account that the previous motor plan (already in execution) will be disturbed by motor noise. Additionally, as targets on the screen show more complex movements due to the added influence of camera rotation, players are also required to have more precise and predictive gaze control skills in synchronization with motor processes. These issues do not exist in typical 2D aimed movements, which is why a simple conversion of existing CR models of aimed movements, such as the point-and-click model (Do et al., 2021), is not sufficient for modeling aim-and-shoot behavior.

To faithfully reproduce the underlying mechanisms of aim-and-shoot behavior, we model human perception, decision-making, and motor functions more elaborately than any other aimed movement model (see Table 1). More specifically, the model proposed in this study is implemented with the point-and-click model, which can be considered a state-of-the-art (SOTA) CR model of aimed movement, presented by Do et al. in 2021 (Do et al., 2021), as a starting base. The point-and-click model provides a solid, broad description of human motor and decision-making functions that must be considered in modeling not only aim-and-shoot behavior but also all aimed movements such as intermittent predictive control (Bye and Neilson, 2008; Bye, 2009), signal-dependent motor noise (Schmidt et al., 1979), and click decision-making (Park and Lee, 2020). On top of that base, our model implements the following mechanisms that are not considered or only limitedly considered in existing models: (1) perception of target

Table 1: Comparison of our aim-and-shoot model with existing aimed movement models

Category	**Items**	**Aim-and-Shoot Model**	**Baseline Model**	Point-and-Click Model[1,2]	Müller model[3]	Fischer model[4]	Jokinen model[2,5]	Cheema model[6]	Ikkala model[7]	Gonzalez model[8]
Perceptual	Noise in peripheral vision	●	○	○	○	○	●	○	○	○
	Noise in speed perception	● ←	● ←	●	○	○	○	○	○	○
	Gaze control dynamics	●	○	○	○	◑	◑	○	○	○
	Target perception with efference copy	●	○	○	○	○	○	○	○	○
Motor	Signal-dependent noise (hand)	● ←	● ←	●	○	●	●	○	○	●
	Intermittent motor control	● ←	● ←	●	○	○	○	○	○	○
	Predictive motor control	● ←	● ←	●	○	○	○	○	○	●
Decision making	Variability in reaction time	●	○	○	○	○	○	○	○	○
	Click planning & execution	●	● ←	●	○	○	●	○	○	○
Model fitting	Amortized inference engine	●	●	●	○	○	●	○	○	○

● Fully considered, ◑ Partially considered, ○ Not considered, ⟵ Inherited from
[1](Do et al., 2021), [2](Moon et al., 2023),[3](Müller et al., 2017), [4](Fischer et al., 2021), [5]Jokinen et al. (2021), [6](Cheema et al., 2020), [7](Ikkala et al., 2022), [8](Gonzalez et al., 2022; Gonzalez and Follmer, 2023)

future position based on efference copy[3] of previous motor plan, (2) movement of gaze on the screen and its kinematics (i.e., main sequence), (3) noise in peripheral vision, and (4) variability in reaction time (Hultsch et al., 2002). The model has eight free parameters representing the cognitive characteristics of the simulated agent, including motor and visual noise, from which it can replicate a wide range of intra-player and inter-player variability. By applying multi-task RL techniques (Moon et al., 2022), the model can simulate optimal aim-and-shoot behavior without additional training even when parameters change.

For the model evaluation, we collected aim-and-shoot behavior from 20 FPS players, consisting of 10 professionals and 10 amateurs. A model that simply converted the point-and-click model to a 3D aim-and-shoot scenario without the aforementioned modifications[4] was used as the baseline for evaluation. Utilizing neural inference techniques (i.e., amortized inference) (Moon et al., 2023),

[3]Motor plans for human movement are updated intermittently, and the copy of the previous motor plan that exists at the time a new motor plan is created is called an efference copy. Efference copies do not contain motor noise because they are copies of the unexecuted plan.

[4]Therefore, the baseline model cannot simulate gaze movements.

both models were fitted to the aim-and-shoot behavior dataset. As a result, our model outperformed the baseline in replicating key features of aim-and-shoot behavior (e.g., trial completion time, shooting success rate, saccadic deviation[5], movement trajectory, etc.). More specifically, our model was able to replicate aim-and-shoot completion time, shooting success rate, and saccadic deviation with mean absolute error (MAE) accuracy of 43.5 ms, 10.8%p, and 0.80°, respectively (see Table 6 for all results in more detail). By analyzing the inferred parameters, we found evidence that the high performance of professional players may result from their distinct motivational characteristics as well as their lower levels of cognitive noise.

We summarize the contributions of this study more specifically as follows:

- A novel CR model was proposed that can simulate player behavior in aim-and-shoot tasks in which perceptual and motor processes are more strongly coupled than typical aimed movements.
- The explanatory and predictive power of the model was tested by fitting it to aim-and-shoot behavior data collected from 20 players, including 10 professionals.
- By analyzing the inferred parameters, we provide initial evidence on how professional players can achieve higher aim-and-shoot performance than amateurs.

2. Backgrounds

2.1. Basic Mechanisms of Human Aimed Movement Control

The underlying cognitive mechanisms by which humans perform aimed movements have been covered extensively in previous studies, showing that a human's aimed movement is performed in four steps (Bye and Neilson, 2008; Bye, 2009; Do et al., 2021; Lee et al., 2020). First in the sensory analysis (SA) step, the position of the target is first estimated based on the given sensory signals (i.e., visual perception). Then, in the response planning (RP) step, an appropriate motor plan is built to move the pointer or body to the estimated target position. In the response execution (RE) step, the built motor plan is executed. At last, if a click action is required at the end of an aimed movement (as in the point-and-click or aim-and-shoot cases), the appropriate click timing must also be estimated during motor planning and execution (Park and Lee, 2020; Do et al., 2021).

Since a non-negligible level of cognitive noise commonly affects each step, achieving the goal of the aimed movement by only executing a single motor plan (i.e., a ballistic movement) is not easy. Therefore, humans periodically update the existing motor plan while the current one is executed. The update period is known to be approximately 100 ms due to the human psychological refractory period (Smith, 1967). This *intermittent motor control* process is widely observed

[5]The visual angle (°) between the central crosshair and the gaze at the moment when the gaze was most deviated from the crosshair

in various types of aimed movements (Martín et al., 2021; Wang et al., 2021; Park and Lee, 2020). The BUMP (Basic Unit of Motor Production) theory (Bye and Neilson, 2008; Bye, 2009; Do et al., 2021) is a useful control model (Müller et al., 2017) that can reliably replicate the general mechanism of human intermittent motor control. BUMP is particularly suitable for modeling that relies on repeated simulations; it includes an algorithm that quickly generates an optimal motion plan that satisfies the minimum acceleration constraints (a.k.a., Optimal Trajectory Generation or OTG function) (Sparrow and Newell, 1998; Jiang et al., 2002; Neilson and Neilson, 2005). BUMP has been adopted in point-and-click model, the latest CR model of human aimed movement (Do et al., 2021), and also applied in this study.

During the intermittent control process, humans have a copy of the previously constructed motor plan, called an *efference copy* (Angel, 1976). This record of the motor plan is kept in the human internal memory and is used to predict the consequences that the execution of the motor plan will bring to the environment (Groß et al., 2002; van Beers et al., 2002; Blakemore et al., 1998). In the aim-and-shoot process, efference copy helps players predict how the target's position on the screen will change as the view camera rotates if the motor plan is executed without noise. In general aimed movements in HCI, such as pointing, there is less need for motor efference copy to be considered in the perception process because the positions of the pointer and target can be controlled relatively independently from each other.

2.2. Challenges in Visual Perception in Aimed Movements

An accurate estimate of the target position is essential for the success of aimed movements, but various limitations of human visual perception may affect it. One dominant factor is the limited resolution of *peripheral vision*. The farther the target is from the gaze position, the less precise the estimate becomes (Strasburger et al., 2011; Wells-Gray et al., 2016; Hussain et al., 2015). If players move their gaze closer to the target, peripheral noise can be minimized, but this is not an easy task because the target moves complexly on the screen due to camera rotation during the aim-and-shoot process. Another problem players may encounter is inaccurate estimates of the crosshair's position while gazing at the target. To overcome this problem, some players seek to use more visible crosshair colors and shapes (Rawat, 2021). The peripheral vision effect can be generally modeled (Hussain et al., 2015) as the standard deviation (σ_p) of the target position estimate distribution increasing in proportion to the distance from the gaze center to the target (ϵ, in visual angle): $\sigma_p \propto \epsilon$.

Rapid gaze movements towards a target are also affected by kinematic constraints, resulting in saccadic movements. A regular relationship is observed between saccadic movements' amplitude (A) and peak velocity (V), which is called the *saccadic main sequence* (Bahill et al., 1975). These kinematic constraints can be critical in aim-and-shoot tasks, where the situation changes rapidly. According to a previous study (Gibaldi and Sabatini, 2021), the main sequence can be approximately described by the linear relationship: $V = a \cdot A + b$, where a and b are free parameters.

In saccadic movements, the landing position of the gaze is known to be determined stochastically due to signal-dependent motor noise (Kowler, 1990). More specifically, the larger the amplitude of gaze movement, the higher the variability of the landing position. The EMMA (Eye Movements and Movement of Attention) model (Salvucci, 2001), which has been widely adopted in HCI studies (Jokinen et al., 2017, 2021; Fleetwood and Byrne, 2006), assumes that the landing position of the gaze at the end of a saccadic movement is sampled from a Gaussian distribution centered on the target position, and its standard deviation increases in proportion to the amplitude of the movement in visual angle.

When a target is moving, as in the aim-and-shoot task, aimed movements toward the target must be based on *predictive control*, i.e., motor plans should be built toward the *future* position of the target. To do this, players have to visually encode the target's movement pattern. If the target's movement pattern is limited to constant-speed linear motion (as assumed in this study), the task for players to estimate the target's future position is simplified to visually estimating the target's speed. The performance of humans in visually perceiving target speed can be explained through Stocker's model (Stocker and Simoncelli, 2006; Wang and Li, 2007). According to the model, the precision in speed estimation decreases as the target speed increases or its contrast with the background decreases.

2.3. Challenges in Motor Execution and Click Decision-Making in Aimed Movements

Even in the case of perfect target localization and motor planning, perfect execution of the motor plan is unlikely because of motor noise and external disturbances. One important limitation comes from signal-dependent motor noise (Schmidt et al., 1979); noise proportional to the average speed of the motor plan is added to the location of arrival (Lin and Tsai, 2015). When pointing toward a stationary target, as in general office work, users can reduce the motor speed when the pointer gets close enough to the target, minimizing the impact of motor noise. In contrast, when the target is moving at high speeds, such as in an aim-and-shoot scenario, the impact of motor noise becomes more significant and persistent as players must constantly move their hands to track the target with the crosshair.

Most aimed movements in HCI, including aim-and-shoot, require target acquisition with a button input (click) at the end of the movement. If the click timing is not properly planned and executed, the task will likely fail even if the target is reached. According to the latest model of human clicking behavior (Park and Lee, 2020), players plan to press a button at a specific moment while the previous motor plan is being executed. In this process, the probability of a successful click is determined by the inherent timing ability of the players (Lee and Oulasvirta, 2016). The further away the button input timing is planned from the present, the lower the precision, which results from the *scalar property* of the human internal clock (Wing and Kristofferson, 1973b,a; Gibbon et al., 1984). People with higher timing abilities can execute button inputs at planned

timing more precisely than others. For example, more precise button input abilities have been observed in musicians (Lee and Oulasvirta, 2016) and gamers (Park and Lee, 2020).

2.4. Modeling Input Behavior and Performance of Computer Users

Building scientific models that predict user input behavior and performance is an important and long-standing research topic in HCI. By modeling input behavior, we gain rich insights to improve (Kim et al., 2018), optimize (Oulasvirta, 2017; Lee et al., 2020), and personalize (Gajos and Weld, 2004; Sarcar et al., 2018) interactions. Scientific models that successfully predict and explain user behavior in tasks such as point-and-click (MacKenzie, 1992; Wobbrock et al., 2008; Park and Lee, 2020), steering (Accot and Zhai, 1997), reaction (Seow, 2005), and moving-target acquisition (Lee et al., 2018; Lee, 2022) have already been built and validated. Traditionally, they are based on simple regression models with a limited number of parameters that predict aggregate statistics such as mean trial completion time or error rate (e.g., Fitts' law (Fitts, 1954)). However, if we want to identify the cognitive mechanisms underlying input behavior and also predict or simulate behavior at higher temporal and spatial resolution, more advanced computational modeling techniques should be considered.

We model the aim-and-shoot behavior based on the theory of *computational rationality* (CR) (Gershman et al., 2015; Lewis et al., 2014; Oulasvirta et al., 2022). Rooted in the theory of Bounded Rationality (Simon, 1990) proposed by Herbert Simon, CR suggests that user behavior is the outcome of maximizing a particular utility (or reward) function under inherent perceptual, cognitive, and motor constraints. Technically, CR models are implemented as virtual agents equipped with perceptual and motor constraints similar to real users. Through deep RL, the agent can be trained to follow an optimal action policy that maximizes the expected reward (Do et al., 2021; Moon et al., 2022). As a result, the agent can mimic the actual user behavior. CR models have successfully replicated the behavior of real users in various tasks such as typing (Jokinen et al., 2021), point-and-click (Do et al., 2021; Moon et al., 2022), visual search (Chen et al., 2015; Acharya et al., 2017), button-pressing (Oulasvirta et al., 2018), and mid-air pointing (Cheema et al., 2020; Ikkala et al., 2022). CR model proposed here is the first one devoted to aim-and-shoot behavior.

3. The Aim-and-Shoot Agent Model

In this section, we detail the process of modeling the cognitive mechanisms underlying aim-and-shoot behavior. The aim-and-shoot task scenario is first explained, an overview of the model's architecture is presented, and then the implementation of each sub-module is explained.

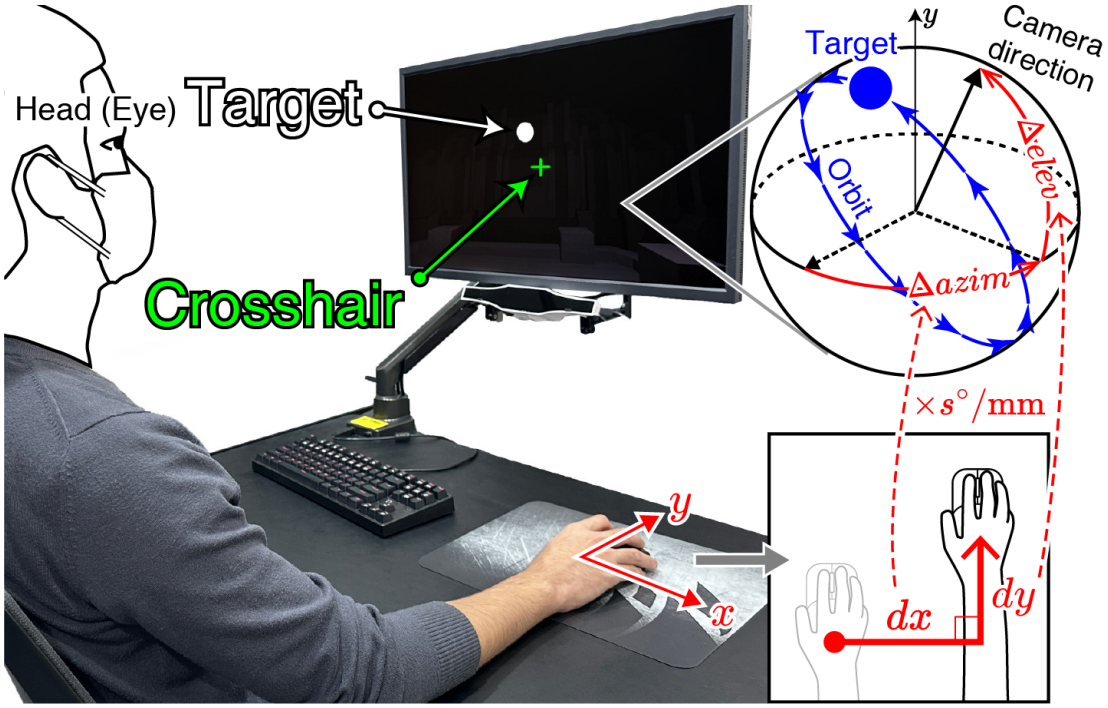

Figure 1: The aim-and-shoot task scenario modeled in this study

3.1. The Aim-and-Shoot Task Scenario

We assume that the simulated agent is using a display of size 53.13 cm (W) × 29.88 cm (H) in a typical desktop environment (see Figure 1). The mouse is located just below the center of the agent's right hand. The agent interacts with a simplified aim-and-shoot task defined in a 3D virtual game environment. When a trial is initiated, a sphere target spawns within the monitor space. The agent can move its gaze position during the trial. The agent has to rotate the first-person camera by moving the mouse to align the sphere and the crosshair (i.e., *aim*) and then click to eliminate the target (i.e., *shoot*). The trial is considered successful if the target is under the crosshair when clicking (hit). Otherwise, it is a failed trial (miss). Regardless of the shoot result, the trial ends when the agent performs a click action. The mouse displacement (in mm) is multiplied by the in-game sensitivity (in °/mm) to determine the change in the camera's azimuth and elevation angles (in °). We select 1°/mm as the agent's mouse sensitivity among the optimal range for FPS aiming as reported by the previous study (Boudaoud et al., 2022). We assume the agent does not lift the mouse, and the head position is fixed during the trial. Translation of the view camera is not allowed.

The virtual environment is rendered with a field of view of 103° in width and 70° in height. The crosshair is fixed at the center of the view. At every trial,

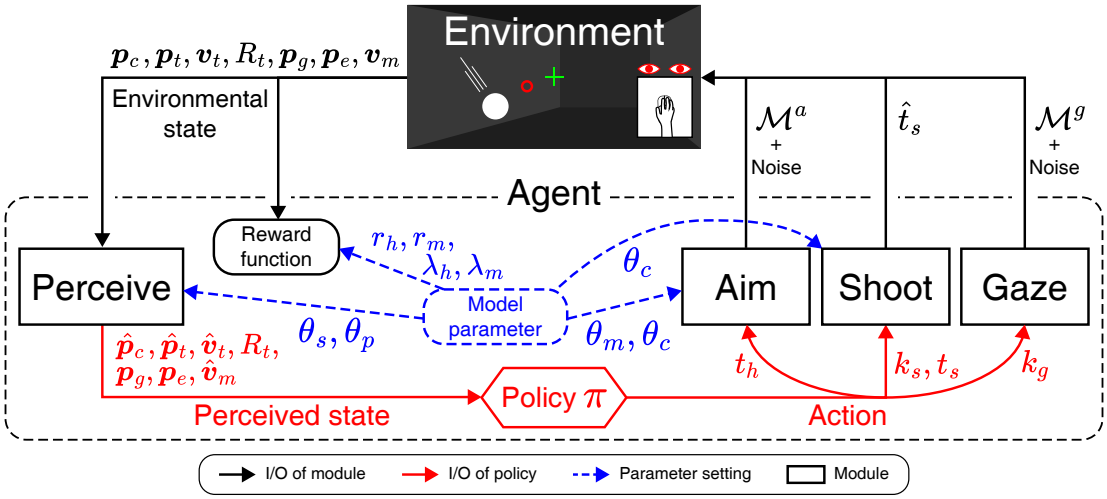

Figure 2: Overview of the model architecture

the initial task conditions are randomized as follows. The camera direction is randomly initialized to have 1.2° or smaller angle size with the reference direction (both azimuth and elevation angles are 0°). The target sphere, with a random radius from 4 mm to 13 mm on the display, spawns at a random position within the range 2° to 37° in azimuth and 1° to 21° in elevation to the reference direction. The target rotates on an orbit equidistant to the camera in 3D space, on a randomly chosen orbit plane, with an angular speed ranging from 0 °/s to 40 °/s. On the monitor, this results in the target linearly moving in a random direction at speed from 0 cm/s to 15 cm/s approximately. Please refer to Appendix A.1 for more details on the task scenario covered in this study.

3.2. Overview of The Aim-and-Shoot Model

3.2.1. Model Architecture

The model consists of four sub-modules: (1) *Perceive*, (2) *Aim*, (3) *Gaze*, and (4) *Shoot* (see Figure 2). The integrated system of four modules constitutes our virtual aim-and-shoot agent. Through the *Perceive* module, the agent estimates the position of the crosshair and the position and speed of the target. Based on the information estimated through the Perceive module, the *Aim* module generates a motor plan of view camera rotation ($\mathcal{M}^a$) that is expected to align the target center on the crosshair. Meanwhile, the *Gaze* module simulates the process by which the agent plans and executes gaze movement (i.e., gaze control plan $\mathcal{M}^g$). The *Shoot* module determines whether and when to perform mouse button input while the motor plans built in the Aim and Gaze modules are being executed. At each decision-making step, the agent's policy function ($\boldsymbol{\pi}$) determines how the Aim, Gaze, and Shoot modules will operate (i.e., action variables) based on the information input from the Perceive module (i.e., state

Table 2: Parameters that determine the level of cognitive noise simulated in each module and the reward setting of the agent

Cognitive $\boldsymbol{\theta}$	**Range**	**Description**	**Module**
θ_m	[0.0, 0.5][1]	Signal-dependent motor noise	Aim
θ_p	[0.0, 0.5][2]	Peripheral vision noise	Perceive
θ_s	[0.0, 0.5][3]	Speed perception noise	Perceive
θ_c	[0.0, 0.5][4]	Internal clock precision	Aim, Shoot
Reward $\boldsymbol{r}$	**Range**	**Description**	
r_h	[1, 64]	Maximum trial success (target hit) reward	
r_m	[1, 64]	Maximum trial failure (target miss) penalty	
λ_h	[5, 95]	Temporal decay rate of trial success reward	
λ_m	[5, 95]	Temporal decay rate of trial failure penalty	

[1](Lin and Tsai, 2015; Schmidt et al., 1979), [2](Hussain et al., 2015), [3](Stocker and Simoncelli, 2006; Wang and Li, 2007), [4](Rakitin et al., 1998; Park and Lee, 2020)

variables). Through deep RL, the policy function is determined to maximize the expected accumulated reward.

3.2.2. Model Parameters

The model has a total of eight free parameters (see Table 2). Four of them are cognitive parameters (i.e., θ) that determine the noise characteristics of each sub-module. The remaining four (i.e., r and λ) determine the shape of the reward function that the agent obtains as a result of interaction with the environment. Each cognitive parameter can be determined within a certain range, and each range was determined as broadly as possible by referring to existing literature. The appropriate range of reward parameters was determined empirically through trial and error.

3.2.3. Mathematical Notations

We use the symbols $\boldsymbol{p}$ and $\boldsymbol{v}$ to represent positions and velocities. As the subscripts in $\boldsymbol{p}$ and $\boldsymbol{v}$, the target, crosshair, mouse, gaze, and eye (head) are denoted as t, c, m, g, and e, respectively. The hat symbol ($\wedge$) implies that the value has been perceived (or estimated) by the simulated agent. For instance, $\boldsymbol{p}_t$ refers to the true target position, and $\hat{\boldsymbol{p}}_t$ refers to the estimated target position. The symbol t is used to represent time-related variables, k is used to represent model coefficients, and $\mathcal{M}$ is used to represent the agent's motor plans.

3.3. Model Agent Implementation

3.3.1. Decision-Making Framework

Before going deeper into the implementation of each submodule, we first explain the basic framework of the agent's decision-making process. Based on

recent studies (Bye and Neilson, 2008; Bye, 2009; Do et al., 2021; Park and Lee, 2020), we assume that human aim-and-shoot behavior is performed through *intermittent motor control* consisting of three distinct steps: (1) Sensory Analysis (SA), (2) Response Planning (RP), and (3) Response Execution (RE). In the SA step, the agent perceives (or estimates) from the task environment the information needed to build motor plans, such as the positions of the target and the crosshair. In the RP step, the agent builds appropriate motor plans, such as camera rotation, gaze movement, and click timing, based on the information obtained in the SA step. In the RE step, the agent actually executes the constructed motor plans and receives rewards from the environment as a result. The time required for each step is generally set to 0.1 s, referring to the human psychological refractory period (PRP) (Smith, 1967). A SA-RP-RE chunk is called a BUMP (Basic Unit of Motor Production). Before one BUMP is finished, a new BUMP is started, resulting in the motor plan being intermittently updated with a more recent one. Adjacent BUMPs overlap by two steps (see Figure 3).

The agent's decision-making is synchronized to the RP step and occurs once per BUMP. Under this framework, t=0 always means the moment the RP of the current BUMP begins. Similarly, the start time of the previous and next RP (or decision step) can be expressed as the moment t=$-t_p$ and t=t_p, respectively. The following sections describe the implementation of each submodule in detail based on this framework.

3.3.2. Aim Module

Let $\hat{\boldsymbol{p}}_c$, $\hat{\boldsymbol{v}}_c$, $\hat{\boldsymbol{p}}_t$, and $\hat{\boldsymbol{v}}_t$ represent estimates of the position and velocity of the crosshair and the position and velocity of the target, respectively. $\hat{\boldsymbol{p}}_c$ and $\hat{\boldsymbol{v}}_c$ are estimated to be the values at the moment t=t_p, and $\hat{\boldsymbol{p}}_t$ and $\hat{\boldsymbol{v}}_t$ are estimated to be the values at the moment t=t_p+t_h, that is, the agent is looking into the future (see Figure 3). Here, t_h is called the *prediction horizon* of motor control. These estimates are obtained in the SA step through the Perceive module, and the operation of the Aim module starts thereafter. The description of the Perceive module is deferred to the next section.

The operation goal of the Aim module is to construct a motor plan that starts execution at t=t_p and ends at t=t_p+t_h, and is expected to make the crosshair and target overlap and the relative speed become zero when execution is completed. There are infinite camera rotations that make this possible, but based on previous literature (Bye and Neilson, 2008; Bye, 2009; Sparrow and Newell, 1998; Jiang et al., 2002; Neilson and Neilson, 2005; Do et al., 2021), we assume that the agent pursues a camera rotation that satisfies the *minimum acceleration* criteria. In the BUMP model, such camera rotation can be obtained in closed-form via the OTG (Optimal Trajectory Generation) function:

$$\mathcal{M}^a \leftarrow \mathrm{OTG}(\hat{\boldsymbol{p}}_c, \hat{\boldsymbol{v}}_c, \hat{\boldsymbol{p}}_t, \hat{\boldsymbol{v}}_t) \tag{1}$$

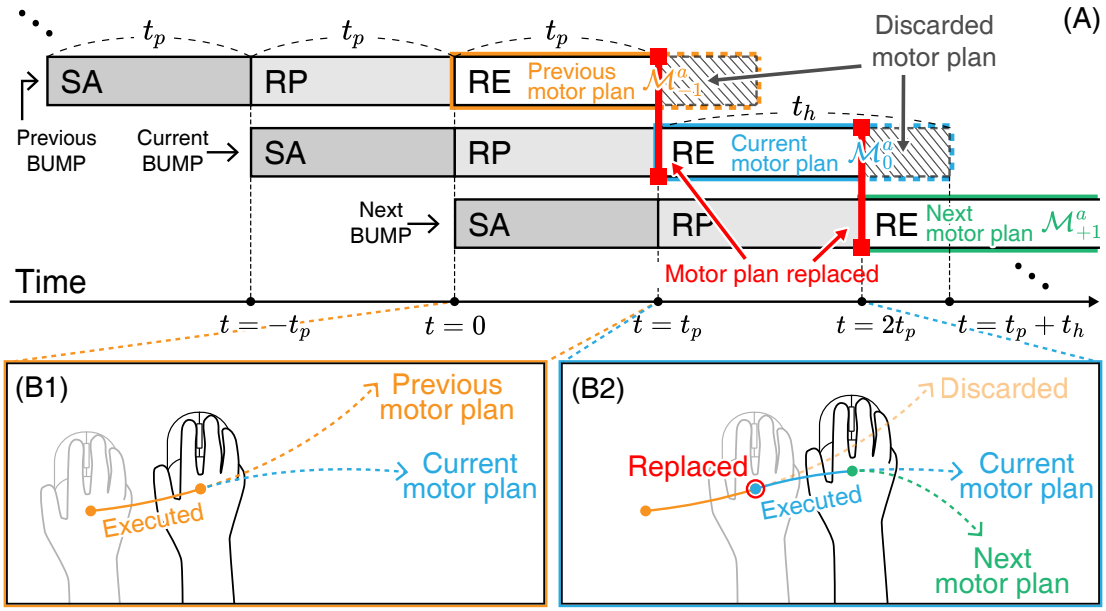

Figure 3: (A) The intermittent motor control process simulated by the Aim module, based on the BUMP model. (B1) The previous motor plan (in orange) is executed from $t = 0$ to $t = t_p$ and replaced with the current motor plan (in blue). (B2) The current motor plan (blue) will be replaced with the next motor plan at $t = 2t_p$ (in green).

Here $\mathcal{M}^a$ represents the (ideal) rotation plan of the view camera[6] from $t = t_p$ to $t = t_p + t_h$. The camera rotation plan generated in the previous and next BUMP are denoted as $\mathcal{M}^a_{-1}$ and $\mathcal{M}^a$+1, respectively. A detailed formula of the OTG function can be found in Appendix A.2.

The motor plan $\mathcal{M}^a$ is then executed during the subsequent RE step, and signal-dependent motor noise (Schmidt et al., 1979; Lin and Tsai, 2015) is added during this process. More specifically, noise is added separately in directions parallel and perpendicular to the camera angular velocity vector $\boldsymbol{\omega}^c$. Noise is sampled from a Gaussian distribution and added to the angular velocity vector, and its standard deviation is proportional to the magnitude of the instantaneous angular speed ($=\|\boldsymbol{\omega}^c\|$) (Lin and Tsai, 2015; Do et al., 2021). The standard deviations of the noise distribution in parallel ($\sigma_\parallel$) and perpendicular ($\sigma_\perp$) directions are expressed as:

$$\sigma_\parallel = \theta_m \|\boldsymbol{\omega}^c\| \,, \quad \sigma_\perp = 0.192 \cdot \theta_m \|\boldsymbol{\omega}^c\| \tag{2}$$

Here, θ_m is a parameter introduced to deal with inter-player variability. For

[6]Since we assume a constant gain function of the mouse, the camera rotation plan and the hand motor plan on the desk can be converted to each other by simply multiplying or dividing them by the agent's mouse sensitivity 1°/mm.

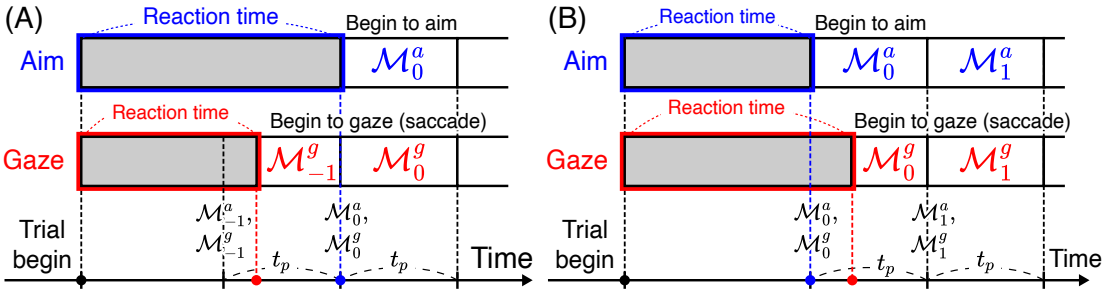

Figure 4: The reaction time for aim or gaze movement can be determined independently for each trial.

simplicity, rather than introducing separate parameters for each direction, the magnitude of noise in the perpendicular direction was assumed to be 0.192 times that of the noise in the parallel direction, as found in a previous study (Moon et al., 2022). Due to motor noise, camera rotation in the RE step is performed differently from the original motor plan $\mathcal{M}^a$.

Note that the camera will naturally remain still from the moment the target is given until the first motor plan is created, i.e., until the first pair of SA-RP steps are completed (i.e., for 0.2 seconds) (Bye and Neilson, 2008; Bye, 2009). This time interval represents the fundamental delay (or reaction time) of human hand (or mouse) movements. Considering that human reaction time may vary stochastically for each trial (Ratcliff, 1978), we implemented the Aim module so that the agent's mouse reaction time can be determined at a *desired* value rather than a fixed value of 0.2 seconds (see Figure 4).

3.3.3. Perceive Module

The role of the Perceive module is to estimate the future position and velocity of the target ($\boldsymbol{p}_t$, $\boldsymbol{v}_t$, at $t = t_p$) and crosshair ($\boldsymbol{p}_c$, $\boldsymbol{v}_c$, at $t = t_p + t_h$) based on the sensory signals given during the SA step (from $t = -t_p$ to $t = 0$) and then pass those estimates to the Aim module[7]. To achieve this goal, the module first estimates the positions and velocities of the target and crosshair at the moment $t = 0$. More specifically, the following two pieces of information are used in this process: (1) the position of the gaze on the screen ($\boldsymbol{p}_{g0}$) at $t = 0$, and (2) the position of the head in 3D space ($\boldsymbol{p}_{e0}$) at $t = 0$. Based on previous studies on peripheral vision noise (Hussain et al., 2015), the Perceive module then estimates the positions of the crosshair and target on the screen as follows:

$$\hat{\boldsymbol{p}}_{c0} \leftarrow \mathcal{N}(\mu = \boldsymbol{p}_{c0},\ \Sigma = \theta_p \cdot \epsilon_c \cdot I_2) \text{ and } \hat{\boldsymbol{p}}_{t0} \leftarrow \mathcal{N}(\mu = \boldsymbol{p}_{t0},\ \Sigma = \theta_p \cdot \epsilon_t \cdot I_2) \quad (3)$$

Here, ϵ_c and ϵ_t represent the eccentricity of the crosshair and target from the gaze position at $t = 0$, respectively, based on the head position (unit: visual

[7]For the sake of simplicity, we assume that the agent can perceive the size (or radius R_t) of the target perfectly without noise.

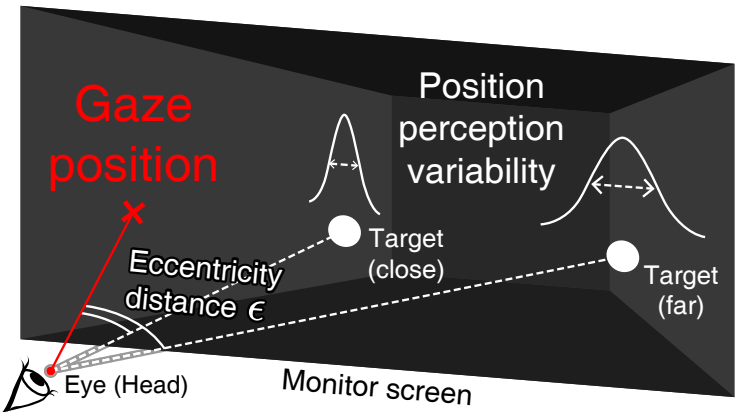

Figure 5: The variability of perceived target position is proportional to the visual distance between the gaze and the target.

angle °, see Figure 5). $\mathcal{N}$ represents a 2D normal distribution with mean μ and covariance Σ; and θ_p is a parameter introduced to deal with inter-player variability in peripheral vision noise.

In addition to the positional information at time $t = 0$, the velocity of the crosshair and target at $t = 0$ must be estimated to finally estimate $\boldsymbol{p}_t$, $\boldsymbol{v}_t$, $\boldsymbol{p}_c$, and $\boldsymbol{v}_c$. First, it is assumed that the agent knows prior information that the crosshair is fixed at the center of the screen and does not move. Therefore, it is assumed that the on-screen velocity of the crosshair perceived by the agent ($=\hat{\boldsymbol{v}}_c$) is always $\mathbf{0}$. Second, for the sake of simplicity, we assume that the agent always perfectly perceives the direction of movement of the target on the screen. However, it is assumed that perceptual noise is added to the *target speed* ($\|\boldsymbol{v}_{t0}\|$) perceived by the agent at time $t = 0$. More specifically, the perceived speed ($\hat{s}$) is sampled from the following log-normal distribution ($\mathcal{LN}$), as the Stocker's model (Jogan and Stocker, 2015) states:

$$\hat{s} = 0.3 \cdot (s' - 1) \;\text{ where }\; s' \leftarrow \mathcal{LN}\left(\mu = \ln\left(1 + \tfrac{s}{0.3}\right),\ \sigma = \theta_s\right) \tag{4}$$

Here, s and $\hat{s}$ represent the actual and estimated on-screen target speed at $t = 0$, respectively, but note that their unit is visual angular speed ($^\circ/s$) with respect to the head position; θ_s is a parameter introduced to reflect inter-player variability in speed perception. Finally, the velocity of the target at $t = 0$ perceived by the agent can be expressed as:

$$\hat{\boldsymbol{v}}_{t0} = (\hat{s}/s) \cdot \boldsymbol{v}_{t0} \;\text{ unit: } m/s \tag{5}$$

Due to the nature of the aim-and-shoot task, the velocity of the target on the screen expressed in Equation 5 is actually the sum of two velocity components: (1) the velocity of the target's own motion and (2) the velocity of target motion created by camera rotation, both at time $t = 0$. From the efference copy of the previous motor plan ($\mathcal{M}^a_{-1}$) currently being executed, the agent can estimate at

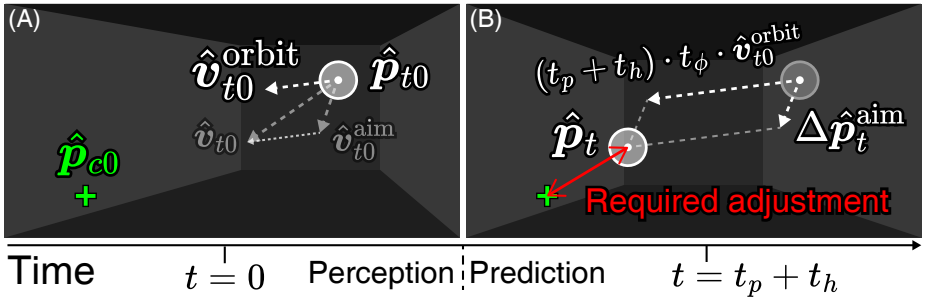

Figure 6: The process by which the agent estimates the future position of the target through the Perceive module

what velocity the target will move on the screen due to camera rotation at time $t = 0$. If the estimated vector is $\hat{\boldsymbol{v}}_t^{\text{aim}}$, by subtracting $\hat{\boldsymbol{v}}_t^{\text{aim}}$ from the estimated on-screen target velocity $\hat{\boldsymbol{v}}_{t0}$, the agent can infer the velocity of the target's intrinsic motion ($\boldsymbol{v}_t^{\text{orbit}}$) as follows:

$$\hat{\boldsymbol{v}}_{t0}^{\text{orbit}} = \hat{\boldsymbol{v}}_{t0} - \hat{\boldsymbol{v}}_{t0}^{\text{aim}} \tag{6}$$

Note here that $\hat{\boldsymbol{v}}_{t0}^{\text{aim}}$ is obtained by the agent under the assumption that the motor plan executes perfectly without noise.

Finally, based on prior information that the target moves at a constant velocity, the agent can estimate the target's future position and velocity at time $t = t_p + t_h$ as follows:

$$\hat{\boldsymbol{p}}_t = \hat{\boldsymbol{p}}_{t0} + \Delta\hat{\boldsymbol{p}}_t^{\text{aim}} + (t_p + t_h) \cdot t_\phi \cdot \hat{\boldsymbol{v}}_{t0}^{\text{orbit}} \quad \text{and} \quad \hat{\boldsymbol{v}}_t = \hat{\boldsymbol{v}}_{t0}^{\text{orbit}} \tag{7}$$

Here, $\hat{\boldsymbol{p}}_t^{\text{aim}}$ is the displacement of the target on the screen that is *expected* to be created by the camera rotation resulting from the ideal execution of the previous motor plan $\mathcal{M}_{-1}^{a}$ from $t = 0$ to $t = t_p$ (i.e., efference copy). On the other hand, t_ϕ refers to the noise added when the agent's internal clock encodes the time interval $(t_p{+}t_h)$ and is sampled at every decision step from a normal distribution as follows:

$$t_\phi \leftarrow \mathcal{N}\left(\mu = 1,\ \sigma = \theta_c\right) \tag{8}$$

Due to the above internal clock noise, the precision of the target position estimated by the agent becomes lower when the estimate is made for the further future. A more detailed description of the internal clock encoding mechanism will be provided in Section 3.3.5, where the implementation of the Shoot module is described. Figure 6 illustrates the mechanism handled by the Perceive module.

3.3.4. Gaze Module

The Gaze module simulates the agent's gaze control process. We assume that gaze control is *synchronized* with the cycle of the Aim module; at the start of each RP (i.e., time $t = 0$), the agent determines the next position on the

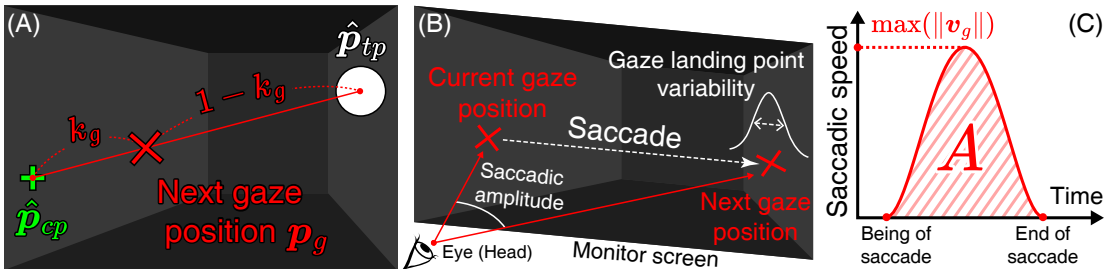

Figure 7: The process by which the Gaze module simulates the agent's gaze movement

screen to which to move the gaze. However, unlike the camera rotation motor plan, the prediction horizon of the gaze control plan is fixed at $t = t_p$. To simplify the model, the potential positions to which the agent can decide to move its gaze are limited to positions on the line segment between the target and the crosshair at time $t = t_p$ (see Figure 7A). This assumption can be made because moving the gaze away from both the target and the crosshair is not a rational decision in terms of minimizing peripheral noise. More specifically, this gaze control process is expressed mathematically as:

$$\boldsymbol{p}_g = (1 - k_g) \cdot \hat{\boldsymbol{p}}_{cp} + k_g \cdot \hat{\boldsymbol{p}}_{tp} \tag{9}$$

Here, $\boldsymbol{p}_g$, $\hat{\boldsymbol{p}}_{cp}$, and $\hat{\boldsymbol{p}}_{tp}$ represent the target gaze position, perceived crosshair position, and perceived target position at time $t{=}t_p$, respectively. The k_g is a variable between 0 and 1 introduced to determine the operation of the Gaze module. When k_g is 0, the module plans to move the gaze to the crosshair position at time $t = t_p$, and when k_g is 1, it plans to move the gaze to the target position at time $t = t_p$ (Figure 7A).

Motor noise is also involved in gaze control (Kowler, 1990), and the position $\hat{\boldsymbol{p}}_g$ where the gaze will *actually* arrive is sampled from a two-dimensional Gaussian distribution with a standard deviation proportional to the saccadic amplitude (A in visual angle °) as follows:

$$\hat{\boldsymbol{p}}_g \leftarrow \mathcal{N}(\mu = \boldsymbol{p}_g,\ \Sigma = 0.1 \cdot A \cdot \boldsymbol{I}) \tag{10}$$

The gaze moves straight on the screen from the current gaze position to $\hat{\boldsymbol{p}}_g$ (Figure 7B). In this process, the peak speed of gaze movement ($\max(|\boldsymbol{v}_g|)$) is determined based on the human main sequence model (Bahill et al., 1975) as follows:

$$\max(\|\boldsymbol{v}_g\|) = a + b \cdot \max(0, A - 1) \quad (\text{unit: } °/\text{s}) \tag{11}$$

Here, a and b are free parameters of the model and are determined empirically in this study. Among the infinite number of gaze trajectories that satisfy the required peak speed ($\max(\|\boldsymbol{v}_g\|)$) and saccadic amplitude (A), the actual gaze movement $\mathcal{M}^g$ is determined as the one that satisfies the minimum jerk assumption (see Figure 7C). In this process, gaze may arrive at $\hat{\boldsymbol{p}}_g$ earlier or later than time $t{=}t_p$. If it arrives early, the gaze remains stationary until $t{=}t_p$. If it arrives

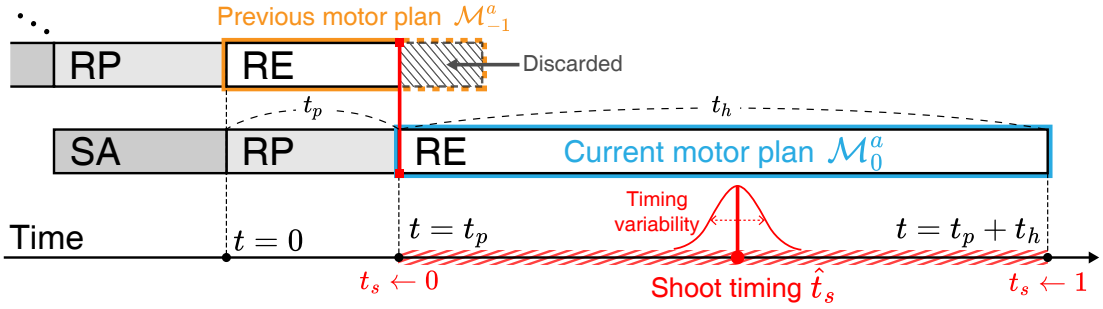

Figure 8: The timing of the shot is planned in the Shoot module, but the actual shot timing is perturbed by internal clock noise.

late, it is replaced and executed with a new gaze plan created at time t=t_p. A detailed formula of the minimum jerk trajectory is described in Appendix A.3.

Similar to the Aim module, the Gaze module is also implemented so that the gaze begins to move only after a certain reaction time elapses after the target is first given. With reference to previous studies on human aimed movement (Boucher et al., 2007), the reaction times of gaze and hands were implemented so that they could be determined independently.

3.3.5. Shoot Module

The Shoot module simulates the process of the agent deciding whether to click a mouse button (whether to shoot or not) and, if so, when to click. The decision is made once every time a new motor plan $\mathcal{M}^a$ is created in the Aim module (i.e., at every t=0). The binary variable representing the decision to click or not to click is k_s; When k_s is 1, it indicates that the agent has decided to click, and when k_s is 0, it indicates that the agent has decided not to click. If the agent decides to click, the value of the variable t_s, which indicates the planned timing of the click, is also determined. The agent determines the value of t_s between 0 and 1; If t_s is 0, it means that the agent decided to click at the start of the motor plan $\mathcal{M}^a$ (i.s., at $t = t_p$), and if t_s is 1, it means that the agent decided to click at the end of the motor plan $\mathcal{M}^a$ (i.e., at $t = t_p + t_h$). If the agent has decided to click within a motor plan, the Aim module stops creating subsequent new motor plans. The process is illustrated in Figure 8.

According to previous studies (Rakitin et al., 1998; Lee and Oulasvirta, 2016; Lee et al., 2018; Lee, 2022; Lee et al., 2024a), the human internal clock participates in determining click timing as a kind of countdown timer, and the precision of the internal clock decreases as it counts down to a more distant future timing. This *scalar property* of the internal clock can be expressed mathematically as follows, using t_ϕ, a Gaussian random variable with standard deviation θ_c and mean 1 (see Equation 8):

$$\hat{t}_s = t_\phi \cdot (t_p + t_s \cdot t_h) \tag{12}$$

Here, $\hat{t}_s$ represents the timing at which shooting actually occurs, including the

influence of internal clock noise. Note that t_p is added to the right-hand side of the equation because the execution of the motor plan $\mathcal{M}^a$ begins at $t = t_p$.

3.4. Optimizing Aim-and-Shoot Control Policy of The Model Agent

Our model agent, which integrates the four sub-modules, can simulate a wide range of aim-and-shoot behavior by determining the values of 4 action variables for each RP step. In this study, we specifically follow the computational rationality framework and assume that agents determine action variables in a way that maximizes expected accumulated rewards. The function that determines the action variable for a given aim-and-shoot situation (i.e., environmental states) is called a policy function (π), and it can be optimized through deep RL.

3.4.1. Problem Formulation

To optimize the policy function, we formalize the agent's decision-making process as a Markov decision-making process (MDP). Decisions are made at the start of each RP, that is, every 0.1 seconds. When state variables are determined at each decision step, the policy function returns corresponding (optimal) action variables. In our formulation, there are 6 state variables and 4 action variables (see Table 3). The specific meaning of each variable can be found in the previous sections on the implementation of sub-modules.

Table 3: State and action variables of the control policy

	Symbol	Definition (Dimension)
State $\boldsymbol{s}$	$\hat{\boldsymbol{p}}_t$	Estimated target position at $t = t_p$ (2D)
	$\hat{\boldsymbol{v}}_t^{\text{orbit}}$	Estimated target intrinsic velocity at $t = t_p$ (2D)
	R_t	Target radius (1D)
	$\hat{\boldsymbol{v}}_m$	Estimated mouse velocity at $t = t_p$ (2D)
	$\boldsymbol{p}_g$	Gaze position on the monitor at $t = 0$ (2D)
	$\boldsymbol{p}_e$	Head (eye) position (3D)
Action $\boldsymbol{a}$	t_h	Prediction horizon in the Aim module (1D)
	k_g	Gaze damping coefficient in the Gaze module (1D)
	k_s	Binary shoot decision in the Shoot module (1D)
	t_s	Shoot timing coefficient in the Shoot module (1D)

As a result of decision-making for each step, the agent receives a specific reward. The function of the reward r obtained for each step can be expressed as follows:

$$r = \begin{cases} 0 & \text{if } k_s = 0 \\ r_h \cdot (1 - \lambda_h/100)^T & \text{if } k_s = 1 \text{ and target hit} \\ -r_m \cdot (1 - \lambda_m/100)^T & \text{if } k_s = 1 \text{ and target missed} \end{cases} \tag{13}$$

Here, r_h (or r_m) refers to the reward (or penalty) that can be obtained when the agent succeeds (or fails) in shooting the target *immediately* after the start of the trial. Under the assumption that humans consider the reward to be obtained *per unit time* (Banovic et al., 2013; Munichor et al., 2006; Klapproth, 2008; Ashby and Gonzalez, 2017) rather than its absolute accumulated value, the reward or penalty that the agent obtains as a result of shooting decreases as the trial completion time (T) increases, with each decay rate being λ_h and λ_m. Intuitively, $\lambda_h = 10$ indicates that the reward for the target hit is reduced by 10% per second. From this we can make the agent pursue quick success while avoiding failure without sufficient exploration, resulting in more realistic behavior. If the agent took too long to aim (i.e., longer than 3 seconds), the target went out of the agent's view, or the elevation angle of the target became too high (i.e., larger than 83°), the agent was considered to have failed to acquire the target, received a huge penalty ($r = -100$), and the trial ended.

3.4.2. Deep RL

We trained the agent through Soft Actor-Critic (SAC) (Haarnoja et al., 2018). Our aim is that the learned policy works optimally with different sets of cognitive $(\theta_m, \theta_p, \theta_s, \theta_c)$ and reward parameters $(r_h, r_m, \lambda_h, \lambda_m)$. Because different parameter values lead to other state transitions on MDP, a conventional policy that is a function of only the state usually cannot solve this problem. Instead, we adopted the recent multi-task RL approach (Moon et al., 2022). We first set the range of $\boldsymbol{\theta}$ and $\boldsymbol{r}$ as in Table 2. The range of $\boldsymbol{\theta}$ includes the known distribution of the parameter values (Lee and Oulasvirta, 2016; Lee et al., 2019, 2018; Park and Lee, 2020; Lee et al., 2021). The policy was then trained on episodes with varying $(\boldsymbol{\theta}, \boldsymbol{r})$ sampled on the ranges. The sampled $(\boldsymbol{\theta}, \boldsymbol{r})$ was provided as auxiliary inputs to the policy network along with the state $\boldsymbol{s}$ so that the output action variables can be determined considering both the given characteristics of the agent $(\boldsymbol{\theta}, \boldsymbol{r})$ and current task state $\boldsymbol{s}$: $\boldsymbol{a} = \pi(\boldsymbol{s}, \boldsymbol{\theta}, \boldsymbol{r})$.

We built the actor and critic networks for the policy, each with an input layer (12 units, same as the state dimension), 3 hidden layers (512 units), and an output layer (4 units, same as the action dimension). While the state $\boldsymbol{s}$ was fed into each network as primary inputs, the model parameters ($\boldsymbol{\theta}$, $\boldsymbol{r}$) were provided by concatenating them to the input layer and hidden layers (Moon et al., 2022). We adopted an Adam optimizer (Diederik and Ba, 2015) with a learning rate of 0.00005 for training. The batch size was 2048, the discount factor was 0.9, and the entropy regularization coefficient was initialized at 0.999 and converged to 0.2. We trained the policy for 20 million episodes, and it took approximately 9 hours with a PC equipped with an AMD Ryzen 9 5950x CPU (16 cores), an NVIDIA RTX 3080 GPU, and 64GB of RAM. Figure 9 shows the changes in success rate and reward during the training process.

3.5. The Baseline Model

Our model is basically an expanded and advanced version of the 2021 point-and-click model (Do et al., 2021) (see Table 1). To rigorously assess how significant such expansions were, we also build a strong baseline model. The baseline

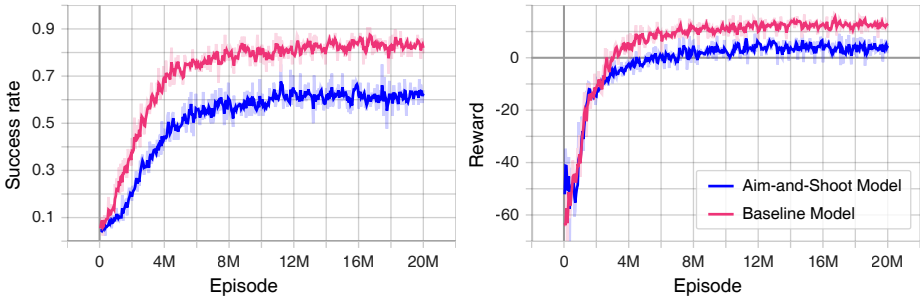

Figure 9: The success rate (left) and reward (right) curve of the aim-and-shoot model and the baseline model during the training

model can be considered a simple conversion of the point-and-click model into the 3D aim-and-shoot interaction scenario without significantly modifying existing modules. More specifically, the baseline model has the following differences compared to our model: (1) there is no Gaze module, and the positions of the target and crosshair are always perceived accurately, (2) the mouse reaction time is fixed at 0.2 seconds (i.e., the first SA-RP interval), (3) the target's future position is extrapolated by directly perceiving the target's own speed (efference copy not considered), (4) the Shoot module is implemented based on the intermittent click planning (ICP) model (Park and Lee, 2020)[8], so the agent does not determine the click timing as a separate action variable.

The parameters of the baseline model are identical to our model except that θ_p is not included (Table 2). The same reward function and deep RL process are applied (see Figure 9 for the reward and success rate during the training). Due to fewer perceptual noises, the baseline model shows an overall higher success rate and reward than our model.

4. The Aim-and-Shoot Behavior Dataset

To validate the model, we first build a dataset of the aim-and-shoot behavior of real FPS players. Both professional and amateur players were included in the dataset to cover a wide range of cognitive and motivational characteristics. In this section, the design and results of the data acquisition user study are described in detail.

4.1. Method

4.1.1. Participants

Twenty participants were recruited. Ten participants were active FPS professionals on `Valorant` (Riot Games, 2020) and `PUBG: BATTLEGROUNDS` (PUBG

[8]With reference to previous studies (Do et al., 2021; Moon et al., 2022), the parameters of the ICP model were determined as follows: c_μ (0.185), ν (19.93), and δ (0.399).

Table 4: Participants in the user study

Group	No.	Age	Gender	Exp. on FPS (y)	Primary game	Habitual sensi. (°/mm)	Handedness	Mouse device	Rank (%)
Professional	1	18	M	9	PUBG[1]	0.63	R	G Pro X Superlight	<0.1
	2	18	M	10	VAL[2]	0.83	R	G Pro X Superlight	<0.1
	3	18	M	7	PUBG	0.43	R	Razor Viper Ultimate	<0.1
	4	17	M	7	VAL	0.64	R	G Pro X Superlight	<0.1
	5	15	M	3	PUBG	1.14	R	BenQ ZOWIE FK2-B	<0.1
	6	17	M	5	VAL	1.07	R	G Pro Wireless	<0.1
	7	18	M	4	VAL	0.49	R	BenQ ZOWIE EC2-B	<0.1
	8	16	M	5	VAL	0.50	R	Logitech G102	<0.1
	9	15	M	6	VAL	2.20	R	Logitech G502 Hero	<0.1
	10	17	M	5	VAL	0.36	R	Logitech G703	<0.1
Amateur	1	25	M	3	DBD[3]	1.50	R	Logitech G304	20∼30
	2	26	F	0	-	1.50	R	G Pro Wireless*	-
	3	23	M	11	PUBG	3.30	R	Logitech G102	-
	4	24	M	14	OW[4]	1.08	L	G Pro Wireless	0.1∼1
	5	24	M	10	OW	1.25	R	G Pro X Superlight	20∼30
	6	24	M	11	OW	0.98	R	G Pro Wireless*	10∼20
	7	28	M	0	-	1.50	R	G Pro Wireless*	-
	8	23	F	3	PUBG	1.50	R	G Pro Wireless*	-
	9	19	F	1	OW	1.50	R	G Pro Wireless*	-
	10	25	M	10	OW	1.30	R	G Pro Wireless*	0.1∼1

[1∼4] PUBG: BATTLEGROUNDS (PUBG Corporation, 2017), Valorant (Riot Games, 2020), Dead by Daylight (Behaviour Interactive, 2016), Overwatch (Blizzard Entertainment, 2016)
* Provided by the experimenter.

Corporation, 2017). All ranked in the top ¡0.1% tier at their primary game. The other 10 participants were amateur players, where two ranked in the top 2% tier, four ranked in the top 10% to 30%, and the rest had not played ranked games. We put participant details in Table 4.

4.1.2. Task

Participants performed an aim-and-shoot task in a typical desktop environment with the same scenario (see Section 3.1) given to our model agent. Each trial begins when participants successfully aim and shoot a red reference target created at a location where both azimuth and elevation angle are 0. When a trial begins, a grayscale-colored main target is created with a specific location, radius, and speed. If a shooting event (left click) occurs when the crosshair is located within the main target, the trial is considered successful. Regardless of whether target acquisition is successful or not, when a shooting event occurs, the trial ends, and another reference target for the next trial is created at the

same 3D location. Participants were instructed to perform this task as quickly and as accurately as possible. The scenes of the task are displayed in Figure 10.

4.1.3. Design

The user followed a 2×2×2×2×2 mixed design. The independent variables and levels were:

- Radius: Small (4.5 mm) or Large (12 mm)
- Speed: Stationary (0 cm/s) or Moving (15 cm/s)
- Color: White (255, 255, 255) or Gray (30, 30, 30)
- Sensitivity: Default (1 °/mm) or Habitual
- Group: Professional or Amateur

Here, Radius, Speed, and Color determine the characteristics of the main target. Color was introduced as an independent variable because the background contrast of the target may affect participants' speed perception noise (Stocker and Simoncelli, 2006). Multiplying Sensitivity by the physical displacement of the mouse (unit: mm) determines the amount of rotation of the first-person view camera (unit: °). The Habitual condition reproduced the familiar sensitivity that each participant uses on a daily basis when playing their primary FPS game (Table 4). Except for Group, all others were within-subject factors.

The total number of unique conditions experienced per participant was 18. The following behavioral data were logged with timestamps: gaze position on the monitor, head position in the physical world, click events, view camera orientation and target position in 3D space. Gaze data was collected at 150 Hz, and the remaining data at 240 Hz.

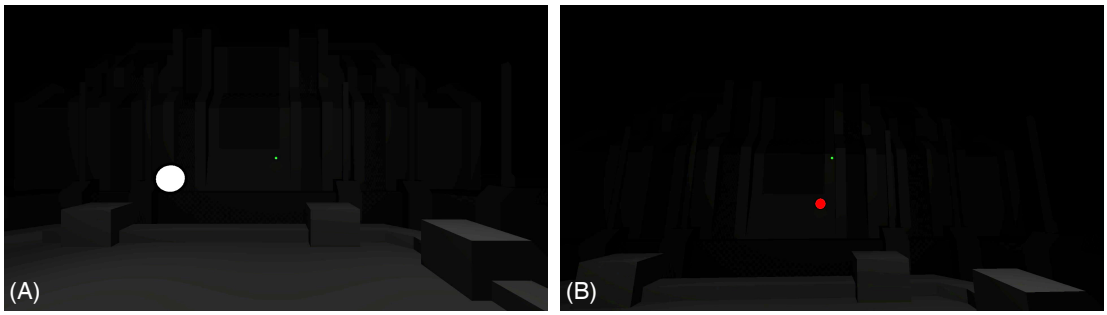

Figure 10: User study screen captures: (A) The green dot is the crosshair. The white circle is the main target to be shot. (B) The red sphere is the reference target. Please note that we increased the exposure to make overall scenes more visible.

4.1.4. Procedure

As each participant arrived, we adjusted the chair and monitor heights to align the participant's eye level with the monitor center. The monitor distance from the eyes was initially set to approximately 58 cm to 63 cm, and participants were asked not to get too far or close to the monitor. The participant signed the

consent form, and then we gave the participant an overview of the experiment. The eye tracker was calibrated after that.

Participants performed 2 blocks for each unique Radius-Speed-Color-Sensitivity combination, containing 60 aim-and-shoot trials per block. As a result, each participant completed 2,160 trials. The participants completed all blocks in the first Sensitivity condition and then moved on to the second Sensitivity condition. Both the orders of Sensitivity conditions and the blocks within a sensitivity condition were randomized across participants. Between blocks, we performed a short calibration verification session of the eye tracker, where participants were asked to gaze at a dot that appears on the 11 fixed positions of the screen. After the verification session, the participants could rest until the next block started. The participants filled out a questionnaire after finishing all trials. The entire experiment took about 80 minutes per participant. Considering the burden of recruiting participants for each group, the compensation was 90 USD for professionals and 15 USD for amateurs. IRB approved our user study protocol.

4.1.5. Apparatus

The task environment was implemented using `FPSci`, an open-source FPS experimentation tool (Spjut et al., 2022). The user study was hosted on the desktop with AMD Ryzen 9 5950x CPU, RTX 3080 GPU, and 32GB RAM and presented on a BenQ Zowie XL2540k gaming monitor with a 240 Hz refresh rate. The eye tracker was Gazepoint GP3 HD Eye-Tracker, operating at a sampling rate of 150 Hz. Participants were encouraged to bring their mouse and use it (see Table 4), but those who did not were provided with a Logitech G Pro wireless mouse. The polling rate of all mice was set to 1,000 Hz.

4.2. Result

4.2.1. Post-processing

We synchronized aim behavior data and gaze data by considering the latency estimated from video captured by a high-speed camera. We then oversampled the gaze data to 240 Hz through cubic spline interpolation (see Appendix B.1 for details). We removed outliers by considering trial completion time and shot error[9]; The 1.5 IQR (Inter-Quartile Range) method was applied. Additionally, trials in which more than 60 % of the measured gaze data were invalid (i.e., due to eye closure, device error, gaze gone out-of-screen) or in which the gaze was more than 3.5 cm away from the crosshair at the moment of shooting the reference target were considered outliers and were removed. As a result, a total of 3,265 trials were removed (7.6 %).

4.2.2. Descriptive Statistics

By analyzing the final dataset, we checked whether there were significant behavioral and performance differences between professionals and amateurs. To

[9]The distance between the target and the crosshair at the moment of the shot

Table 5: Main effect of independent variables on each dependent variable: significant differences ($p < 0.05$) are indicated in green. The values in parentheses are standard deviations.

Condition	**Level**	**TCT** (ms)	**ACC** (%)	**SCD** (°)	**GRT** (ms)	**MRT** (ms)
Player	Professional	515 (86.3)	83.0 (20.3)	5.24 (1.51)	132 (23.0)	161 (9.9)
Group	Amateur	568 (97.1)	75.6 (24.9)	4.37 (1.42)	170 (35.8)	162 (9.0)
Target	Large	480 (59.7)	91.1 (10.1)	4.66 (1.44)	149 (31.0)	160 (9.1)
Radius	Small	603 (84.5)	67.6 (26.2)	4.95 (1.61)	153 (39.7)	163 (9.5)
Target	Stationary	530 (89.4)	93.4 (7.0)	4.56 (1.55)	150 (30.4)	161 (9.5)
Speed	Moving	553 (100.2)	64.8 (24.3)	5.05 (1.47)	152 (40.1)	161 (9.4)
Target	White	533 (91.0)	79.2 (22.6)	4.78 (1.53)	145 (33.6)	158 (8.9)
Color	Gray	550 (99.6)	79.5 (23.5)	4.84 (1.54)	157 (36.6)	165 (8.6)
Mouse	Default	543 (93.8)	78.8 (23.8)	4.85 (1.42)	150 (36.3)	161 (9.7)
Sensi.	Habitual	540 (97.6)	79.9 (22.3)	4.76 (1.63)	152 (34.9)	161 (9.2)

remove the learning effect, we discarded trials in the former block among two blocks in each unique task condition. Accordingly, we employed 20,045 trials (46.4 %) in the analysis. The following five metrics were calculated for each participant and each unique task condition:

- Mean trial completion time (TCT): the average time from trial start to a shot
- Accuracy (ACC): the rate of successful shots
- Mean saccadic deviation (SCD): the average visual angle from the initial gaze position to the furthest gaze deviated position
- Mean gaze reaction time (GRT): the average time from the start of a trial until a significant gaze shift is observed
- Mean mouse reaction time (MRT): the average time from the start of a trial until a significant mouse displacement is observed

Here, GRT was calculated more specifically as the time from the start of a trial to the start of the first saccade. Saccades were parsed from gaze trajectories using the Python `pymovement`[10] package (Krakowczyk et al., 2023), and saccades with an amplitude of less than 1 ° or that ended before 100 ms were ignored in this process. In MRT calculations, the time of the mouse's first movement was determined based on acceleration thresholding[11].

To confirm statistical differences in performance metrics between Group levels, mixed ANOVA with an α level of 0.05 was performed. TCT of Professionals (M=515 ms, SD=86) was significantly shorter than that of Amateurs (M=568 ms, SD=97): $F_{1,18}$=5.131, p=0.036, η_p^2=0.222. ACC was significantly higher in Professionals (M=83.0%, SD=20.3) than in Amateurs (M=75.6%, SD=24.9):

[10] We used the function `pymovement.events.microsaccades` with parameters set as `threshold_factor=3.5`, `minimum_duration=6`, and `threshold='engbert2015'`.

[11] When S is the mouse speed at t=0.1 (s), and $S_{\max}$ is the maximum mouse speed observed when $t = T$ (s), the acceleration threshold is $(S_{\max} - S)/(T - 0.1)$.

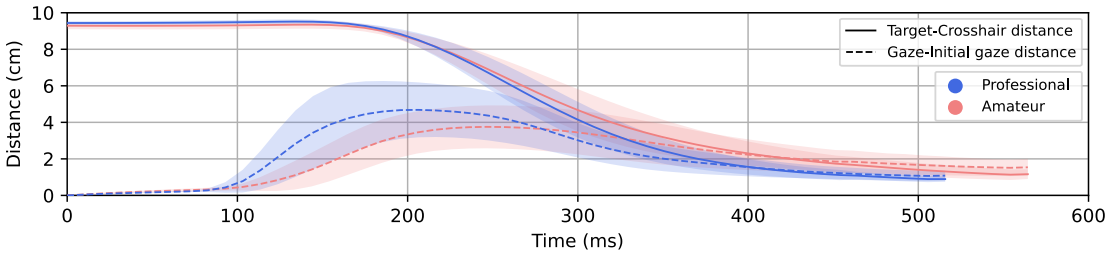

Figure 11: Changes in the distance from the crosshair to the target and the distance between the initial gaze and the current gaze over time: the shaded area represents the standard deviation. This is a visualization of data obtained from the user study, not a simulation.

$F_{1,18}$=5.445, p=0.031, η_p^2=0.232. Interestingly, Professionals showed higher ACC than Amateurs despite a shorter TCT. There was no significant difference in SCD between Professionals (M=5.24°, SD=1.51) and Amateurs (M=4.37°, SD=1.42): $F_{1,18}$=1.995, p=0.175, η_p^2=0.100. GRT was significantly shorter on Professionals (M=132 ms, SD=23.0) than Amateur (M=170 ms, SD=35.8): $F_{1,18}$=11.096, p=0.004, η_p^2=0.381. No significant difference was found in MRT between Professionals (M=161 ms, SD=9.9) and Amateurs (M=162 ms, SD=9.0): $F_{1,18}$=0.062, p=0.807, η_p^2=0.003. We leave the summary of the overall result in Table 5 (see Appendix B.2 for the detailed statistical results). In addition, Figure 11 shows for each Group how the target and gaze moved over time on average on the screen.

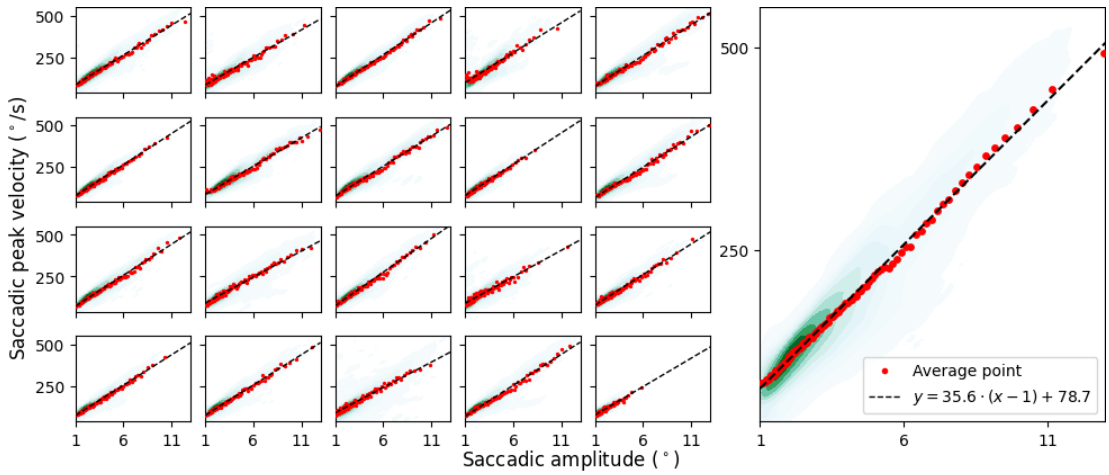

Figure 12: Fitting results of the main sequence model: (left) for each participant, (right) for all participants combined

4.2.3. Main Sequence Modeling

The Gaze module of our agent model replicates the dynamics of human gaze control based on the main sequence model (see Equation 11). To determine the values of the model parameters (i.e., a and b), we fit the model to the dataset obtained from the user study. As a result of model fitting for each Group and each participant (see Figure 12), we found no significant difference in the intercept (a) between Professionals (M=79.4, SD=8.6) and Amateurs (M=78.4, SD=8.1), $F_{1,18}$=0.072, p=0.792, η_p^2=0.004, or in the slope (b) between Professionals (M=36.0, SD=1.6) and Amateurs (M=35.1, SD=1.6), $F_{1,18}$=0.696, p=0.415, η_p^2=0.037. Therefore, we fitted the model on the entire dataset without distinguishing Groups, determined the main sequence equation as follows, and loaded it on the model agent: $\max(\|\boldsymbol{v}_g\|) = 78.7 + 35.6 \cdot \max(0, A - 1)$.

5. The Aim-and-Shoot Model Validation

In this section, we evaluate how well the implemented aim-and-shoot model can fit the behavior of real players. The aim-and-shoot model and baseline model are fitted to the dataset constructed from the user study, and the fitting performance of the two is rigorously compared in various aspects. Considering the general complexity of CR models, model fitting was performed by applying the latest amortized inference technique (Moon et al., 2023).

5.1. Amortized Inference Engines

The goal of model fitting is to find optimal model parameters that make the simulation of the CR model as similar as possible to the given actual human behavior. Iteratively searching the parameter space to achieve this (Moon et al., 2022) take extremely long time (i.e., a few days) in CR models with multiple parameters (e.g., 8 for our model and 7 for the baseline). The recently proposed amortized inference technique (Moon et al., 2023) resolves this issue, guaranteeing high fitting performance as well as fast fitting time to tens of milliseconds. More specifically, we implemented an amortized inference engine for each of our aim-and-shoot model and baseline model. By inputting the observed aim-and-shoot behavior into each engine, optimal model parameters can be derived through only a single forward pass of the neural network.

5.1.1. Engine Architecture

The inference engines were implemented using the neural network structure presented in the previous study (Moon et al., 2023). To ensure a fair comparison, the same structure was used for both our model and the baseline (see Figure 13). The structure takes two different types of data as input: (1) summary statistics and (2) behavioral trajectories. Summary statistics include trial completion time, normalize shot error[12], saccadic deviation (N/A for baseline), mouse reac-

[12]The distance between the target and the crosshair at the shooting moment, divided by the target radius. A value smaller than 1 indicates a target hit.

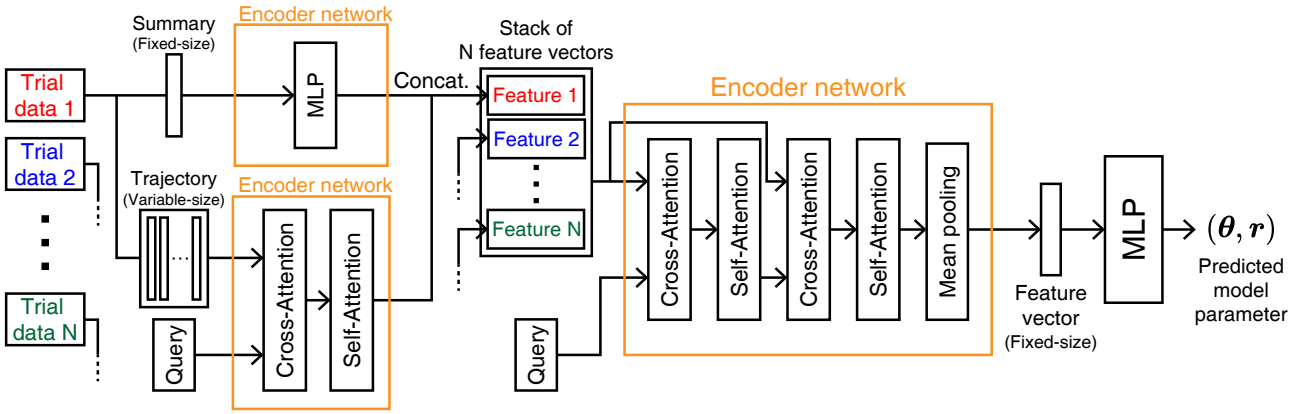

Figure 13: The network architecture of the model parameter inference engine

tion time, gaze reaction time (N/A for baseline), target initial position, target speed and size, and initial head position (N/A for baseline). Behavioral trajectories include time series of on-screen target position and camera orientation, all with timestamps.

The summary statistics are input into a multi-layer perceptron (MLP) with two hidden layers of 128 units each, producing an output of size 64. The behavioral trajectories are processed through the Transformer-based architecture known as the *Perceiver*, introduced by Jaegle et al. (Jaegle et al., 2021). This network utilizes a combination of repeated cross-attention and self-attention mechanisms. Employing a query vector of size (4×32) effectively transforms the trajectories into a condensed vector of size 4. Consequently, each trial data yields a 68-D vector. These 68-D vectors, stacked from multiple trials, are further processed by another Perceiver network, which, using a query vector of size (4×32), distills the information into a 32-D vector of latent features. Finally, we used an MLP with two hidden layers of 128 units each, outputting the predicted model parameter values. ReLU activations were used in the MLPs, and GELU activations were utilized in the Perceiver networks.

5.1.2. Engine Training

The parameters of the neural networks included in each engine are trained based on large amounts of synthetic data. Here, synthetic data is generated through the simulation of each model. More specifically, we first sampled a total of 21M unique sets of model parameters for each model (see Table 2 for sampling range). Then, for each unique parameter set, 64 trials were simulated following the aim-and-shoot task scenario outlined in Section 4.1.2. Finally, supervised learning was performed based on the obtained parameter-trial pair datasets (1.3 billion pairs for each model). For the training, we utilized the Adam optimizer, incorporating gradient clipping at a maximum of 0.5. For the learning rate, we adopted Cosine Annealing with Warm Restarts (Glöckler et al., 2017), setting the maximum learning rate to 0.0001 and employing a learning

rate decay factor of 0.9. The training took 200 thousand steps with a batch size of 64. Dataset synthesis took approximately 120 hours, and the training took 45 hours with a PC (AMD Ryzen 9 5950x CPU, an NVIDIA RTX 3080 GPU, 64GB of RAM). We detailed the inference performances of trained engines in Appendix C.

5.2. Model Fitting

5.2.1. Methods

Using amortized inference engines, we fit our model and the baseline model to the dataset obtained from the user study. The fitting of each model for each participant was performed four times, dividing the dataset into the following four conditions: (1) White-Default, (2) White-Habitual, (3) Gray-Default, and (4) Gray-Habitual. The reason data was not aggregated for each participant was to address the impact that color and sensitivity may have on visual or motor noise (Boudaoud et al., 2022; Hussain et al., 2015; Stocker and Simoncelli, 2006; Shen et al., 2015). As a result, a total of 4 sets of 8 or 7 parameters are obtained for each participant for our model and the baseline model, respectively. For gaze reaction time (only in our model) and mouse reaction time, the actual measured values for each trial were utilized.

To assess how similar the output of the fitted model is to the participants' actual behavior, we analyze:

- Mean absolute error (MAE) of TCT, ACC, and SCD averaged for each of the 16 unique task conditions
- Correlation (R^2) between actual data and model output in TCT, ACC, and SCD, binned by participant (R^2_{Inter}) or binned by task condition (R^2_{Intra})
- Kullback-Liebler divergence (KLD) between each TCT, ACC, and SCD's distribution and each corresponding model output distribution, for each participant
- The extent to which the gaze and camera trajectories output by the model are similar to those in the dataset, for each participant and each trial, via the dynamic time warping (DTW) algorithm (Vintsyuk, 1968; Berndt and Clifford, 1994)
- How well the simulated pointing performance for stationary targets conforms to Fitts' law[13] (Fitts, 1954; MacKenzie, 1989, 1992) (R^2_{Fitts})

Note that analyzes related to SCD and gaze trajectories were not performed on the baseline model because it does not address gaze movements. In addition to fitting to the entire dataset, two-fold cross-validation was also performed to check for overfitting issues.

[13] $E[\text{TCT}] = a + b \cdot \log_2(D/W + 1)$, where D is the initial distance from the crosshair to the target, W is the diameter of the target

Table 6: Summary of results from model fitting and ablated inference studies: statistical significance was determined through paired t-test.

Evaluation Metrics		Full Inference			Two-Fold Cross Validation			Summary-Only Inference	Gaze-Ablated Inference
		Aim-and-Shoot Model	Baseline Model	p-value	Aim-and-Shoot Model	Baseline Model	p-value	Aim-and-Shoot Model	Aim-and-Shoot Model
MAE	TCT	43.5 ms	219.6 ms	**<.001**	45.7 ms	231.7 ms	**<.001**	47.0 ms	41.6 ms
	ACC	10.8%p	13.4%p	**<.001**	12.4%p	15.5%p	**<.001**	12.4%p	12.3%p
	SCD	0.80°	N/A	-	0.88°	N/A	-	0.81°	1.18°
KLD	TCT	0.026	0.163	**<.001**	0.027	0.257	**<.001**	0.028	0.029
	ACC	0.508	0.545	**0.014**	0.617	0.605	0.661	0.514	0.570
	SCD	0.130	N/A	-	0.135	N/A	-	0.131	0.177
R^2_{Intra}	TCT	0.70	0.41	**<.001**	0.65	0.33	**<.001**	0.65	0.73
	ACC	0.71	0.56	**0.002**	0.58	0.37	**<.001**	0.68	0.63
	SCD	0.44	N/A	-	0.42	N/A	-	0.40	0.20
R^2_{Inter}	TCT	0.82	0.72	-	0.81	0.58	-	0.75	0.97
	ACC	0.74	0.19	-	0.67	0.22	-	0.72	0.69
	SCD	0.82	N/A	-	0.81	N/A	-	0.74	0.18
DTW	Camera	308.1°	504.1°	**<.001**	328.7°	524.4°	**<.001**	322.2°	295.9°
	Gaze	1.502 m	N/A	-	1.569 m	N/A	-	1.536 m	1.664 m
R^2_{Fitts}	White	0.97	0.87	**<.001**	0.95	0.76	**<.001**	0.98	0.97
	Gray	0.97	0.85	**<.001**	0.96	0.74	**<.001**	0.98	0.97

5.2.2. Fitting Performance

Table 6 summarizes the model fitting results. Our model significantly outperformed the baseline model in all of the 9 comparable evaluation metrics. In particular, our model significantly improved the prediction of TCT and intra-player variability of ACC compared to the baseline. Furthermore, our model showed moderate fitting performance for gaze-related metrics that were not obtained in the baseline model. In cross validation, when compared to the fitting results for the entire dataset, the prediction performance of the baseline model decreased more significantly than that of our model. In particular, our model simulated behavior that sufficiently complied with Fitts' law even in cross-validation, but the baseline model did not. This shows that the baseline model, despite having fewer parameters, is less free from the overfitting problem than our model.

For all metrics, Figures 14, 15, 16, and 17 show how well our model and the baseline model predicted intra-player and inter-player variability. The fitting performance of each model to the distributions of TCT, normalized shot error, and SCD can be seen in Figure 18. From these figures, we can see that the

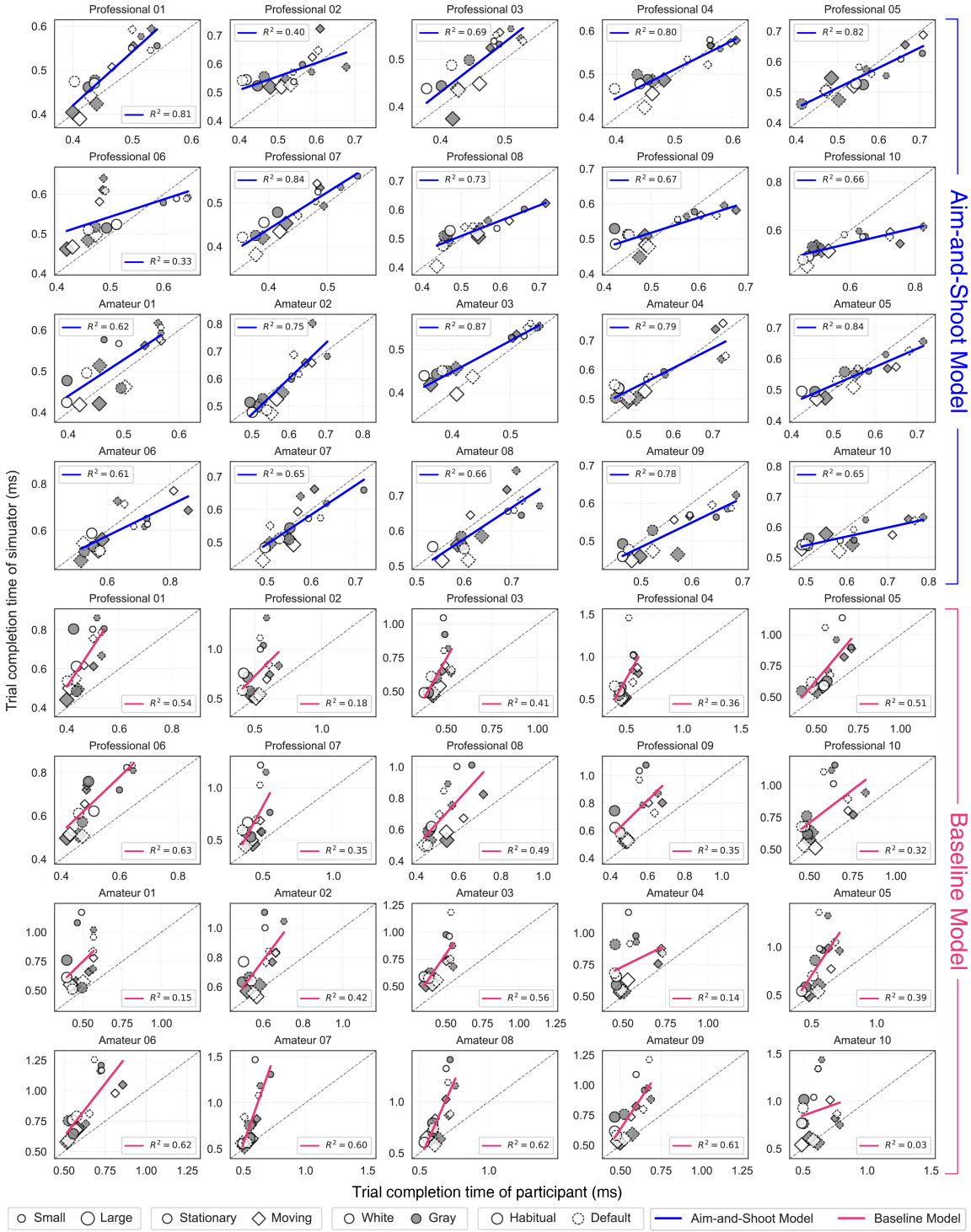

Figure 14: Correlation between simulated TCT and participant TCT (intra-player variability)

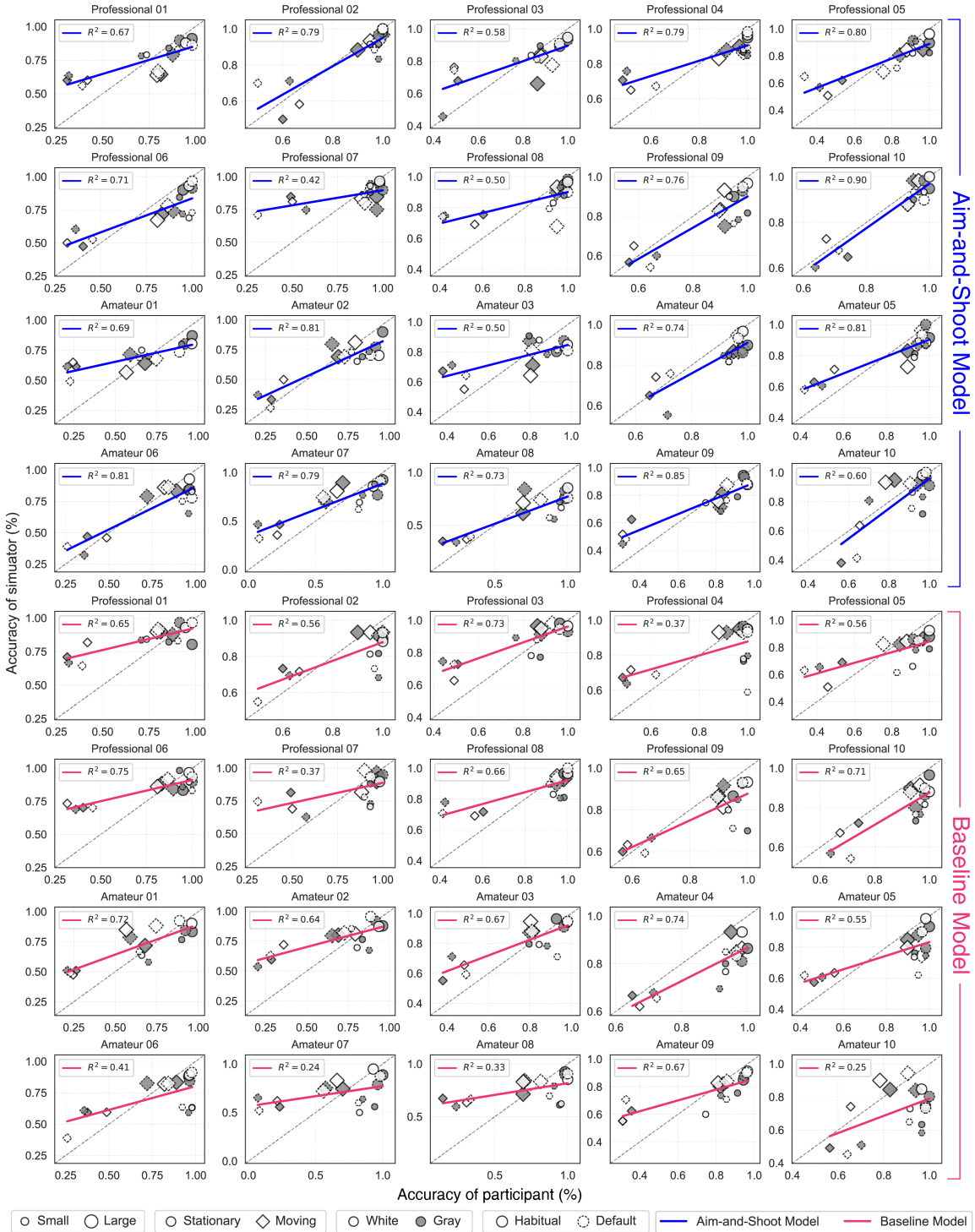

Figure 15: Correlation between simulated ACC and participant ACC (intra-player variability)

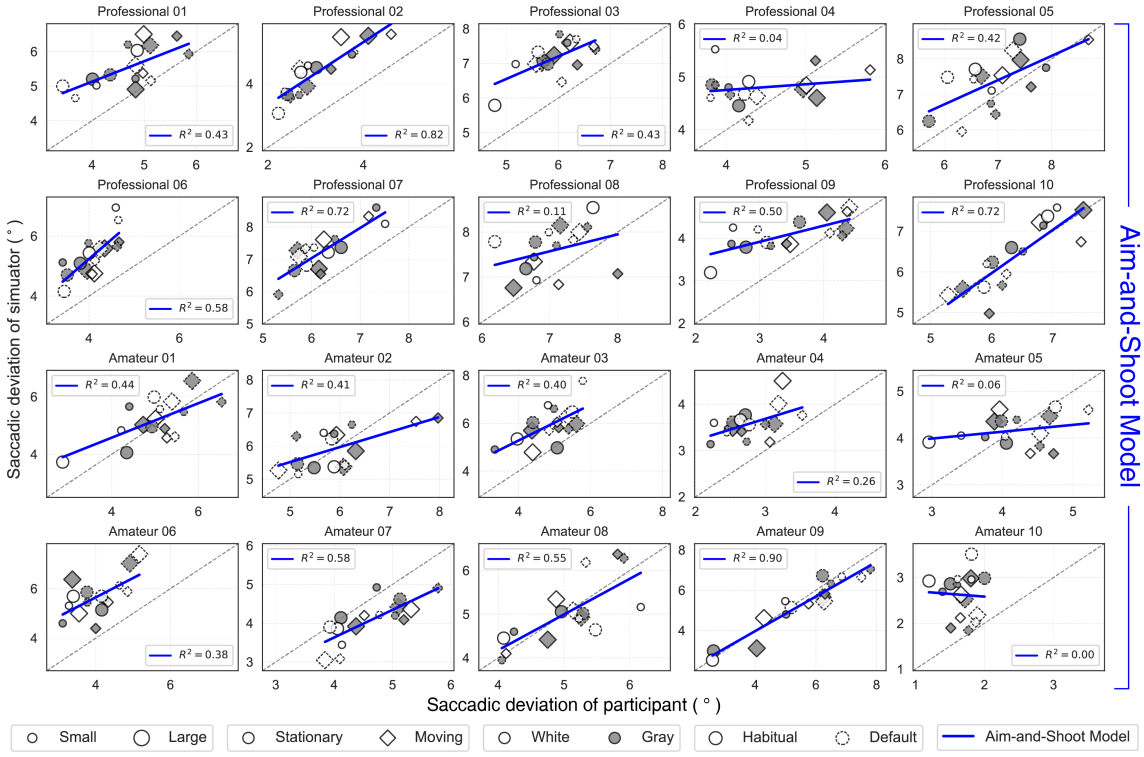

Figure 16: Correlation between simulated SCD and participant SCD (intra-player variability)

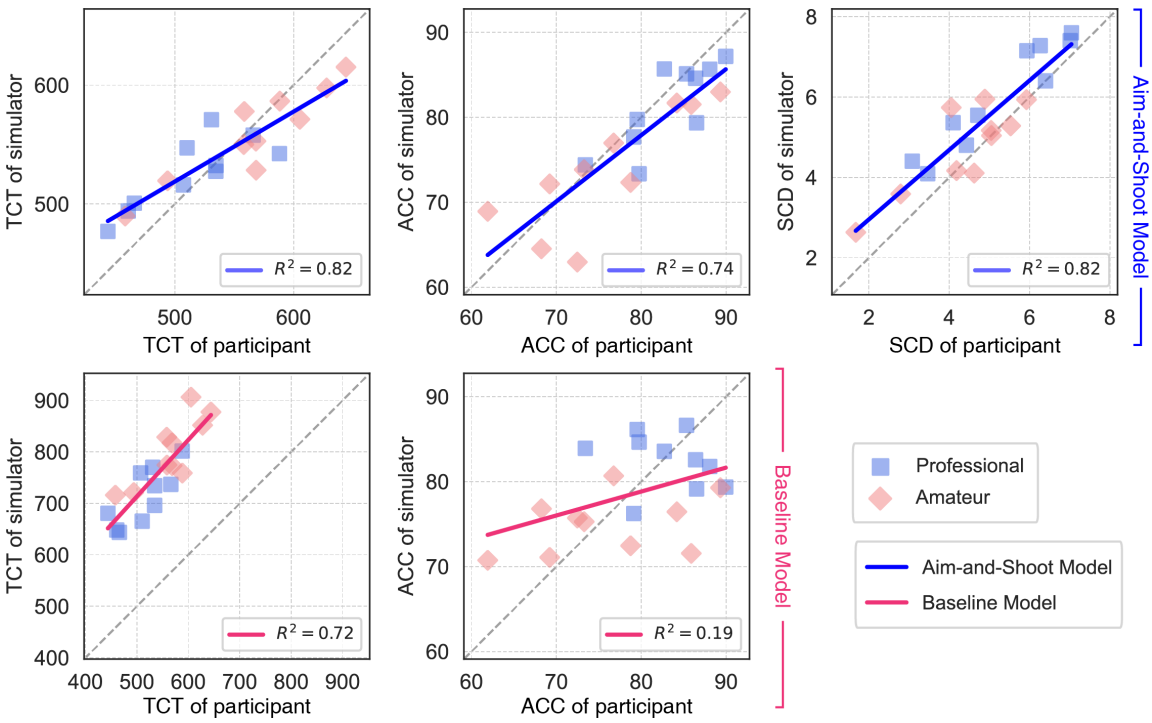

Figure 17: Correlations between simulated TCT, ACC, and SCD and participant TCT, ACC, and SCD (inter-player variability)

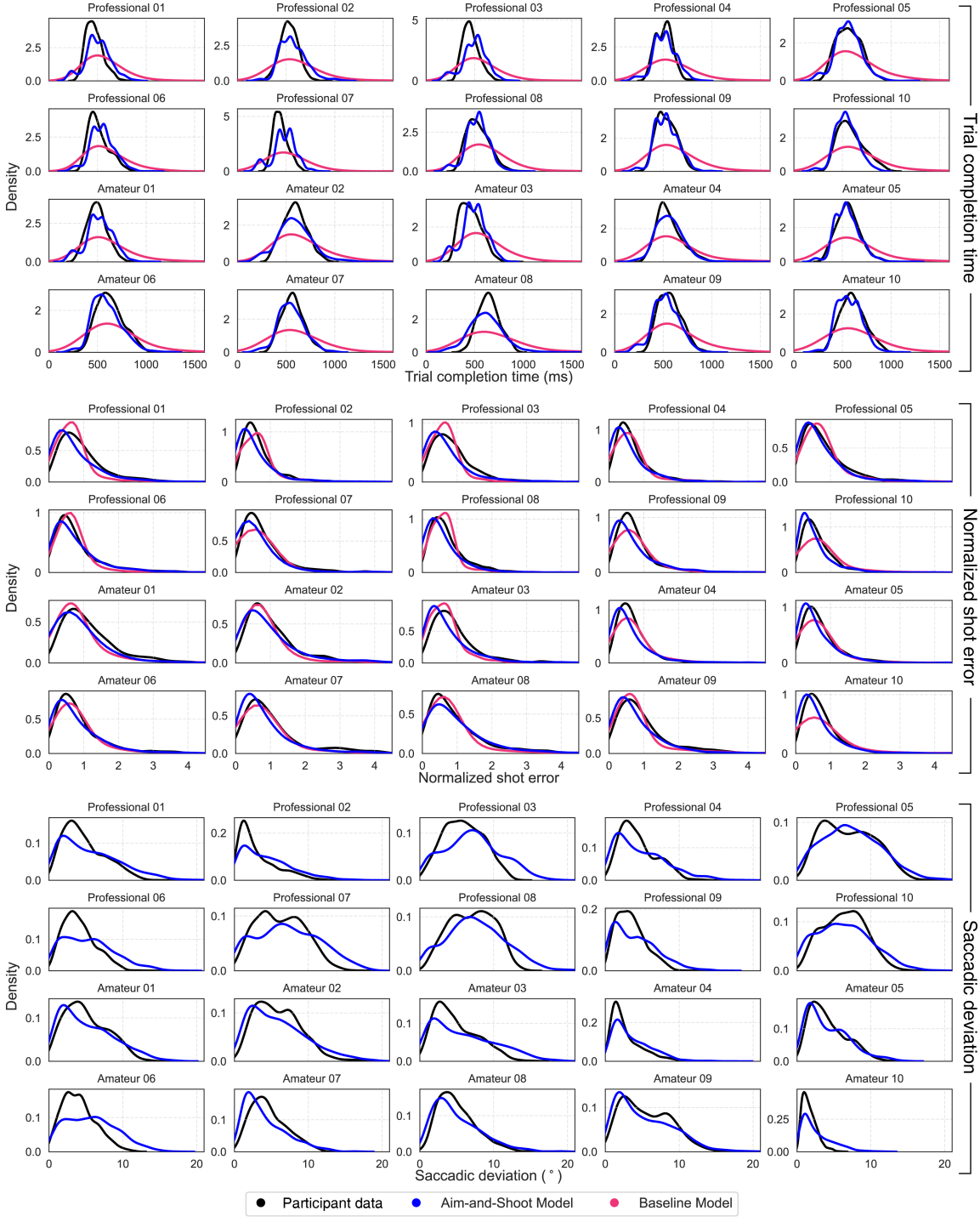

Figure 18: Distributions of TCT, normalized shot error, and SCD reproduced by each model

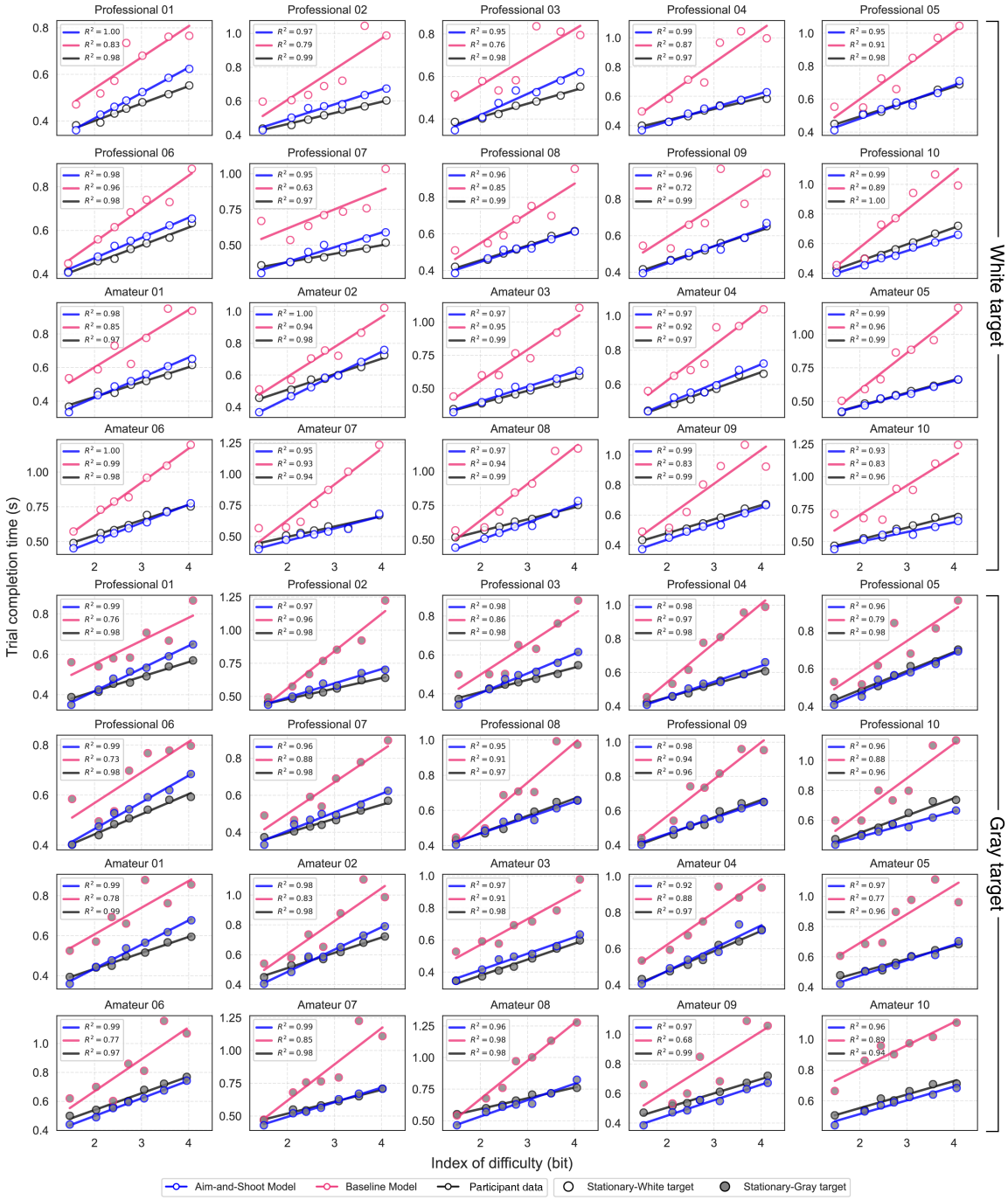

Figure 19: Fitts' law fitting results of model simulations and participant data

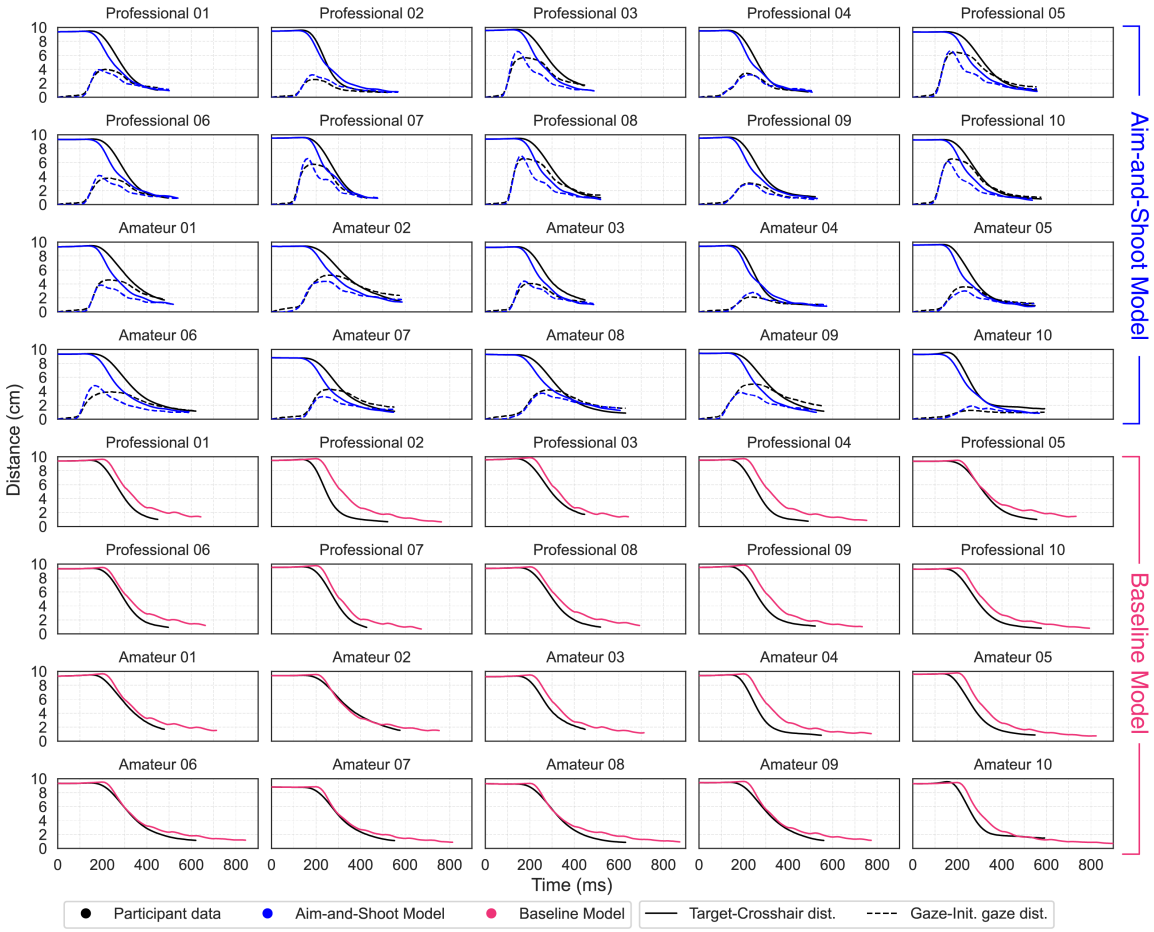

Figure 20: Changes in the distance from the crosshair to the target and the distance between the initial gaze and the current gaze over time: comparison of model simulations with actual participant behavior

TCT simulated by the baseline model tends to be significantly higher than the TCT of actual participants. This trend is also visible in Fitts' law fitting results for stationary targets (see Figure 19); Simulations of the baseline model showed higher Fitts' law model slopes (b) than actual participants and also showed poor fitting performance (R^2_{Fitts}). Similar to Figure 11, we displayed how the target and gaze moved in the player's and simulation's behavior in Figure 20. Simulations of our model tend to show faster return saccade after peak saccadic deviation. Simulations of the baseline model performed shots more slowly.

5.2.3. Fit Parameters

By analyzing the dataset obtained from the previous user study, we found that Professionals had lower TCT and higher ACC than Amateurs. We also analyzed whether Group had a significant effect on the model parameters obtained through fitting. For this purpose, we performed a mixed ANOVA analysis with an α level of 0.05 for each parameter and including all independent variables (Group, Color, and Sensitivity).

Upper two rows in Figure 21 show the Group differences in parameters obtained by fitting our model and the baseline model to the full dataset. Our model found statistically significant differences between Professionals and Amateurs for only one parameter: λ_h ($F_{1,18}$=11.621, p=0.003, η_p^2=0.392). The λ_h refers to the rate at which trial success rewards decay over time, and the λ_h of Professionals (M=43.1, SD=5.8) was 21.1% higher than that of Amateurs (M=35.6, SD=6.1). In other words, professionals have a stronger motivation to reduce the time required for a successful shot. On the other hand, λ_m, which refers to the rate at which the failure penalty decays over time, tended to be 21.9% higher for Amateurs (M=50.8, SD=20.1) than for Professionals (M=39.7, SD=11.5). In other words, Professionals have the motivational characteristic of trying to avoid failure more consistently and persistently (not statistically significant, but with a moderate effect size). Looking at other effects that were not statistically significant but had relatively high effect sizes, we can see that Professionals are able to perceive the target's position (η_p^2=0.166) and speed (η_p^2=0.068) more precisely and also receive higher rewards when successfully acquiring the target (η_p^2=0.115). It is also worth noting that motor noise, which was expected to significantly contribute to aim-and-shoot performance, was not noticeably lower in Professionals than in Amateurs ($F_{1,18}$=0.791, p=0.404, η_p^2=0.039). The Group effect observed in the baseline fitting was overall similar to that of our model, except for the result that the motor noise of Amateurs (M=0.31, SD=0.09) was significantly higher than that of Professionals (M=0.18, SD=0.06): $F_{1,18}$=12.507, p=0.002, η_p^2=0.410.

The third to sixth rows in Figure 21 show the main effects of other independent variables on each model parameter. In our model fitting, as the contrast of target color decreases, noise perceiving target position or speed tends to increase, which is in good agreement with findings from previous visual perception studies (Stocker and Simoncelli, 2006; Avidan et al., 2002; Duinkharjav et al., 2022). On the other hand, the factor Color can be expected to have no significant effect on motor noise (as shown in the fitting results of our model), but

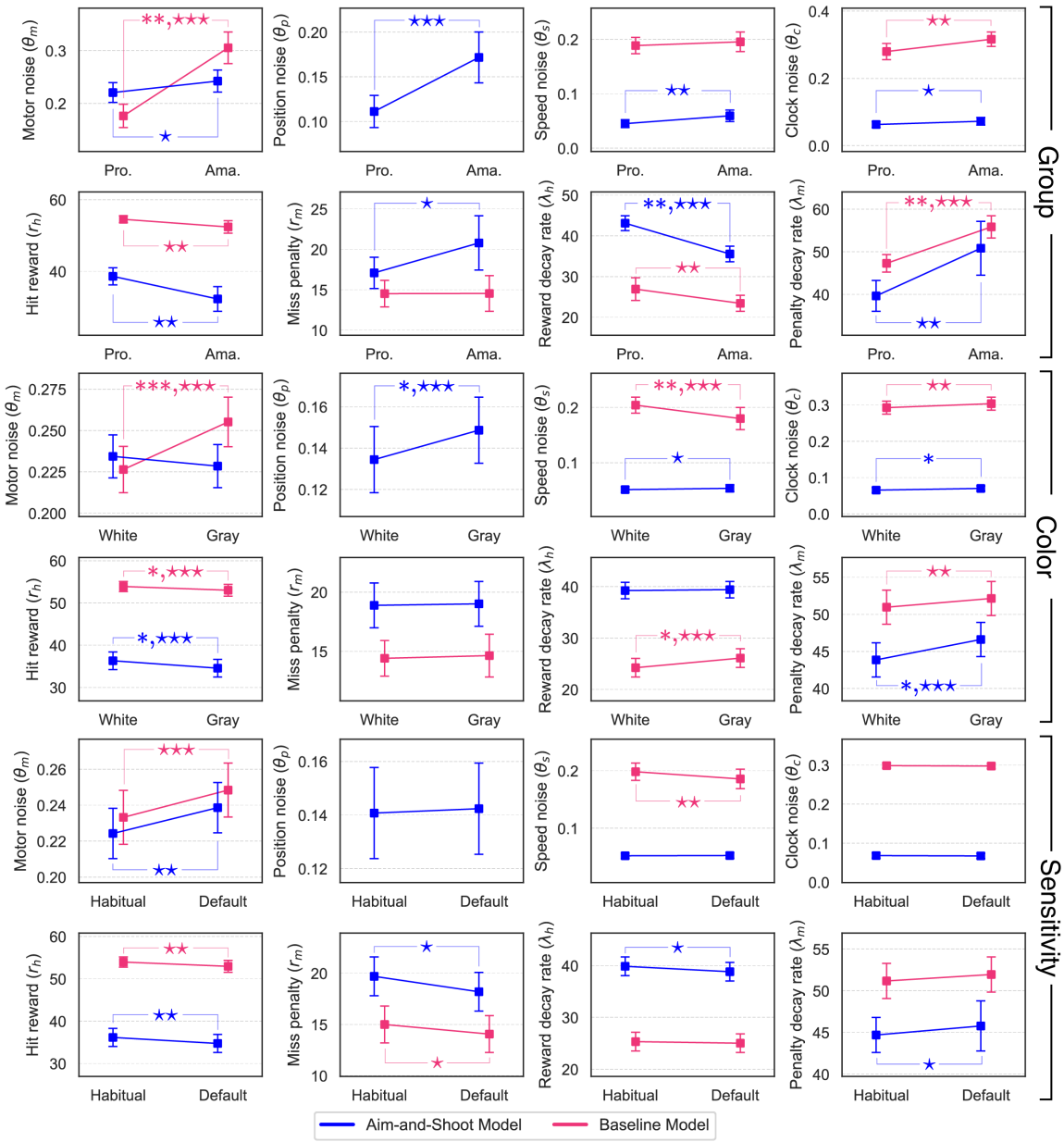

Figure 21: Model parameters inferred by the aim-and-shoot model and the baseline model: the error bar represents a 95% confidence interval; statistically significant differences are indicated as asterisks ($*$: $p < 0.05$, $**$: $p < 0.01$); effect sizes are indicated as stars ($\star$: $\eta_p^2 \geq 0.01$, $\star\star$: $\eta_p^2 \geq 0.06$, $\star\star\star$: $\eta_p^2 \geq 0.14$).

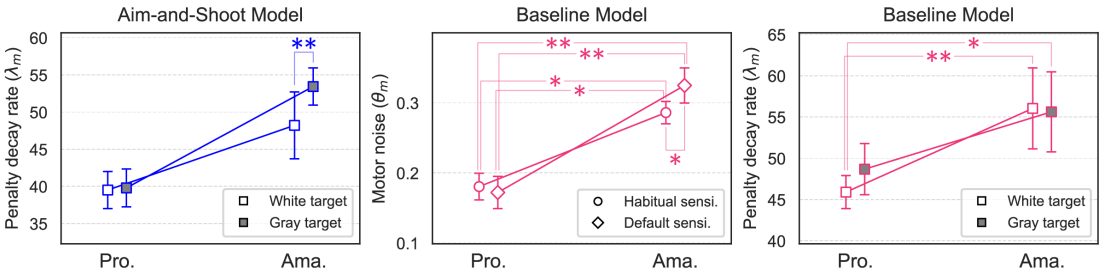

Figure 22: Interaction effects of Group, Color, and Sensitivity on inferred parameters: the error bar represents a 95% confidence interval; statistically significant differences are indicated as asterisks ($*$: $p < 0.05$, $**$: $p < 0.01$).

when fitting the baseline model, motor noise was found to be significantly larger in the Gray condition: $F_{1,18}$=19.563, p¡0.001, η_p^2=0.521. Furthermore, baseline model fitting showed that speed perception noise was statistically lower in the Gray condition ($F_{1,18}$=12.018, p=0.003, η_p^2=0.400), which conflicts with findings from previous studies (Stocker and Simoncelli, 2006; Avidan et al., 2002; Duinkharjav et al., 2022). The effect of the Sensitivity factor was observed similarly in both our and baseline model fittings; In particular, although it was not statistically significant, participants' motor noise tended to increase in the unfamiliar Sensitivity (i.e., Default) condition. Meanwhile, the baseline model fitting showed a tendency for the amount of speed perception noise to decrease in the Default condition, although it is difficult to explain theoretically.

Only three interaction effects were significant: (our model) the effect of the interaction between Color and Group on λ_m (p=0.015), (baseline) the effect of the interaction between Group and Sensitivity on θ_m (p=0.008), and the effect of the interaction between Group and Color on λ_m (p=0.049). These three interaction effects are depicted in Figure 22.

5.3. Feature-Ablated Inference

From the perspective of practitioners who wish to apply our model to the training and evaluation of esports athletes, all of the information required to perform inference may not always be measurable. For example, collecting raw trajectories of the camera and eyes may be burdensome due to storage space issues, or gaze-related data may not be obtained at all due to the lack of an eye tracker. In this section, we verify the fitting performance of our model when only partial information about aim-and-shoot behavior is given.

While keeping the neural network architecture intact, we trained two new amortized inference engines: (1) Summary-Only and (2) Gaze-Ablated. The Summary-Only engine excludes target and camera orientation trajectories and only the following summary statistics are given as input to the network: TCT, normalized shot error, SCD, both mouse and gaze reaction times, and initial task conditions. In the Gaze-Abalated engine, all information related to gaze is excluded. At this time, gaze reaction time (149 ms) and head position (0 m,

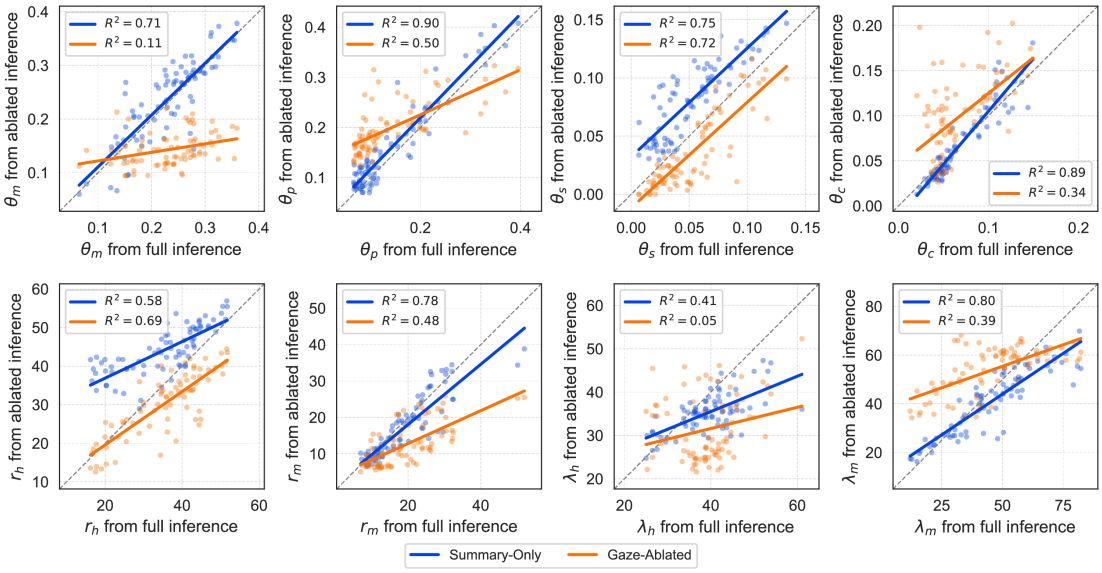

Figure 23: The correlation between the parameters obtained from the full inference (x-axis) and from the ablated inference (y-axis)

0.08 m, 0.575 m) were fixed as the overall average of the dataset measured in the user study.

The model fitting performance through the two feature-ablated engines is included in the right two columns of Table 6. The model fitting performance of Summary-Only inference was worse than that of the full-inference engine, but still showed sufficient performance for practical use. The performance of Gaze-Ablated inference showed similar result to the full-inference except that gaze-based evaluations substantially deteriorated, especially in R^2_{Inter} of SCD (from 0.82 to 0.18). On the other hand, Figure 23 shows the correlation between model parameters fitted with the full inference engine and parameters fitted with the feature-ablated engines. Overall, the coefficient of determination (R^2) of the Summary-Only engine (M=0.73) tended to be higher than that of the Gaze-Ablated engine (M=0.41). In summary, both engines can be useful for predicting TCT or ACC, but if the reliability of parameters obtained through fitting must be guaranteed, the use of the Gaze-Ablated engine should be avoided.

6. Discussion

The contributions of this study presented up to this point can be summarized as follows: (1) a CR model that precisely simulates the cognitive process underlying aim-and-shoot behavior was proposed and implemented (Section 3); (2) aim-and-shoot behavior of 20 FPS players was collected (Section 4); (3) The simulation performance of our model and the baseline model was evaluated

through model fitting based on amortized inference (Section 5). The results of the model fitting study are:

- Our model fit the aim-and-shoot behavior of real players significantly better than the baseline and showed less risk of overfitting.
- The unique cognitive characteristics of professionals that allow them to outperform amateurs have been revealed; professionals had lower levels of cognitive noise overall and a stronger motivation for quick success.
- The parameters of the aim-and-shoot model can be robustly estimated using only the summary-statistics of mouse (or view camera) and gaze movement patterns.
- Tracking players' gaze seems essential for meaningful analysis of aim-and-shoot behavior.

In this section, we discuss the significance of the above findings in more depth.

Beating The Baseline

Our model was able to fit the TCT, ACC (or normalized shot error), and SCD of real players much better than the baseline, despite having only one more free parameter (i.e., θ_p). What is more noteworthy is that the performance of the baseline deteriorated more significantly than our model in two-fold cross-validation (see Table 6). In other words, the baseline model had a higher risk of overfitting even though it had fewer parameters. This can be confirmed again in the analysis of model fit parameters. Unlike our model, main effects of independent variables that were difficult to explain theoretically were observed in the baseline model fitting. For example, in baseline fitting, the motor noise parameter (θ_m) significantly varied by the target color, which is logically difficult to explain that signal-dependent motor noise, which is added *independently* of the visual perceptual process (see Equation 2), varies depending on the target color. As target color contrast decreases, the precision of target position (Hussain et al., 2015) and speed (Stocker and Simoncelli, 2006) perception decreases, resulting in lower aim-and-shoot performance (see Table 5), and we speculate that the baseline model overfits such performance degradation with a single motor noise parameter. Baseline model fitting also showed that the speed noise parameter (θ_s) was lower when the target color was gray than when it was white, which also contradicts findings in previous studies. In general, target color have not been considered as key variables in modeling human aimed movement (Do et al., 2021; MacKenzie, 1992; Looser et al., 2005; Ikkala et al., 2022), but our results show that an experimental design that takes them into account can differentiate the effects of visual perception and motor noise to more rigorously verify the model's performance.

Assuming that the participants in our study are not outliers, we can also diagnose whether the models are overfitting the dataset by examining whether the model fit parameters are within the range reported in previous studies. Table 7 shows the ranges of each parameter reported in previous studies and the mean and standard deviation of the model fit parameters obtained in this study. Most model fit parameters were well within the range reported in previous studies, but the internal clock noise (θ_c) estimated from the baseline model

Table 7: Mean and standard deviation of inferred parameters, and the range of parameters investigated in previous literature

Parameter	Aim-and-Shoot Model	Baseline Model	Range Observed
θ_m	0.23 (SD=0.06)	0.24 (SD=0.11)	[0.10, 0.42][1]
θ_p	0.14 (SD=0.08)	-	[0.09, 0.33][2]
θ_s	0.05 (SD=0.03)	0.19 (SD=0.05)	[0.05, 0.40][3]
θ_c	0.06 (SD=0.03)	0.30 (SD=0.07)	[0.07, 0.20][4]

[1](Lin and Tsai, 2015; Do et al., 2021; Moon et al., 2022), [2](Hussain et al., 2015), [3](Moon et al., 2022; Stocker and Simoncelli, 2006; Do et al., 2021), [4](Lee and Oulasvirta, 2016; Lee et al., 2018; Lee, 2022)

tended to be excessively high. Similar to the case of the motor noise parameter, we interpret this as the baseline model overfitting missed shot cases (presumably due to visual noise) through forced adjustment of the internal clock parameter.

Aside from the problems with model fit parameters, the baseline model showed lower performance than our model in the fitting of TCT and ACC. In particular, it is noteworthy that the TCT of the baseline model was significantly longer (approximately more than 200 ms) than that of actual participants, regardless of whether the target was stationary or moving. This may be partly due to the fact that the hand reaction time of the baseline model was fixed at 200 ms, which is about 40 ms longer than the average of actual participants (see Table 5). Furthermore, the TCT error of the baseline model tended to be amplified as the task became more difficult, which may be because the noise parameters of the baseline model were fitted too high compared to the reality. If the noise parameters of the baseline model are lowered to better fit the TCT on relatively high difficulty tasks, the TCT may instead unintentionally become shorter than that of actual participants on relatively easy tasks. In fact, the distributions of TCT in Figure 18 show that the baseline model fits the mode of the distributions well overall, while giving up fitting the two tails. This is an interesting result, considering that the point-and-click model, which is the parent of the baseline model, successfully replicated both the TCT and end point distributions in the 2D point-and-click task (Do et al., 2021). These results provide evidence that the unique perceptual challenges of aim-and-shoot that we postulated may be real and that a deep understanding of aim-and-shoot performance is difficult without considering them.

The Professional Behavior

It is already widely known that professional FPS players have aim-and-shoot performance that overwhelms amateurs (Park et al., 2021; Dahl et al., 2021; Rogers et al., 2024), and is consistent with the findings in this study. Factors that make the difference have mainly been pointed out as professional players'

shorter reaction times (Park et al., 2021; Koposov et al., 2020), more accurate motor control skills (Donovan et al., 2022; Park et al., 2021), and more optimized peripheral settings (Watson et al., 2024; Boudaoud et al., 2023; Kim et al., 2020; Lee et al., 2020). Some of those differences were replicated well in our study: the pros showed, on average, 38 ms shorter gaze reaction times and 8.3% lower motor noise (θ_m). The difference in motor noise between the two groups was not as large as expected, but this does not directly mean that the overall effect of motor noise on aim and shoot performance in FPS is small. Readers should note that this study targeted only a small portion[14] of the complex and diverse aim-and-shoot skills required in actual FPS games. In real FPS, where more complex target movements are given, the differences in motor noise between the two groups revealed in this study may result in more critical differences in game performance (Allard et al., 1980; Allard and Starkes, 1980).

Meanwhile, the most interesting discovery we made in model fitting was the high motivation of professional players to perform trials quickly and successfully (higher λ_h). Two different interpretations are possible for this. First, to facilitate recruitment, we provided professional players with higher compensation than amateurs, which may have resulted in them entering the study with a higher motivational state. In fact, model fitting also showed that the r_h reward from trial success was higher for professionals, although this was not statistically significant (but with a medium effect size). From one perspective, this may be considered a failure of experimental control, but on the other hand, it can be interpreted that we have once again confirmed the realism of the model simulation, as the compensation differences in reality are reflected in the actual model fit parameters. We believe that our study is the first to observe reward-related model fit parameters in a CR model changing correspondingly to the amount of monetary compensation given in the experiment.

The second interpretation focuses on the fact that professionals sought faster success while also being less willing to give up quickly (lower λ_m, η_p^2=0.113). In our experiments, failed trials were not repeated, so in terms of hourly reward, it is better to quickly give up on trials that take relatively longer time (e.g., small and fast targets). Therefore, we speculate that other hidden factors, not just greater monetary compensation, cause professionals to have motivational characteristics that distinguish them from amateurs. For example, one factor may be that the aim-and-shoot scenarios that professionals have experienced are significantly different from those of amateurs. In a typical FPS, players are matched with players of similar skill levels, so professional players are naturally placed under stronger time pressure. Since failing to shoot an enemy in a professional match is more likely to lead to one's death, players may have learned to have a unique motivational state that seeks faster success and not giving up.

So, if professionals and amateurs had similar motivational parameters (r_h, r_m, λ_h, λ_m), how different would their performance be? In general, the mo-

[14]No camera translation was allowed and the target was circular and moved at a constant velocity.

Table 8: Mean and standard deviation of trial completion time and accuracy from the empirical data, simulation data from the model fitting, and simulation data where the reward parameters of professionals and amateurs are swapped

Performance	**Group**	**Empirical**	**Simulation (Original result)**	**Simulation (Reward swapped)**
TCT (ms)	Pro.	514.2 (SD=111.7)	526.7 (SD=128.0)	559.4 (SD=118.2)
	Ama.	565.6 (SD=130.0)	557.9 (SD=156.9)	594.3 (SD=140.9)
ACC (%)	Pro.	83.1 (SD=37.4)	81.3 (SD=39.0)	86.5 (SD=34.2)
	Ama.	76.4 (SD=42.4)	74.2 (SD=43.8)	81.7 (SD=38.7)

tivational state of user study participants cannot be directly and accurately controlled (Moon et al., 2022), but in answering this question, our model can predict changes in aim-and-shoot performance by varying only the motivational parameters while keeping cognitive characteristics fixed. Table 8 shows the model's simulation results when the motivational parameters of professional or amateur players are set to the average of the opposing group. This additional analysis reveals a significant effect of motivational state on TCT and ACC in our model. It has also been confirmed in previous studies that participant motivation has a significant impact on reaction (Ziv et al., 2022) or pointing (Moon et al., 2022) performance. Another interesting observation is that the model predicts that even if amateurs have the motivational state of professionals, they will still have lower aim-and-shoot performance than professionals, possibly due to fundamental differences in cognitive characteristics.

It also needs to be verified in future studies whether the parameters obtained through fitting our model correspond well to the predictions of traditional psychological theories that deal with human motivation, such as flow theory (Csikszentmihalyi and Csikzentmihaly, 1990) or attributional theory (Kukla, 1972; Weiner, 1985). For example, flow theory (Csikszentmihalyi and Csikzentmihaly, 1990) predicts that the difficulty of a given task can affect motivation to perform the task. In fact, in our study, a statistically significantly higher hit reward (r_h) was fitted for the white target, which was easier to obtain, than for the gray target. However, in this study, other variables that significantly change task difficulty (e.g. target radius or speed) were assumed to have no effect on reward parameters, and the validity of this assumption needs to be tested more rigorously in future studies. At this point, readers may wonder whether model fitting could be performed on the smallest task units to explore the effect of target radius or speed on reward parameters. For example, in this study, model fitting could be performed for each of a total of 16 unique conditions ($2\times2\times2\times2$). However, in our experience, this significantly increases the risk of model overfit because it reduces the effective size of the dataset and also unnecessarily mobilizes perceptual noise parameters in the fitting, which are assumed to be unaffected by target radius or speed in the model. In other words, we should probably find a way to keep cognitive parameters fixed regardless of

target radius or speed during the model fitting process, while allowing reward parameters to vary freely. To our knowledge, this is a challenge that has not been addressed in previous CR models based on amortized inference. One possible solution might be to express the reward the agent receives as a function of target radius (R) and speed (S), as shown below[15]:

$$r_h = r_{h0}(c_0 + c_1 \cdot R + c_2 \cdot S) \tag{14}$$

where c_0, c_1, and c_2 are free parameters. However, this method significantly increases the number of model free parameters and may require more computing resources as well as advanced RL and inference architecture.

7. Guidelines for Gaze Control

In the CR framework, if a player is a novice, it means that the player has *not yet learned* the optimal action policy π within the given cognitive bounds (e.g., motor or vision noise). From that perspective, the way serious players of competitive video games train today is not efficient; they often blindly copy the actions seen in videos of professional players to improve their skills (Park et al., 2021). The cognitive bounds of professionals may differ significantly from those of amateurs, and therefore the optimal behavioral policy they should pursue may also differ significantly from that of professionals. Our model is implemented under the CR framework, so theoretically it can provide guidelines on what the optimal aim-and-shoot behavior is for a player, regardless of his or her cognitive characteristics.

To show the practical value of CR modeling and help efficient training of novice FPS players, we provide quantitative guidelines on what the optimal gaze control policy is for performing aim-and-shoot tasks, based on our model simulations. This decision took into account the fact that there are relatively fewer guidelines for optimal gaze control[16] than for optimal mouse control (Kang et al., 2024) in FPS, and that eye trackers are generally not affordable to amateur players, making it difficult for them to perform gaze analysis themselves. Our guidelines aim to show how much it is appropriate for gaze to deviate from the crosshair on average (i.e., the SCD metric) when target distance, speed, and radius each change over a wide range. We prepared and simulated a total of four different agents: (1) Low cognitive noise & low motivational state (`LL`), (2) low cognitive noise & high motivational state (`LH`), (3) high cognitive noise & low motivational state (`HL`), (4) High cognitive noise & high motivational state (`HH`). The parameter settings for each agent are in Table 9.

As a result of the simulation, Figure 24 shows how the SCD of each agent changed on average when each target condition varied. The target velocity condition was divided into cases where the target was moving away from the

[15] With reference to the flow theory, a quadratic function may be introduced instead.

[16] We did an informal web search and found that guiding articles on eye movement in FPS are rare compared to those on mouse movement.

Table 9: The parameter settings for the agent LL, LH, HL, and HH

Agent	θ_m	θ_p	θ_s	θ_c	r_h	r_m	λ_h	λ_m
LL	0.18	0.08	0.03	0.04	28.81	24.07	35.91	56.51
LH	0.18	0.08	0.03	0.04	43.05	11.62	43.58	32.55
HL	0.28	0.19	0.07	0.09	28.81	24.07	35.91	56.51
HH	0.28	0.19	0.07	0.09	43.05	11.62	43.58	32.55

crosshair and cases where the target was approaching the crosshair. When simulating a specific target condition, the remaining conditions were randomized within the range specified in the scenario in Section 3.1. Based on the interpretation of the graphs, we can provide the following guidelines to FPS players:

- For targets that are more difficult to hit, such as those that are farther away or smaller in size, it is better to move the gaze closer to the target.
- Even if the target is large enough, it is recommended that players with relatively high congitive noise do not fixate their eyes on the crosshair.
- Considering the target's movement direction and speed, the gaze must be moved predictably to the target's future position; That is, targets moving away from the crosshair can result in a higher SCD, and targets moving closer can result in a lower SCD.
- In conditions where the target approaches the crosshair, players with higher cognitive noise are recommended to have a gaze control policy that takes into account that the crosshair may overshoot or pass through the target. This may result in a higher SCD.
- Even if there is a greater motivation to hit the target, there is no need to significantly change the gaze control policy.

However, note that the above guidelines apply only when the target is moving at constant velocity, and there is no camera translation. Furthermore, our model assumes that the gaze is fixed on the crosshair at the time a target is given.

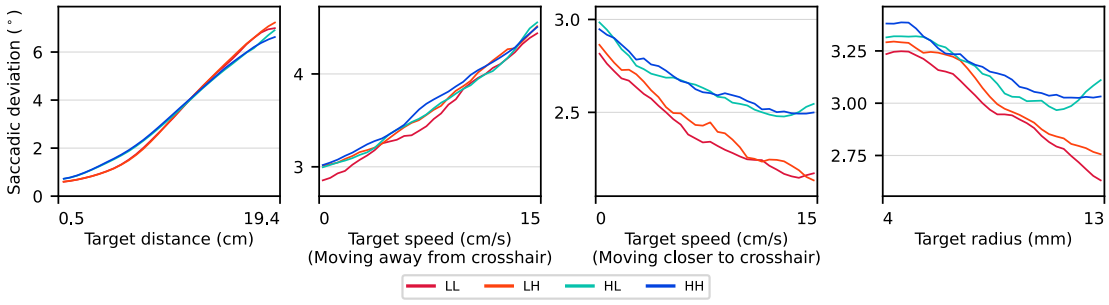

Figure 24: Average SCD of four agents (LL, LH, HL, HH) over increasing target distance, speed, and radius

8. Limitations and Future Work

This study has several limitations, which may lead to interesting follow-up studies in the near future. First, in contrast to the extensive treatment of the underlying cognitive mechanisms, this study deals with an aim-and-shoot task scenario that is significantly simpler than those presented in real FPS games. For targets with more complex movement patterns and shapes, further research is needed to determine how the perceptual and motor control modules proposed in this study should be modified. Furthermore, there may be multiple on-screen targets with many distracting visual elements in the background. To extend our model to such more realistic situations, we anticipate that it will be essential to develop an image-based perception module that directly receives screen pixel information as input (Kempka et al., 2016). An image-based peripheral vision model (Rosenholtz et al., 2012; Lukanov et al., 2021; Wells-Gray et al., 2016) or a saccadic suppression model (Matin, 1974) could also be included in the model.

Second, our model does not take into account a fundamental aspect of any FPS: the presence of enemies. In future research, a scenario where multiple FPS agents aim and shoot each other could be formulated as a multi-agent RL problem (Vinyals et al., 2019). To implement more realistic competitive scenarios, it is also necessary to expand the action space so that the agent can translate the view camera (e.g., by pressing keyboard buttons with left hand fingers). Implementation of the multi-agent model's parameter inference engine and its use in FPS training should also be addressed.

Third, the Gaze module implemented in this study has room for significant improvement. In particular, the module assumed that gaze control decisions occur every 100 ms in synchronization with the hand control cycle, which makes the model's fixation duration shorter than reality (i.e., 200 to 300 ms) (Einhäuser and Nuthmann, 2016). This is the reason why the gaze returns to the crosshair in our model's simulation faster than that of the participants (see Figure 20). Furthermore, target characteristics (e.g., familiarity) cause human fixation duration to vary significantly (Loftus and Mackworth, 1978; Salvucci, 2001), and our model with a fixed duration of 100 ms cannot replicate that phenomenon. To solve these issues, hierarchical RL could be applied with hand control policy and gaze control policy being independent, as was done in a recent study of typing behavior (Jokinen et al., 2021).

Lastly, this study did not consider in modeling that upper limb kinetics have a significant impact on mouse control performance (Lee and Bang, 2015; Kang et al., 2024). In particular, we speculate that the effect on mouse sensitivity observed in this study is deeply related to upper limb kinetics. With the rapid development of human body biomechanics simulation libraries (Saul et al., 2015; Todorov et al., 2012), several recent CR modeling studies (Moon et al., 2024; Ikkala et al., 2022; Fischer et al., 2021; Hetzel et al., 2021) that included them as model components have achieved good results in predicting human input behavior. The Aim module of our model can be extended to have a biomechanics component and allow the agent to move the mouse through the activation of muscle tendons. We expect that such extensions will allow us to address poorly

understood mechanisms in input performance, such as how wrist-aiming habit affects performance and workload (Kang et al., 2024), or why optimal mouse transfer functions exist (Lee et al., 2020).

9. Conclusion

In this study, we presented a CR model that can realistically simulate the process by which FPS players control their hands and gaze to aim-and-shoot a moving target on the screen. The model can broadly replicate the human cognitive mechanisms underlying the aim-and-shoot process, for example, intermittent motor control and signal-dependent motor noise for motor planning and execution, and saccadic main sequence and peripheral vision for visual perception. Based on amortized inference engines, both our model and a baseline model were fitted to a dataset of aim-and-shoot behavior of 20 FPS players, including 10 professionals. As a result, we confirmed that our model has significantly higher fitting performance on several key metrics and less risk of overfitting than the baseline. Through analysis of model fit parameters, we also discovered hidden reasons why professionals can outperform amateurs. Professionals had lower overall levels of cognitive noise than amateurs and also had distinct motivational states, such as seeking quick success and not giving up easily. We expect our model to open a new chapter in FPS skill analysis, in the field of esports where even the slightest performance differences can be critical (Park et al., 2021; Lee et al., 2024b; Kang et al., 2024). From a broader HCI research perspective, our model provides a useful starting point for tackling important remaining challenges in CR modeling, such as the inclusion of biologically plausible image-based visual perception or the exploration of competitive/cooperative multiagent scenarios.

Acknowledgements

This research was funded by National Research Foundation of Korea [RS-2023-00223062], and Institute of Information and Communications Technology Planning and Evaluation [2020-0-01361]. We thank Game Coach Academy (GCA) for their assistance in participant recruitment.

Appendices

A. The Aim-and-Shoot Model Details

A.1. Task Initial Condition

Table A1 shows the detailed range and sampling distribution. The head position, gaze position, and reaction time distributions were set based on the empirical dataset (Section 4).

Table A1: The range and distribution of the aim-and-shoot task initial conditions and agent initial states.

	Condition & State	Distribution	Constraint	Reference	Unit
Task condition	Target position (azimuth)	$\mathcal{U}(-37,-2)\cup\mathcal{U}(2,37)$		0	°
	Target position (elevation)	$\mathcal{U}(-21,-1)\cup\mathcal{U}(1,21)$			
	Target radius	$\mathcal{U}(4,13)$			mm
	Target angular speed	$\mathcal{U}(0,40)$			°/s
	Target orbit axis (azimuth)	$\mathcal{U}(-180,180)$		0	°
	Target orbit axis (elevation)	$\mathcal{U}(-90,90)$			
Agent state	Camera direction (azimuth)	$\mathcal{U}(-1.2,1.2)$	$\|\text{direction}\| \leq 1.2$	0	°
	Camera direction (elevation)				
	Head position (vertical)	$\mathcal{N}(\mu$=0$,\sigma$=1.7)	[-6.5, 6.5]	crosshair	cm
	Head position (horizontal)	$\mathcal{N}(\mu$=7.8$,\sigma$=1.9)	[-4.2, 19.8]	crosshair	
	Head position (distance)	$\mathcal{N}(\mu$=57.5$,\sigma$=2.4)	[40.8, 74.2]	monitor	
	Gaze position	$\mathcal{N}(\mu$=$\boldsymbol{p}_c, \Sigma$=$0.62\cdot\mathbf{I})$	$\|\boldsymbol{p}_c-\boldsymbol{p}_g\| \leq 3.5$	crosshair	cm
	Mouse reaction time	$\mathcal{SN}(\mu$=132$,\sigma$=34.5$,\alpha$=1.43)	[100, 300]		ms
	Gaze reaction time	$\mathcal{SN}(\mu$=80.1$,\sigma$=88.1$,\alpha$=6.04)	[50, 300]		

A.2. Optimal Trajectory Generation

Let the pairs of 1D position and velocity are given: $\mathbf{x}(0)=[x(0),\dot{x}(0)]^T$ as an initial state and $\mathbf{x}(N)=[x(N),\dot{x}(N)]^T$ as a final state. The objective is to sample the N-step trajectory $\{\mathbf{x}(k)=[x(k),\dot{x}(k)]^T \mid k\in[0,N]\}$ between an initial and final state that satisfies the minimum acceleration criterion. Let the interval between states (i.e., sample interval) be t (s). We pre-define the following matrices:

$$\mathbf{G}=\begin{bmatrix}1 & t\\ 0 & 1\end{bmatrix},\ \mathbf{H}=\begin{bmatrix}t^2/2\\ t\end{bmatrix},\ \mathbf{\Gamma}(k)=\sum_{j=0}^{k-1}\mathbf{G}^j\mathbf{H}\mathbf{H}^T\mathbf{G}^{T(N-k+j)}\ \ (k\in[1,N])$$

Then, we can compute the k-th state of the optimal trajectory as

$$\mathbf{x}(k)=\mathbf{G}^k\mathbf{x}(0)+\mathbf{\Gamma}(k)\mathbf{\Gamma}(N)^{-1}\left(\mathbf{x}(N)-\mathbf{G}^N\mathbf{x}(0)\right)$$

For the N-dimensional state case, doing this process on each dimension separately will result in N-dimensional trajectory (in our case, N=2). For a more detailed explanation of the process, please refer to the original paper Bye and Neilson (2008).

A.3. Minimum Jerk Trajectory

For the given saccadic amplitude A and peak velocity V, the saccadic duration d is determined as $\frac{15A}{8V}$. For the time interval $t \in [0, d]$, the saccadic speed profile with the minimum jerk is expressed as

$$s_g(t) = \frac{30A}{d}\left(\left(\frac{t}{d}\right)^4 - 2\left(\frac{t}{d}\right)^3 + \left(\frac{t}{d}\right)^2\right)$$

This equation satisfies the objective constraints. First, the speed is zero at the beginning and the end of the saccade: $s_g(0) = s_g(d) = 0$. Second, it reaches the peak speed at the halfway point: $s_g(d/2) = V$. Third, the gaze lands on the destination at $t = d$: $\int_0^d s_g(t)dt = A$.

B. The Aim-and-Shoot Dataset Details

B.1. Data Processing

B.1.1. System Latency

We synchronized the game event and gaze data by subtracting the system latency of the eye tracker. We employed the following procedure to measure the latency. First, we implemented an eye-tracking logger that provided a real-time display of the user's gaze position. Next, we positioned a mirror in front of the user and below the monitor, allowing us to capture the user's eye movements while observing the eye-tracking logger's display. Utilizing an iPhone 14 Pro with slow-motion recording capabilities at 240 frames per second (fps), we recorded a video capturing both the eye-tracking logger's display and the user's eyes reflected in the mirror. We analyzed the video to identify the optimal time subtraction value that precisely synchronized the movement pattern of the user's eye with the corresponding display changes in the eye-tracking logger. Finally, we subtracted the monitor latency from the obtained value, and the measured system latency was 52 ms. This approach allowed us to accurately determine the system latency of the eye tracker by establishing the temporal offset between the user's eye movements and the corresponding updates in the eye-tracking logger's display.

B.1.2. Improving Eye Tracker Data

We applied cubic spline interpolation on data with different sampling rates. The game events were logged at 240 fps, while the gaze-related events were at 150 fps. We used the timestamp of the game events as a standard. We removed invalidly logged physical eye positions so that the interpolated data replace them. Since the eye tracker has its own filtering system for invalidly logged

Table B1: Main effect of independent variables on each dependent variable: significant differences ($p < 0.05$) are indicated in green. The values in parentheses are standard deviations.

Condition	Level	TCT (ms)	ACC (%)	SCD (°)	GRT (ms)	MRT (ms)
Player	Professional	515 (86.3)	83.0 (20.3)	5.24 (1.51)	132 (23.0)	161 (9.9)
Group	Amateur	568 (97.1)	75.6 (24.9)	4.37 (1.42)	170 (35.8)	162 (9.0)
Signif.	$F_{1,18}$, p, η_p^2	5.131, 0.036, 0.222	5.445, 0.031, 0.232	1.995, 0.175, 0.100	11.096, 0.004, 0.381	0.062, 0.807, 0.003
Target	Large	480 (59.7)	91.1 (10.1)	4.66 (1.44)	149 (31.0)	160 (9.1)
Radius	Small	603 (84.5)	67.6 (26.2)	4.95 (1.61)	153 (39.7)	163 (9.5)
Signif.	$F_{1,18}$, p, η_p^2	271.741, < 0.001, 0.938	354.094, < 0.001, 0.952	11.011, 0.004, 0.380	0.892, 0.357, 0.047	25.971, < 0.001, 0.591
Target	Stationary	530 (89.4)	93.4 (7.0)	4.56 (1.55)	150 (30.4)	161 (9.5)
Speed	Moving	553 (100.2)	64.8 (24.3)	5.05 (1.47)	152 (40.1)	161 (9.4)
Signif.	$F_{1,18}$, p, η_p^2	4.834, 0.041, 0.212	197.650, < 0.001, 0.917	31.137, < 0.001, 0.634	0.323, 0.577, 0.018	0.230, 0.637, 0.013
Target	White	533 (91.0)	79.2 (22.6)	4.78 (1.53)	145 (33.6)	158 (8.9)
Color	Gray	550 (99.6)	79.5 (23.5)	4.84 (1.54)	157 (36.6)	165 (8.6)
Signif.	$F_{1,18}$, p, η_p^2	23.922, < 0.001, 0.571	0.278, 0.604, 0.015	1.341, 0.262, 0.069	69.740, < 0.001, 0.795	145.567, < 0.001, 0.890
Mouse	Default	543 (93.8)	78.8 (23.8)	4.85 (1.42)	150 (36.3)	161 (9.7)
Sensi.	Habitual	540 (97.6)	79.9 (22.3)	4.76 (1.63)	152 (34.9)	161 (9.2)
Signif.	$F_{1,18}$, p, η_p^2	0.084, 0.775, 0.005	1.307, 0.268, 0.068	0.453, 0.510, 0.025	0.455, 0.508, 0.025	0.161, 0.693, 0.009

gaze positions (on the monitor screen), we removed it only when its filtered value is also invalid (e.g., out-of-screen).

The captured eye position data used the coordinate system originated from the eye tracker focal camera with featured axes aligned or orthogonal to the camera's direction. The eye tracker was fixed at the bottom of the monitor, with a 3 cm vertical distance from the monitor's bottom. The camera's directional vector formed an angle of 23.5° with the perpendicular vector of the monitor. To facilitate analysis, we converted the eye tracker camera's coordinate system to align with the monitor space (with the origin at the crosshair).

At last, we improved the accuracy of gaze position data in two steps. First, in the verification session of each block, we obtained 11 clusters of gaze points (i.e., fixations) corresponding to 11 fixed positions on the screen. We applied the linear transformation to gaze position data that yielded the least square error between them. Next, we translated the gaze position data so that the mean of the initial gaze position lay on the crosshair.

B.2. Empirical Data Analysis

In Table B1, we put the statistical significance unreported in Table 5.

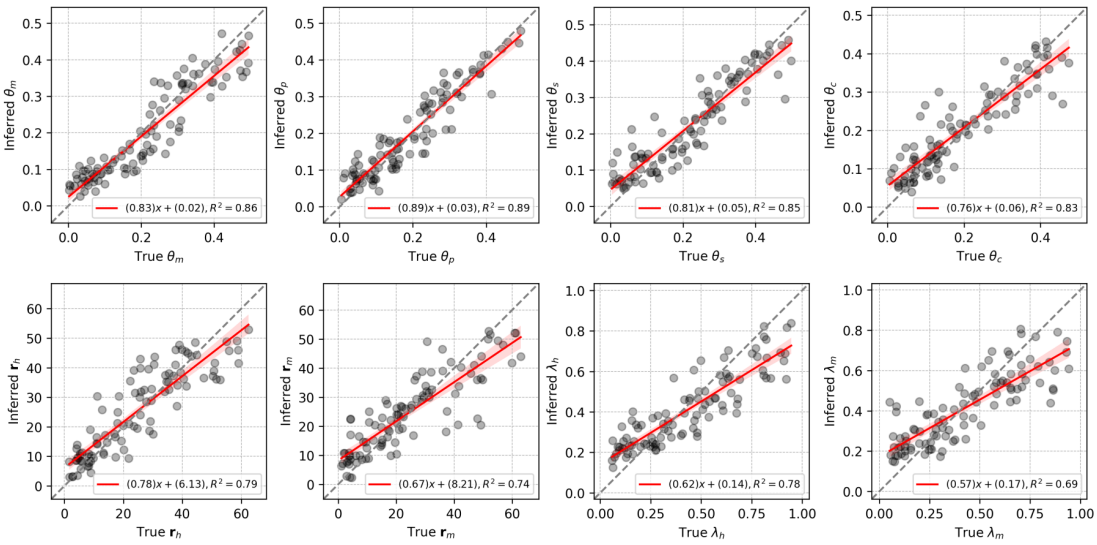

Figure C1: Parameter recovery performance of the aim-and-shoot model's inference engine (100 parameter samples × 200 observed trials).

C. The Amortized Inference Engine performance

The validation dataset consists of 100 (uniform randomly sampled) parameters and 600 simulated trials for each parameter. We randomly sampled 200 trials on each parameter and inferred the parameter using the trained inference engines. The coefficient of determination (R^2) between the true and inferred parameters refers to the parameter recovery performance. We repeated this process 100 times and averaged R^2 on every parameter. All inference engines showed moderate or strong recovery performance on overall parameters (see Figure C1 and Table C1). The result supports the reliability of the inference engines' output.

Table C1: Parameter recovery performance (R^2) of inference engines.

	θ_m	θ_p	θ_s	θ_c	r_h	r_m	λ_h	λ_m
Aim-and-Shoot Model	0.88	0.90	0.84	0.86	0.77	0.73	0.76	0.69
Baseline Model	0.94	-	0.90	0.53	0.92	0.80	0.70	0.71
Summary-Only	0.86	0.89	0.81	0.85	0.76	0.70	0.71	0.65
Gaze-Ablated	0.85	0.83	0.84	0.85	0.73	0.72	0.72	0.63

We measured the completion time of inference when 100, 200, 400, 800, or 1600 random trials were observed using the desktop environment in Section 5.1. The aim-and-shoot model, the baseline model, and Gaze-Ablated inference engine case took approximately $0.05 \cdot N + 5.8$ ms when N trials were

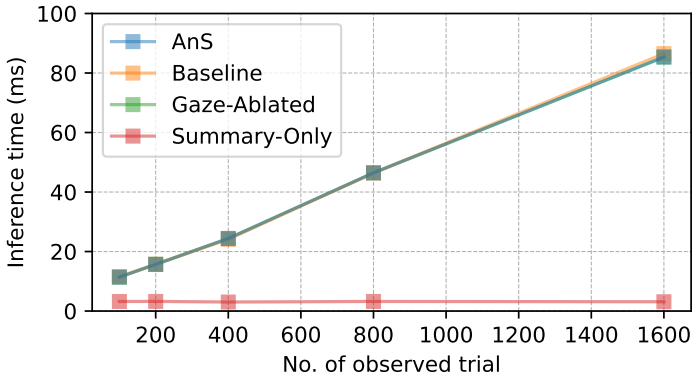

Figure C2: The inference time by the number of observed trials.

given (R^2=0.99). Summary-Only inference engine showed consistent inference time, approximately 3.1 ms. The inference time linearly increases when the trajectory part in the observed trial exists (i.e., processing trajectory data in the Perceiver is a bottleneck). We visualize the results in Figure C2.

References

Accot, J., Zhai, S., 1997. Beyond fitts' law: models for trajectory-based hci tasks, in: Proceedings of the ACM SIGCHI Conference on Human factors in computing systems, pp. 295–302.

Acharya, A., Chen, X., Myers, C.W., Lewis, R.L., Howes, A., 2017. Human visual search as a deep reinforcement learning solution to a pomdp., in: CogSci, pp. 51–56.

Allard, F., Graham, S., Paarsalu, M.E., 1980. Perception in sport: Basketball. Journal of sport and exercise psychology 2, 14–21.

Allard, F., Starkes, J.L., 1980. Perception in sport: Volleyball. Journal of sport and exercise psychology 2, 22–33.

Angel, R.W., 1976. Efference copy in the control of movement. Neurology 26, 1164–1164.

Ashby, N.J., Gonzalez, C., 2017. The influence of time estimation and time-saving preferences on learning to make temporally dependent decisions from experience. Journal of Behavioral Decision Making 30, 807–818.

Avidan, G., Harel, M., Hendler, T., Ben-Bashat, D., Zohary, E., Malach, R., 2002. Contrast sensitivity in human visual areas and its relationship to object recognition. Journal of neurophysiology 87, 3102–3116.

Bahill, A.T., Clark, M.R., Stark, L., 1975. The main sequence, a tool for studying human eye movements. Mathematical biosciences 24, 191–204.

Banovic, N., Grossman, T., Fitzmaurice, G., 2013. The effect of time-based cost of error in target-directed pointing tasks, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1373–1382.

van Beers, R.J., Baraduc, P., Wolpert, D.M., 2002. Role of uncertainty in sensorimotor control. Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences 357, 1137–1145.

Behaviour Interactive, 2016. Dead by daylight. Video game.

Berndt, D.J., Clifford, J., 1994. Using dynamic time warping to find patterns in time series, in: Proceedings of the 3rd international conference on knowledge discovery and data mining, pp. 359–370.

Blakemore, S.J., Goodbody, S.J., Wolpert, D.M., 1998. Predicting the consequences of our own actions: the role of sensorimotor context estimation. Journal of Neuroscience 18, 7511–7518.

Blizzard Entertainment, 2016. Overwatch. Video game.

Boucher, L., Stuphorn, V., Logan, G.D., Schall, J.D., Palmeri, T.J., 2007. Stopping eye and hand movements: are the processes independent? Perception & psychophysics 69, 785–801.

Boudaoud, B., Spjut, J., Kim, J., 2022. Mouse sensitivity in first-person targeting tasks, in: 2022 IEEE Conference on Games (CoG), IEEE.

Boudaoud, B., Spjut, J., Kim, J., 2023. Mouse sensitivity in first-person targeting tasks. IEEE Transactions on Games .

Bye, R.T., 2009. The BUMP model of response planning. Ph.D. thesis. Ph. D. Dissertation. Sydney, Australia: The University of New South Wales.

Bye, R.T., Neilson, P.D., 2008. The bump model of response planning: Variable horizon predictive control accounts for the speed–accuracy tradeoffs and velocity profiles of aimed movement. Human movement science 27, 771–798.

Cheema, N., Frey-Law, L.A., Naderi, K., Lehtinen, J., Slusallek, P., Hämäläinen, P., 2020. Predicting mid-air interaction movements and fatigue using deep reinforcement learning, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–13.

Chen, X., Bailly, G., Brumby, D.P., Oulasvirta, A., Howes, A., 2015. The emergence of interactive behavior: A model of rational menu search, in: Proceedings of the 33rd annual ACM conference on human factors in computing systems, pp. 4217–4226.

Csikszentmihalyi, M., Csikzentmihaly, M., 1990. Flow: The psychology of optimal experience. volume 1990. Harper & Row New York.

Dahl, M., Tryding, M., Heckler, A., Nyström, M., 2021. Quiet eye and computerized precision tasks in first-person shooter perspective esport games. Frontiers in psychology 12, 676591.

Diederik, K., Ba, J.A., 2015. A method for stochastic optimization. arxiv 2014. arXiv preprint arXiv:1412.6980 .

Do, S., Chang, M., Lee, B., 2021. A simulation model of intermittently controlled point-and-click behaviour, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–17.

Donovan, I., Saul, M.A., DeSimone, K., Listman, J.B., Mackey, W.E., Heeger, D.J., 2022. Assessment of human expertise in first-person shooter games. bioRxiv .

DreamHack, 2016. Dreamhack zowie open bucharest 2016. Esports Tournament. Virtus.pro (Winner) vs. Team Dignitas, Semi Final #2.

Duinkharjav, B., Chakravarthula, P., Brown, R., Patney, A., Sun, Q., 2022. Image features influence reaction time: A learned probabilistic perceptual model for saccade latency. ACM Transactions on Graphics (TOG) 41, 1–15.

Einhäuser, W., Nuthmann, A., 2016. Salient in space, salient in time: Fixation probability predicts fixation duration during natural scene viewing. Journal of Vision 16, 13–13.

Fischer, F., Bachinski, M., Klar, M., Fleig, A., Müller, J., 2021. Reinforcement learning control of a biomechanical model of the upper extremity. Scientific Reports 11, 14445.

Fitts, P.M., 1954. The information capacity of the human motor system in controlling the amplitude of movement. Journal of experimental psychology 47, 381.

Fleetwood, M.D., Byrne, M.D., 2006. Modeling the visual search of displays: a revised act-r model of icon search based on eye-tracking data. Human-Computer Interaction 21, 153–197.

Gajos, K., Weld, D.S., 2004. Supple: automatically generating user interfaces, in: Proceedings of the 9th international conference on Intelligent user interfaces, pp. 93–100.

Gershman, S.J., Horvitz, E.J., Tenenbaum, J.B., 2015. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. Science 349, 273–278.

Gibaldi, A., Sabatini, S.P., 2021. The saccade main sequence revised: A fast and repeatable tool for oculomotor analysis. Behavior Research Methods 53, 167–187.

Gibbon, J., Church, R.M., Meck, W.H., et al., 1984. Scalar timing in memory. Annals of the New York Academy of sciences 423, 52–77.

Glöckler, M., Deistler, M., Macke, J.H., 2017. Sgdr: Stochastic gradient descent with warm restarts, in: International Conference on Learning Representations.

Gonzalez, E.J., Chase, E.D., Kotipalli, P., Follmer, S., 2022. A model predictive control approach for reach redirection in virtual reality, in: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, pp. 1–15.

Gonzalez, E.J., Follmer, S., 2023. Sensorimotor simulation of redirected reaching using stochastic optimal feedback control, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pp. 1–17.

Groß, J., Timmermann, L., Kujala, J., Dirks, M., Schmitz, F., Salmelin, R., Schnitzler, A., 2002. The neural basis of intermittent motor control in humans. Proceedings of the National Academy of Sciences 99, 2299–2302.

Haarnoja, T., Zhou, A., Abbeel, P., Levine, S., 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: International conference on machine learning, PMLR. pp. 1861–1870.

Hetzel, L., Dudley, J., Feit, A.M., Kristensson, P.O., 2021. Complex interaction as emergent behaviour: Simulating mid-air virtual keyboard typing using reinforcement learning. IEEE Transactions on Visualization and Computer Graphics 27, 4140–4149.

Horst, R., Zander, S.M., Dörner, R., 2021. Cs: Show–an interactive visual analysis tool for first-person shooter esports match data, in: International Conference on Entertainment Computing, Springer. pp. 15–27.

Hultsch, D.F., MacDonald, S.W., Dixon, R.A., 2002. Variability in reaction time performance of younger and older adults. The Journals of Gerontology Series B: Psychological Sciences and Social Sciences 57, P101–P115.

Hussain, Z., Svensson, C.M., Besle, J., Webb, B.S., Barrett, B.T., McGraw, P.V., 2015. Estimation of cortical magnification from positional error in normally sighted and amblyopic subjects. Journal of Vision 15, 25–25.

Ikkala, A., Fischer, F., Klar, M., Bachinski, M., Fleig, A., Howes, A., Hämäläinen, P., Müller, J., Murray-Smith, R., Oulasvirta, A., 2022. Breathing life into biomechanical user models, in: Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology, pp. 1–14.

Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., Carreira, J., 2021. Perceiver: General perception with iterative attention, in: International conference on machine learning, PMLR. pp. 4651–4664.

Jiang, J., Shen, Y., Neilson, P.D., 2002. A simulation study of the degrees of freedom of movement in reaching and grasping. Human Movement Science 21, 881–904.

Jogan, M., Stocker, A.A., 2015. Signal integration in human visual speed perception. Journal of Neuroscience 35, 9381–9390.

Jokinen, J., Acharya, A., Uzair, M., Jiang, X., Oulasvirta, A., 2021. Touchscreen typing as optimal supervisory control, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–14.

Jokinen, J.P., Sarcar, S., Oulasvirta, A., Silpasuwanchai, C., Wang, Z., Ren, X., 2017. Modelling learning of new keyboard layouts, in: Proceedings of the 2017 CHI conference on human factors in computing systems, pp. 4203–4215.

Kang, D., Kim, N., Kang, D., Yoon, J.S., Kim, S., Lee, B., 2024. Quantifying wrist-aiming habits with a dual-sensor mouse: Implications for player performance and workload, in: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, pp. 1–18.

Kempka, M., Wydmuch, M., Runc, G., Toczek, J., Jaśkowski, W., 2016. Vizdoom: A doom-based ai research platform for visual reinforcement learning, in: 2016 IEEE conference on computational intelligence and games (CIG), IEEE. pp. 1–8.

Kim, S., Lee, B., Oulasvirta, A., 2018. Impact activation improves rapid button pressing, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–8.

Kim, S., Lee, B., Van Gemert, T., Oulasvirta, A., 2020. Optimal sensor position for a computer mouse, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–13.

Klapproth, F., 2008. Time and decision making in humans. Cognitive, Affective, & Behavioral Neuroscience 8, 509–524.

Koposov, D., Semenova, M., Somov, A., Lange, A., Stepanov, A., Burnaev, E., 2020. Analysis of the reaction time of esports players through the gaze tracking and personality trait, in: 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), IEEE. pp. 1560–1565.

Kowler, E., 1990. The role of visual and cognitive processes in the control of eye movement. Reviews of oculomotor research 4, 1–70.

Krakowczyk, D.G., Reich, D.R., Chwastek, J., Jakobi, D.N., Prasse, P., Süss, A., Turuta, O., Kasprowski, P., Jäger, L.A., 2023. pymovements: A python package for processing eye movement data, in: 2023 Symposium on Eye Tracking Research and Applications, Association for Computing Machinery, New York, NY, USA. URL: https://doi.org/10.1145/3588015.3590134, doi:10.1145/3588015.3590134.

Kukla, A., 1972. Foundations of an attributional theory of performance. Psychological review 79, 454.

Lamers James, R.G., O'Connor, A.R., 2023. Impact of focus of attention on aiming performance in the first-person shooter videogame aim lab. PLoS one 18, e0288937.

Lee, B., 2022. Cue integration in input performance. Bayesian Methods for Interaction and Design , 287.

Lee, B., Bang, H., 2015. A mouse with two optical sensors that eliminates coordinate disturbance during skilled strokes. Human–Computer Interaction 30, 122–155.

Lee, B., Kim, S., Oulasvirta, A., Lee, J.I., Park, E., 2018. Moving target selection: A cue integration model, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–12.

Lee, B., Nancel, M., Kim, S., Oulasvirta, A., 2020. Autogain: Gain function adaptation with submovement efficiency optimization, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–12.

Lee, B., Oulasvirta, A., 2016. Modelling error rates in temporal pointing, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 1857–1868.

Lee, D., Kim, S., Noh, J., Lee, B., 2024a. User performance in consecutive temporal pointing: An exploratory study, in: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, pp. 1–15.

Lee, H., Lee, S., Nallapati, R., Uh, Y., Lee, B., 2024b. Characterizing and quantifying expert input behavior in league of legends, in: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, pp. 1–21.

Lee, I., Kim, H., Lee, B., 2021. Automated playtesting with a cognitive model of sensorimotor coordination, in: Proceedings of the 29th ACM International Conference on Multimedia, pp. 4920–4929.

Lee, I., Kim, S., Lee, B., 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, pp. 1–12.

Lewis, R.L., Howes, A., Singh, S., 2014. Computational rationality: Linking mechanism and behavior through bounded utility maximization. Topics in cognitive science 6, 279–311.

Lin, R.F., Tsai, Y.C., 2015. The use of ballistic movement as an additional method to assess performance of computer mice. International Journal of Industrial Ergonomics 45, 71–81.

Loftus, G.R., Mackworth, N.H., 1978. Cognitive determinants of fixation location during picture viewing. Journal of Experimental Psychology: Human perception and performance 4, 565.

Looser, J., Cockburn, A., Savage, J., 2005. On the validity of using first-person shooters for fitts' law studies. People and Computers XIX 2, 33–36.

Lukanov, H., König, P., Pipa, G., 2021. Biologically inspired deep learning model for efficient foveal-peripheral vision. Frontiers in Computational Neuroscience 15, 746204.

MacKenzie, I.S., 1989. A note on the information-theoretic basis for fitts' law. Journal of motor behavior 21, 323–330.

MacKenzie, I.S., 1992. Fitts' law as a research and design tool in human-computer interaction. Human-computer interaction 7, 91–139.

Martín, J.A.Á., Gollee, H., Müller, J., Murray-Smith, R., 2021. Intermittent control as a model of mouse movements. ACM Transactions on Computer-Human Interaction (TOCHI) 28, 1–46.

Matin, E., 1974. Saccadic suppression: a review and an analysis. Psychological bulletin 81, 899.

Meta, T., 2018. Kovaak's.

Moon, H.S., Do, S., Kim, W., Seo, J., Chang, M., Lee, B., 2022. Speeding up inference with user simulators through policy modulation .

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

Moon, H.S., Oulasvirta, A., Lee, B., 2023. Amortized inference with user simulations, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pp. 1–20.

Müller, J., Oulasvirta, A., Murray-Smith, R., 2017. Control theoretic models of pointing. ACM Transactions on Computer-Human Interaction (TOCHI) 24, 1–36.

Munichor, N., Erev, I., Lotem, A., 2006. Risk attitude in small timesaving decisions. Journal of experimental psychology: applied 12, 129.

Neilson, P.D., Neilson, M.D., 2005. An overview of adaptive model theory: solving the problems of redundancy, resources, and nonlinear interactions in human movement control. Journal of neural engineering 2, S279.

Newzoo, 2022. Global esports & live streaming market report 2022. Newzoo Esports. Available online at: https://newzoo.com/insights/trend-reports/newzoo-global-esports-live-streaming-market-report-2022-free-version (accessed on 9 January 2023) .

Oulasvirta, A., 2017. User interface design with combinatorial optimization. Computer 50, 40–47.

Oulasvirta, A., Jokinen, J.P., Howes, A., 2022. Computational rationality as a theory of interaction, in: CHI Conference on Human Factors in Computing Systems, pp. 1–14.

Oulasvirta, A., Kim, S., Lee, B., 2018. Neuromechanics of a button press, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–13.

Park, E., Lee, B., 2020. An intermittent click planning model, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–13.

Park, E., Lee, S., Ham, A., Choi, M., Kim, S., Lee, B., 2021. Secrets of gosu: Understanding physical combat skills of professional players in first-person shooters, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–14.

ProGames Studio, 2016. Aim hero. URL: `https://store.steampowered.com/app/518030/Aim_Hero/`.

PUBG Corporation, 2017. Playerunknown's battlegrounds. Video game.

Rakitin, B.C., Gibbon, J., Penney, T.B., Malapani, C., Hinton, S.C., Meck, W.H., 1998. Scalar expectancy theory and peak-interval timing in humans. Journal of Experimental Psychology: Animal Behavior Processes 24, 15.

Ratcliff, R., 1978. A theory of memory retrieval. Psychological review 85, 59.

Rawat, A.S., 2021. Most popular cs:go crosshairs of 2021 - settings, statistics, detailed guide - afkgaming.com. `https://afkgaming.com/csgo/guide/8045-most-popular-csgo-crosshairs-of-2021-settings-statistics-detailed-guide`.

Rerick, M.A., Moritz, S.E., 2023. Coaches as teachers and facilitators of esports imagery use. Journal of Imagery Research in Sport and Physical Activity 18, 20230013.

Riot Games, 2020. Valorant. Video game.

Rogers, E.J., Trotter, M.G., Johnson, D., Desbrow, B., King, N., 2024. Kovaak's aim trainer as a reliable metrics platform for assessing shooting proficiency in esports players: a pilot study. Frontiers in Sports and Active Living 6, 1309991.

Roldan, C.J., Prasetyo, Y.T., 2021. Evaluating the effects of aim lab training on filipino valorant players' shooting accuracy, in: 2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA), pp. 465–470. doi:10.1109/ICIEA52957.2021.9436822.

Rosenholtz, R., Huang, J., Raj, A., Balas, B.J., Ilie, L., 2012. A summary statistic representation in peripheral vision explains visual search. Journal of vision 12, 14–14.

Salvucci, D.D., 2001. An integrated model of eye movements and visual encoding. Cognitive Systems Research 1, 201–220.

Sarcar, S., Jokinen, J.P., Oulasvirta, A., Wang, Z., Silpasuwanchai, C., Ren, X., 2018. Ability-based optimization of touchscreen interactions. IEEE Pervasive Computing 17, 15–26.

Saul, K.R., Hu, X., Goehler, C.M., Vidt, M.E., Daly, M., Velisar, A., Murray, W.M., 2015. Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. Computer methods in biomechanics and biomedical engineering 18, 1445–1458.

Schmidt, R.A., Zelaznik, H., Hawkins, B., Frank, J.S., Quinn Jr, J.T., 1979. Motor-output variability: a theory for the accuracy of rapid motor acts. Psychological review 86, 415.

Seow, S.C., 2005. Information theoretic models of hci: A comparison of the hick-hyman law and fitts' law. Human-computer interaction 20, 315–352.

Shen, Z., Xue, C., Li, J., Zhou, X., 2015. Effect of icon density and color contrast on users' visual perception in human computer interaction, in: Engineering Psychology and Cognitive Ergonomics: 12th International Conference, EPCE 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings 12, Springer. pp. 66–76.

Simon, H.A., 1990. Bounded rationality, in: Utility and probability. Springer, pp. 15–18.

Smith, M.C., 1967. Theories of the psychological refractory period. Psychological bulletin 67, 202.

Sparrow, W.A., Newell, K.M., 1998. Metabolic energy expenditure and the regulation of movement economy. Psychonomic Bulletin & Review 5, 173–196.

Spjut, J., Boudaoud, B., Binaee, K., Majercik, Z., McGuire, M., Kim, J., 2022. Firstpersonscience: Quantifying psychophysics for first person shooter tasks. arXiv preprint arXiv:2202.06429 .

Statespace, 2018. Aim lab. URL: `https://aimlab.gg/`.

Steelseries, 2021. 3d aim trainer. URL: `https://www.3daimtrainer.com/`.

Stocker, A.A., Simoncelli, E.P., 2006. Noise characteristics and prior expectations in human visual speed perception. Nature neuroscience 9, 578–585.

Strasburger, H., Rentschler, I., Jüttner, M., 2011. Peripheral vision and pattern recognition: A review. Journal of vision 11, 13–13.

Todorov, E., Erez, T., Tassa, Y., 2012. Mujoco: A physics engine for model-based control, in: 2012 IEEE/RSJ international conference on intelligent robots and systems, IEEE. pp. 5026–5033.

Vintsyuk, T.K., 1968. Speech discrimination by dynamic programming. Cybernetics 4, 52–57.

Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., et al., 2019. Grandmaster level in starcraft ii using multi-agent reinforcement learning. Nature 575, 350–354.

Wang, M., Zhao, H., Zhou, X., Ren, X., Bi, X., 2021. Variance and distribution models for steering tasks, in: The 34th Annual ACM Symposium on User Interface Software and Technology, pp. 1122–1143.

Wang, Z., Li, Q., 2007. Video quality assessment using a statistical model of human visual speed perception. JOSA A 24, B61–B69.

Warburton, M., Campagnoli, C., Mon-Williams, M., Mushtaq, F., Morehead, J.R., 2023. Kinematic markers of skill in first-person shooter video games. PNAS nexus 2, pgad249.

Watson, B., Spjut, J., Kim, J., Lee, B., Yoo, M., Shirley, P., Raymond, R., 2024. Is less more? rendering for esports. IEEE Computer Graphics and Applications 44, 110–116.

Weiner, B., 1985. An attributional theory of achievement motivation and emotion. Psychological review 92, 548.

Wells-Gray, E., Choi, S., Bries, A., Doble, N., 2016. Variation in rod and cone density from the fovea to the mid-periphery in healthy human retinas using adaptive optics scanning laser ophthalmoscopy. Eye 30, 1135–1143.

Wing, A.M., Kristofferson, A., 1973a. The timing of interresponse intervals. Perception & Psychophysics 13, 455–460.

Wing, A.M., Kristofferson, A.B., 1973b. Response delays and the timing of discrete motor responses. Perception & Psychophysics 14, 5–12.

Wobbrock, J.O., Cutrell, E., Harada, S., MacKenzie, I.S., 2008. An error model for pointing based on fitts' law, in: Proceedings of the SIGCHI conference on human factors in computing systems, pp. 1613–1622.

Ziv, G., Lidor, R., Levin, O., 2022. Reaction time and working memory in gamers and non-gamers. Scientific Reports 12, 6798.