N-ary Gaussian Model Modeling Pointing Uncertainty Across Task Scenarios Using an Automated Multi-Gaussian Modeling Pipeline

Hao Zhang
Institute of Software
Chinese Academy of Sciences
Beijing, China
zhanghao2023@iscas.ac.cn

Yixiao Xiao*
Institute of Software, Chinese Academy of Sciences
Beijing, Beijing, China
School of Computer Science and Technology, University of Chinese Academy of Sciences
Beijing, Beijing, China
xiaoyixiao24@mails.ucas.ac.cn

Yawen Zheng
Institute of Software Chinese Academy of Sciences
Beijing, China
zhengyawen@iscas.ac.cn

Jin Huang†
Institute of Software
Chinese Academy of Sciences
Beijing, China
huangjin@iscas.ac.cn

Xinan Yan
School of Advanced Technology
Xi'an Jiaotong-Liverpool University
Suzhou, Jiangsu, China
Institute of Software, Chinese Academy of Sciences
Beijing, China
xinan.yan22@student.xjtlu.edu.cn

Xuning Hu
Department of Computing
Xi'an Jiaotong-Liverpool University
Suzhou, China
xuning.hu22@student.xjtlu.edu.cn

Nianlong Li
Institute of Software, Chinese Academy of Sciences
Beijing, China
linianlong16@mails.ucas.ac.cn

Huawei Tu
Department of Computer Science and Information Technology
La Trobe University
Melbourne, Australia
h.tu@latrobe.edu.au

Feng Tian‡
Institute of software, Chinese Academy of Sciences
Beijing, China
tianfeng@iscas.ac.cn

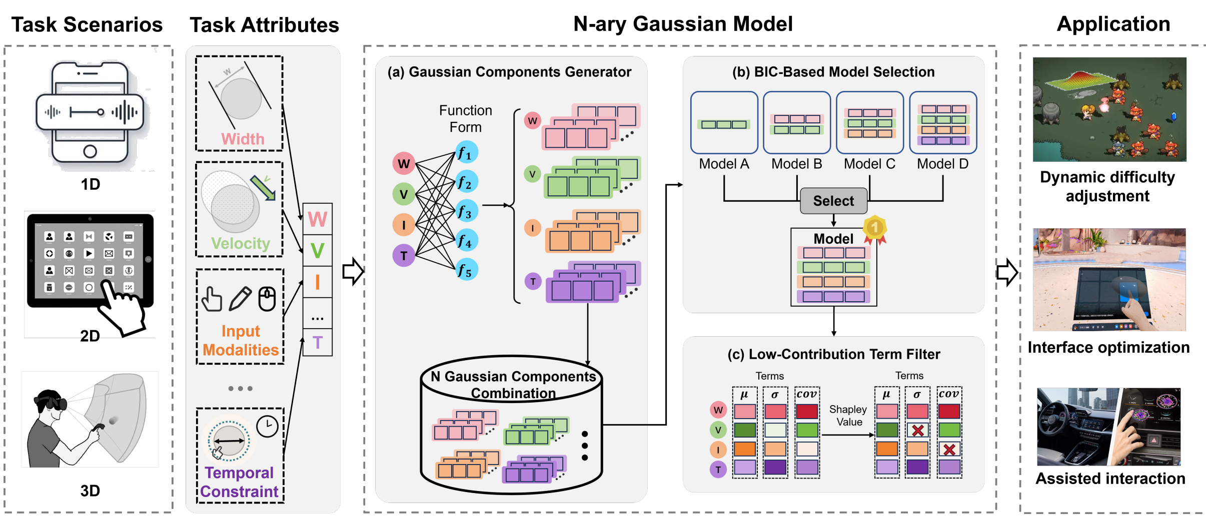


Figure 1: Overview of the N-ary Gaussian Model. (a) Generating forms for each Gaussian component by mapping task attributes to functional forms. (b) Selecting the optimal overall model using the Bayesian Information Criterion (BIC), balancing model complexity and accuracy. (c) Reducing model complexity by filtering out low-impact terms while retaining key terms, ensuring simplicity without sacrificing predictive power.

Abstract

This paper presents an N-ary Gaussian Model for predicting endpoint distributions in pointing tasks across task scenarios. Built on the foundational principles of the Ternary Gaussian model series, our model framework allows researchers to define parameter constraints and automatically refine model combinations, eliminating the need for predefined equations based on data analysis. We utilize the Bayesian Information Criterion (BIC) for model selection, ensuring simplicity while maintaining predictive accuracy. We conducted a comparative analysis against published baselines across 7 diverse datasets, covering 1D, 2D, and 3D tasks, different input modalities, different display devices, and time-constrained scenarios, demonstrating the robustness and generalization of the N-ary Gaussian Model. The N-ary Gaussion model offers an automated solution for modeling pointing uncertainty, and also incorporates cross output device, input modality, and temporal constraint factors into spatial pointing uncertainty modeling for the first time.

CCS Concepts

• **Human-centered computing → HCI theory, concepts and models**.

Keywords

Static Target Selection, Moving Target Selection, Spatial-Temporal Target Selection, Pointing Uncertainty, Models, Endpoint Distribution

ACM Reference Format:

Hao Zhang, Yixiao Xiao, Yawen Zheng, Jin Huang, Xinan Yan, Xuning Hu, Nianlong Li, Huawei Tu, and Feng Tian. 2026. N-ary Gaussian Model Modeling Pointing Uncertainty Across Task Scenarios Using an Automated Multi-Gaussian Modeling Pipeline. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3772318.3791672

1 Introduction

Human-computer interaction (HCI) relies on a wide range of interaction techniques that enable users to operate digital systems effectively. These techniques serve as the foundation for tasks ranging from everyday smartphone use to complex immersive applications. Among them, target selection tasks are a fundamental form of interaction in HCI, and play a critical role across diverse interactive systems, including mobile devices [47], desktop applications [29], and games [15]. These tasks, such as button clicking [45], tracking moving objects in video surveillance systems [27], and quickly targeting in first-person shooter (FPS) games [9], are influenced by multiple factors, including target size [6], velocity [26], and time urgency [51]. Understanding the factors that influence pointing uncertainty—how endpoint distributions change when users repeatedly select a target—can help predict key performance metrics like selection error rate (ER) [26, 27] and movement time (MT) [5, 65], which are crucial for assessing user performance. Despite the prevalence of target selection tasks, current research lacks a comprehensive model that integrates the diverse influences of these factors on endpoint distribution across different task settings. Developing such a model would provide valuable insights into user behavior and performance, allowing for more accurate predictions of outcomes in various scenarios.

Existing endpoint distribution models in HCI can be broadly categorized into two types: black-box models [39] and white-box models [27]. Black-box models often lack interpretability, making it challenging to understand the underlying mechanisms influencing endpoint distributions. Moreover, these models can be computationally intensive and typically require large amounts of data to achieve high accuracy [13], which limits their practicality in many HCI applications. In contrast, white-box models are widely used in HCI because of their interpretability and transparency. Among white-box models, regression models allow for detailed examination against both practical and theoretical assumptions, making them highly valuable in research and application [44]. They enable rapid computation of predictions, which is advantageous for real-time applications like adaptive interfaces and interface optimization [44]. However, building these models often involves significant limitations, such as prolonged modeling cycles that typically start with extensive data analysis to formulate hypotheses [27], followed by constructing the model and validating its performance. This iterative process can be time-consuming and resource-intensive, especially when each stage requires careful adjustment and fine-tuning to ensure the model's generalizability across various conditions.

This paper aims to develop an easily extendable automated regression model that does not require manual predefinition of its form when introducing new factors. Building such an automated model for endpoint distribution faces three primary challenges: 1) Establish a robust framework that underpins the model. The framework must accommodate varying conditions and also be capable of capturing the common characteristics of user behavior across diverse scenarios, providing a comprehensive view of endpoint behaviors. 2) While adding more parameters can improve a model's predictive power, it often increases complexity and diminishes simplicity. Therefore, it is crucial to balance complexity and parsimony by focusing on parameters that significantly enhance performance and excluding those that do not contribute meaningfully. 3) Beyond reflecting positive and negative effects through

*Yixiao Xiao is also with the School of Computer Science and Technology, University of Chinese Academy of Sciences.

†Corresponding authors.

‡Feng Tian is also with the School of Artificial Intelligence, University of Chinese Academy of Sciences.

CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/26/04

https://doi.org/10.1145/3772318.3791672

its structure, a model should provide insights into how each task parameter contributes to pointing uncertainty.

To address these challenges, we propose the N-ary Gaussian Model to describe endpoint distributions in generalizable target selection tasks. The N-ary Gaussian Model is built upon the multi-Gaussian framework established by a series of prior works (e.g., Dual and Ternary Gaussian models [6, 27]), which represent pointing uncertainty as a sum of multiple Gaussian components. Each Gaussian component corresponds to a specific task attribute (e.g., target size or speed), allowing the framework to provide clear explanatory insights into how different task attributes jointly influence user performance. However, similar to other white-box models, constructing models within this framework is constrained by the need for extensive task analysis conducted by domain experts. Our approach address this limitation by automatically generating and optimizing mathematical expressions related to task attribute. The overview of the N-ary Gaussian Model is shown in Fig. 1. The N-ary Gaussian Model first generates mathematical expressions to represents effects of any potential task attribute (Fig. 1 (a)). To navigate the trade-off between model complexity and performance, we adopt a dual selection strategy using the Bayesian Information Criterion (BIC) [53] and Shapley value analysis [57] (Fig. 1 (b) (c)). BIC penalizes over-parameterization, while Shapley values quantify the contribution of each term, enabling both model simplification and interoperability. For evaluation, we compared the generated models to those reported in the HCI literature. The datasets varied across several dimensions, including transitioning from 1D, 2D to 3D spaces, static to dynamic, and even spatial-temporal target selection. Additionally, partial datasets accounted for various factors, such as different input methods and display devices. Our method outperformed baseline models on 7 datasets, with an average reduction of 28.759% in Mean Wasserstein Distance (MWD), and demonstrated comparable performance in the remaining cases. Beyond these performance improvements, our framework enables researchers and designers to explore the modelability of new task attributes and assess their impact on user performance. Importantly, the approach retains interpretability while yielding actionable insights for predicting error rates and guiding UI design. Rather than replacing existing models, our approach offers a systematic, extensible, and accessible alternative for exploring and refining modeling spaces in HCI research.

In summary, our key contributions include:

• We propose the N-ary Gaussian Model, a pipeline that automates endpoint distribution modeling through structure search and joint selection using BIC and Shapley values, reducing reliance on manual model design.

• We validate the model across 7 diverse datasets, covering 1D, 2D, and 3D tasks, different input modalities, different display devices, and time-constrained scenarios. Results show that our model outperforms white-box baselines and achieves competitive accuracy to black-box models with significantly fewer parameters.

• Across 7 diverse datasets, we summarize the findings of the N-ary Gaussian pipeline, which capture the influence of output display (PPI), input modality (DHR), and temporal constraints (t_c) on pointing uncertainty, and we further illustrate the extensibility and potential application of the model across diverse interaction scenarios.

2 Related Work

This section reviews related work and summarizes them into three categories: models for target selection, multi-Gaussian models, and automated modeling attempts in HCI.

2.1 Models for Target Selection

Fitts' Law is one of the most widely used probabilistic models in HCI for predicting human performance in pointing and selection tasks [41, 46]. The law quantifies and predicts movement time (MT) by considering two key parameters: the width of the target and the distance between the starting position and the target, effectively modeling the difficulty of selecting a target [33, 52]. It has been rigorously evaluated and validated across various task types [68] and input devices [20, 28, 49, 52], demonstrating its extensive applicability in diverse environments and interaction contexts. Although Fitts' Law provides valuable insights into user behavior during target selection, it falls short of accurately predicting selection accuracy and efficiency, particularly under varying conditions of endpoint uncertainty. Research into selection uncertainty has focused largely on the form and distribution of endpoints. Wobbrock et al. distinguished between constant errors, defined as the average distance from the touchpoint to the target center, and variable errors, which describe the spread or deviation of hits around the target [60]. Building on the parameters of Fitts' Law, they developed a model that accurately predicts error rates for pointing tasks involving both one-dimensional [58] and two-dimensional targets [7]. Similarly, the concept of endpoint distribution has been adapted into Fitts' Law by substituting target width with effective width, a metric derived from the mean and standard deviation of the endpoint distribution [50]. This modification enhances the model's ability to account for the variability inherent in human motor performance, offering a more nuanced approach to predicting user errors in selection tasks. Besides task-level models like Fitts' law, control theory-based models [24, 42] can offer a dynamic, closed-loop perspective to model and predict human pointing behavior, integrating system states, feedback mechanisms, and control objectives, thereby enabling a deeper understanding of interaction dynamics and informing design improvements.

In previous work on selecting moving targets, efforts have been made through modeling to predict selection time and error rates to better understand the difficulty and uncertainty associated with selecting moving targets. Jagacinski et al. [31] and Hoffmann [22] extended the original Fitts' Law to explore the relationship between velocity and the index of selection difficulty, introducing steady-state position error as a critical factor. Based on this, they further proposed models to predict movement time (MT) when selecting moving targets, thereby enhancing the foundational understanding of dynamic target acquisition processes. Complementing this, Huang et al. approached the problem from the perspective of error rate prediction, modeling endpoint uncertainty and developing one-dimensional and two-dimensional target selection models as well as crossing models for moving targets, which accurately predict error rates during the selection of moving targets [25–27]. Additionally, Lee et al. proposed a model for predicting error rates in selecting moving objects under delay conditions and suggested a method to compensate for the effects of delay by adjusting the geometric size of the target [37], further enriching the understanding of delay factors. Extensive research has also explored the modeling of selecting moving targets using mice and controller joysticks, revealing the performance and adaptability of these input devices in dynamic environments [14–16]. Collectively, these studies consistently demonstrate that, compared to static targets, the difficulty and error rates associated with selecting moving targets are significantly higher [21, 30]. This increase in difficulty and error rate is significantly influenced by target width and velocity, highlighting the crucial role these parameters play in the overall complexity of the task [18]. Notably, the initial distance between the cursor and the target has a markedly reduced impact across various dimensions, suggesting that initial positioning may have a lesser effect on performance than previously assumed. Furthermore, in input modalities such as mice and styluses, movement time and aiming precision are often considered secondary factors in the context of selecting moving targets, indicating that the dynamic nature of the task may overshadow these influences [26, 31]. These findings not only deepen our understanding of the selection process in dynamic environments but also provide a solid theoretical foundation for developing more precise predictive models, emphasizing the complex interplay between movement, target characteristics, and user input methods.

2.2 Multi-Gaussian Model

Our work is inspired by studies done on the multi-Gaussian models. The concept of using the sum of Multi-Gaussian components to represent the influences of various factors on endpoints was first introduced by Bi et al. in 2013. They developed this approach to model the endpoints of target selection with finger touch on smartphones [5, 6]. Their work proposed a dual Gaussian distribution hypothesis to explain the distribution patterns of endpoints for finger-based input. However, as HCI application scenarios became more complex, especially with the introduction of dynamic target selection, traditional models showed limitations in handling more intricate dynamic scenarios. To address this, Huang et al. [26] proposed a ternary Gaussian model, which incorporates motion and target size to account for motion uncertainty and explains the endpoint uncertainty in one-dimensional moving targets.

Building on the one-dimensional ternary Gaussian model, Huang et al. [27] proposed a two-dimensional ternary Gaussian model for 2D moving target selection tasks. This model aligns the x-axis with the target movement direction and the y-axis perpendicular to it, modeling endpoints separately on each axis. It extends the ternary Gaussian model to 2D space and offers insights for future research on 2D target selection. With the rise of new input methods such as pen, finger, and mid-air gestures [2, 40, 56, 63], crossing-based selection has gained attention for its natural adaptation to these methods and potential to improve user performance. Unlike pointing tasks, crossing-based selection involves crossing target boundaries. In such tasks, the influence of target movement direction on user selection cannot be ignored. To address this, and build upon the evolving use of Gaussian models, Huang et al. [25] introduced a quaternary Gaussian model that incorporates target movement direction to better quantify endpoint uncertainty. This model represents a significant step in extending Gaussian-based approaches to more dynamically complex selection tasks. Building on the previous exploration of Gaussian-based models for endpoint uncertainty in dynamic target selection, further research has also examined the impact of target shape. Grossman and Balakrishnan [19] developed a probabilistic model to investigate how target shape affects static target pointing performance, demonstrating a significant influence. Extending this to dynamic scenarios, Zhang et al. [69] introduced a method that combines a ternary Gaussian model with a Gaussian Mixture Model (GMM) to model the effect of target shape on endpoint uncertainty for moving targets. While this model improved fitting accuracy compared to earlier approaches, it required shape-specific training and did not account for the effects of target movement direction. To address these limitations, Zhang et al. [66] introduced a new model based on a dual Gaussian hypothesis. This model utilizes a Dual Decomposition (DUDE) algorithm to decompose target shapes and generate endpoint distributions using a simplified rectangular structure.

Considering tasks that involve both spatial and temporal constraints, such as timing a swing to hit a moving ball in tennis, Huang and Lee [23] combined a spatial ternary Gaussian model with a temporal model to predict error rates in spatial-temporal tasks. Their experimental setup demonstrated the model's ability to handle tasks with time constraints effectively. Building upon these studies, Adrian L. et al. [51] proposed a dynamic spatial-temporal model for aiming tasks, incorporating elements from both 1D/2D ternary Gaussian models and Lee et al.'s work [35] on integrating different types of temporal cues for reliable task estimation. This combined approach significantly improves prediction accuracy for user performance in complex tasks involving both spatial and temporal dynamics. However, in these spatial-temporal target selection tasks, researchers typically model time and space separately, without integrating the temporal effects into endpoint distribution models. This separation overlooks the potential interactions between time constraints and spatial accuracy, which may play a critical role in influencing user behavior under complex task conditions. By failing to incorporate temporal factors into the spatial distribution framework, these models may lack the capacity to fully capture the nuances of user performance in tasks where both spatial and temporal elements are intertwined.

2.3 Automated Modeling Attempts in HCI

In the pursuit of automated modeling techniques, researchers have long aimed to develop approaches capable of explaining and modeling user performance across diverse interaction tasks using a unified model form. In the field of HCI, Oulasvirta et al. [44] introduced an automated nonlinear regression method to model target selection times, grounded in Fitts' law. This method predefines the functional relationship between task and time, using a stochastic local search to iteratively construct the model. By defining constraints rather than equations, this approach provides flexibility in identifying the optimal model form without manual intervention. This automated modeling method has proven effective in improving model fit in seven out of 11 cases and directly inspired the work in this paper. We extend this approach to focus on endpoint distribution, aiming to build an automated modeling tool that captures the nuances of user-pointing behavior across diverse scenarios. In addition, Zhang et al. [67] proposed a model based on artificial potential fields for trajectory-based target selection tasks, adaptable to various path constraints. Their approach demonstrated high predictive accuracy for movement time, average trajectory, and movement uncertainty.

Additionally, the increasing use of Partially Observable Markov Decision Processes (POMDP) and Reinforcement Learning (RL) for simulating human behavior in interactive systems reflects a growing interest in task generalization [39]. These models are often employed to simulate decision-making under uncertainty, a key characteristic of human interaction tasks. For example, studies have applied POMDP and RL frameworks to model complex tasks such as menu selection [11], visual search [12], typing [32], pointing [10], and drawing [54]. These models excel at adapting to various task conditions and learning optimal strategies for user interaction. While powerful, these approaches are often opaque, requiring significant computational resources and interpretability trade-offs. To balance interpretability and performance, it is often necessary to incorporate domain knowledge into these models by formalizing key patterns or behaviors and embedding them within black-box frameworks [39]. However, such efforts require substantial expertise and understanding of the task domain, as well as careful consideration of how best to formalize these insights within the model.

3 N-ary Gaussian Model

We propose the N-ary Gaussian Model, to enable flexible and automated modeling of endpoint distributions in pointing tasks across scenarios. As mentioned earlier, the N-ary Gaussian Model is built upon the multi-Gaussian framework, which represents pointing uncertainty as a sum of multiple Gaussian components. By associating each Gaussian component with a specific task attribute, the framework can model both the pointing uncertainty related to each attribute and the combined influence of these attributes on the overall pointing uncertainty. While it retains the core idea of the multi-Gaussian framework, the N-ary Gaussian Model further introduces methods to generate and optimize the form and combination of the Gaussian components, allowing the final model of pointing uncertainty to be derived automatically.

This section begins by introducing the multi-Gaussian framework (Fig. 2), followed by the description of the other three components of the N-ary Gaussian Model that enable automated modeling: Gaussian components automatic generator (Fig. 2 (a)), BIC-based model selection (Fig. 2 (b)), and low-contribution term filter (Fig. 2 (c)).

3.1 Multi-Gaussian Framework

The multi-Gaussian framework emerged gradually as HCI researchers advanced modeling uncertainty in pointing tasks. Among the earliest works are the Dual Gaussian [6] and the subsequent series of Ternary Gaussian models [25–27]. The core idea of the multi-Gaussian framework is using the sum of multiple Gaussian components to represent influences of different factors on endpoints. This idea is first introduced by Bi et. al. in 2013 to model target selection endpoints with finger touch in smart phone [6]. They assumes that the endpoints of a 1D target selection task [1] is a random variable X following a normal distribution, and that X is the sum of two normally distributed random variables associated with absolute pointing precision X_a and target size X_s. Since both components are Gaussian random variables, their sum is also a Gaussian random variable:

$$X = X_a + X_s \sim N(\mu, \sigma^2) \tag{1}$$

Building on this concept, Huang et al. introduced the Ternary-Gaussian Model [26] to model pointing uncertainty for moving targets, by incorporating an additional Gaussian component X_m to represent pointing uncertainty corresponding to target motion:

$$X = X_a + X_m + X_s \sim N(\mu, \sigma^2) \tag{2}$$

Similarly, if we assume there are N factors (or task attributes) that may influence pointing uncertainty, such as shown in Fig. 3, we can use the sum of N Gaussian components to represent the N factors and formulate the overall pointing uncertainty as follows:

$$X = \textstyle\sum_{i=1}^{n} X_i \sim N(\mu, \sigma^2), \tag{3}$$

where X represents the random variable of the location of the endpoints, and it is the sum of n normally distributed random variables (i.e., Gaussian components) associated with different factors that could potentially affect the overall endpoint distribution (All endpoints were transformed into a local velocity coordinate system, with the transformation process detailed in the Appendix). Since all the n Gaussian components are Gaussian random variables, their sum X is also a Gaussian random variable.

As the output endpoint distribution of the multi-Gaussian framework is also Gaussian distribution, thus it is determined by the two Gaussian parameters of μ and σ. According to the theorem on the sum of Gaussian random variables, Gaussian parameter μ can be obtained by having the sum of the means of each Gaussian component μ_i:

$$\mu = \textstyle\sum_{i=1}^{n} \mu_i. \tag{4}$$

And σ^2 is the sum of the variances of each component σ_i^2, if the components are independent:

$$\sigma^2 = \textstyle\sum_{i=1}^{n} \sigma_i^2 \quad \text{or} \quad \sigma = \sqrt{\textstyle\sum_{i=1}^{n} \sigma_i^2}, \tag{5}$$

otherwise, σ^2 is influenced by the covariances between non-independent components:

$$\sigma^2 = \textstyle\sum_{i=1}^{n} \sigma_i^2 + \displaystyle\sum_{1 \le i < j \le n} Cov(X_i, X_j) \quad \text{or} \quad \sigma = \sqrt{\textstyle\sum_{i=1}^{n} \sigma_i^2 + \displaystyle\sum_{1 \le i < j \le n} Cov(X_i, X_j)}, \tag{6}$$

where, X_i and X_j are pairs of non-independent components.

[1] For ease of understanding, we will continue to use 1D static target selection task to introduce this framework in the following chapters.

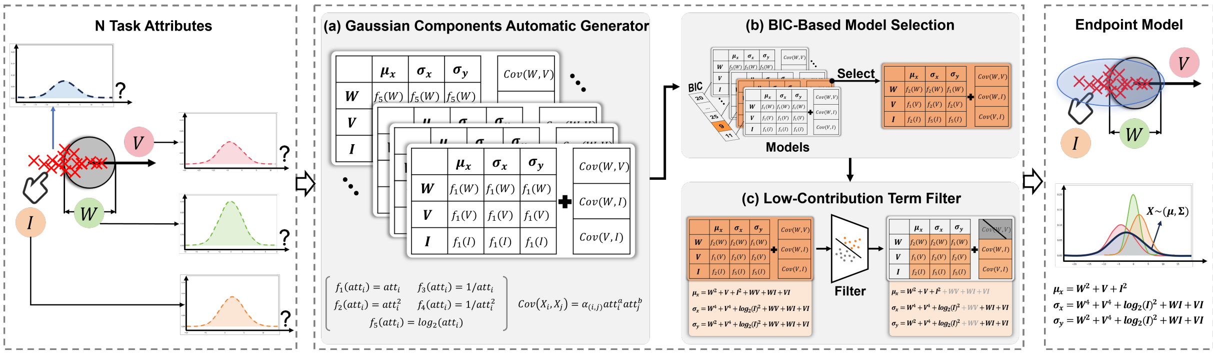


Figure 2: Workflow of the N-ary Gaussian Model. Three components were incorporated into the multi-Gaussian framework to enable the modeling of pointing tasks with any given N task attributes: (a) model term generation, where task attributes are mapped to Gaussian components and instantiated as specific terms; (b) parameter estimation and optimal model selection via BIC, which exhaustively evaluates candidate combinations of terms and retains the one with the lowest BIC; and (c) complexity reduction, where Shapley values quantify each term's contribution and low-contribution terms are filtered to yield a compact endpoint distribution model.

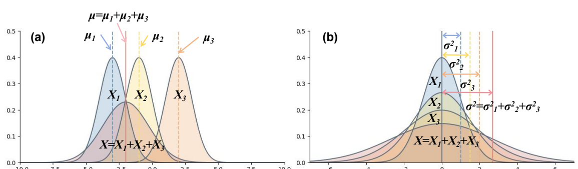


Figure 3: A visual representation of the cumulative impact of individual Gaussian components on the total random variable. (a) illustrates the cumulative effect on the mean (μ) of the total random variable. (b) focuses on the cumulative effect on the variability (σ).

We now elucidate the physical interpretation of the multi-Gaussian framework. The distribution of endpoints described by the multi-Gaussian framework is a random variable composed of the sum of n Gaussian components. Because the endpoints are determined as locations in a coordinate system centered on the target, the mean of the total random variable can be considered as the result of the bias produced by each Gaussian component on the endpoint distribution center, while the variability is the sum of the variabilities of each Gaussian component. Fig. 3 provides a visual representation of the cumulative impact of individual Gaussian components on the total random variable. Fig. 3 (a) illustrates the cumulative on the mean (μ), showing how the bias introduced by each Gaussian component shifts the center of the endpoint distribution. On the other hand, Fig. 3 (b) demonstrates the cumulative impact on the variability (σ), depicting how the individual variabilities from each Gaussian component combine to influence the overall distribution's spread.

Building on the multi-Gaussian framework for 1D target selection, we can extend it to 2D/3D target selection by applying the same formulation independently along each coordinate axis. For instance, the μ and Σ of the 2D N-ary Gaussian can be written as $\mu = [\mu_x, \mu_y]^T$, and $\Sigma = diag[\sigma_x, \sigma_y]$.

Note that the multi-Gaussian framework only specifies the combination method of the Gaussian components (i.e., the sum of Gaussian random variables) without defining the number of random variables or their expressions (i.e., functions of μ_i and σ_i). Therefore, when using the multi-Gaussian framework to address a specific problem, a set of methods—referred to as the automated multi-Gaussian modeling pipeline—is needed to select the appropriate Gaussian components for the model and determine the expression for each component, including any potential correlation coefficients between the components.

Specifically, the automated multi-Gaussian modeling pipeline consists of three steps: 1) Generating specific terms for each Gaussian component and the resulting overall models: deriving functional terms for each attribute and performing linear regression to estimate model parameters (Fig. 2 (a)); 2) Selecting the optimal overall model: applying the Bayesian Information Criterion (BIC) to select the optimal model form, balancing fit and complexity (Fig. 2 (b)); 3) Reducing the complexity of the optimal model: removing terms with minimal impact to simplify the model while retaining key terms(Fig. 2 (c)). We introduce these three steps in the following sections.

3.2 Gaussian Components Automatic Generator

The Gaussian Components Automatic Generator establishes a mapping from task attributes to Gaussian component parameters. Multi-Gaussian models generally assume that a certain Gaussian component corresponds to a specific task attribute. For example, the size component X_w corresponds to target width W, and the velocity component X_v corresponds to target velocity V [26]. In this work, for a Gaussian component X_i corresponding to a task attribute att_i, we define the parameters μ_i and σ_i of the Gaussian component as:

$$\begin{aligned} \mu_i &= f(att_i) \\ \sigma_i &= g(att_i), \end{aligned} \tag{7}$$

where f and g represent the mappings of a task attribute to μ and σ of a Gaussian component. There are several common methods

Table 1: The operations that map a task attribute to μ and σ of a Gaussian component in this paper

Function Name	Function Form
Linear	$f(att_i) = att_i$
Quadratic	$f(att_i) = att_i^2$
Reciprocal	$f(att_i) = 1/att_i$
Reciprocal Quadratic	$f(att_i) = 1/att_i^2$
Logarithmic	$f(att_i) = log_2(att_i)$

for constructing these mappings, which can involve both algebraic and transcendental operations. Algebraic operations typically include power functions with low exponents, while transcendental operations cover functions such as logarithmic [44]. Compared to Oulasvirta [44], we deliberately restrict the operation set in Table 1 to low-order polynomials. For the datasets and pointing tasks considered in this paper, existing analytical models of endpoint behavior are also formulated using such low-order, monotonic forms, rather than trigonometric or exponential expressions. To balance interpretability and computational efficiency, we currently implement 5 distinct operations, though the framework is flexible enough to accommodate additional operations. The operations (see Table 1) we employ include linear, quadratic, and their reciprocal forms and logarithmic functions. These operations involve single terms, i.e., each term does not independently consider the influence of constants.

As for $\mathrm{Cov}(X_i, X_j)$, we similarly quantify the interaction between two Gaussian components X_i and X_j on the endpoint distribution, by expressing their covariance as a function of the task attributes associated with these components as follows:

$$Cov(X_i, X_j) = \alpha_{(i,j)} att_i^a att_j^b, \quad a, b \in [-2, 0) \cup (0, 2], \quad a, b \in \mathbb{Z}, \tag{8}$$

where $\alpha_{(i,j)}$, a and b can be calculated based on the observed data. Using this power function model to describe the independent attributes and their interactive effects can reveal various interaction relationships, including nonlinear relationships, gain and loss effects, and positive and negative interaction effects.

At this stage, task attributes are mapped to Gaussian components and instantiated as specific terms. Given that these terms maintain a linear relationship with the parameters of the overall Gaussian distribution, multiple linear regression can be applied to determine the optimal combination of terms and their corresponding coefficients.

3.3 BIC-Based Model Selection

When applying regression models, selecting an appropriate model is essential for accurately capturing the relationships between variables and making reliable predictions. This selection process involves balancing model fit and complexity by incorporating penalties for models with a greater number of parameters. Given the diversity of available regression models, it is critical to employ robust model selection criteria, such as the Akaike Information Criterion (AIC) [3] and the Bayesian Information Criterion (BIC) [53]. Since BIC provides a more substantial penalty for model complexity than AIC, we choose BIC as the primary criterion for selecting the optimal regression model. In our pipeline, we use BIC as the primary objective for model selection: candidate formulations of μ_x and σ_x^2 are fitted to the data, their BIC values are computed, and the model with the lowest total BIC is retained as the optimal analytic structure.

The BIC is calculated using the following formula:

$$BIC = kln(m) - 2ln(L), \tag{9}$$

where $kln(m)$ is the penalty term, it increases with the number of parameters k and the number of observations m, it penalizes more complex models with a larger number of parameters, thereby discouraging overfitting; $-2ln(L)$ represents the log-likelihood of the model, it measures how well the model explains the observed data.

For linear regression, the residuals are assumed to be normally distributed, and the log-likelihood $ln(L)$ can be effectively estimated using the Residual Sum of Squares (RSS). This is given by:

$$BIC = mln(\frac{RSS}{m}) + kln(m), \tag{10}$$

where $RSS = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$ is the sum of squared residuals, y_i are the observed values, and $\hat{y}_i$ are the predicted values from the regression model. BIC allows for objectively comparing models based on their fit to the data while adjusting for the number of parameters.

3.4 Low-Contribution Term Filter

Following the process outlined above, we obtained the final form of the Gaussian model. To further control complexity, we then assess the relative contribution of each term using Shapley values. This term-level attribution allows us to identify terms that only marginally improve prediction accuracy and to filter them while preserving the overall quality of fit, resulting in a more compact model expression. Each term's Shapley value ϕ_i is calculated using the following formula:

$$\phi_i = \sum_{S \subseteq N \setminus i} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left[f(S \cup i) - f(S) \right], \tag{11}$$

where i is the term of the instance to be explained, N is the number of terms, S is a subset of terms excluding term i, $f(S)$ represents the model score obtained based on the term subset S, $f(S \cup i)$ is the model score obtained after adding term i to the subset S. For example, suppose there are two terms, $\{W, A\}$, so $N = 2$. To compute the Shapley value of W (i.e., $i = W$), all subsets S that do not contain W are $S \in \{\emptyset, \{A\}\}$. In this case, $f(\emptyset)$ is the score of a model that uses neither W nor A, $f(\{W\})$ is the score using only W, $f(\{A\})$ is the score using only A, and $f(\{W, A\})$ is the score using both terms. The marginal contribution of W for each subset is $f(\{W\}) - f(\emptyset) \mathrm{and} f(\{W, A\}) - f(\{A\})$,, and the Shapley value of W is the weighted average of these two marginal contributions.

The contribution ratio value of each term ($Ratio_i$) in the model is:

$$Ratio_i = \frac{\phi_i}{\sum_{j=1}^{|N|} \phi_j}. \tag{12}$$

By setting a threshold of 5%, we can eliminate terms with low contributions, thus maximizing the model's simplicity. However, the threshold determines the strictness level of low-contribution filtering and affects the model's final performance. When the threshold is set too low, it may fail to sufficiently filter out terms with low contributions. Conversely, if the threshold is set too high, it might result in the removal of terms that have a significant impact on the endpoint distribution. Empirically, we set it to 5%, but this value can be adjusted in practical applications.

3.5 Example: 1D Static Pointing Task

To make the procedure concrete, consider a 1D static pointing task in which the user starts from start button and has to acquire a circular target as quickly and accurately as possible. The target is placed at an amplitude (distance) A from the start button and has a width W. The horizontal endpoint X of each trial is modeled as $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$.

In our pipeline, we view μ_x and σ_x^2 as unknown analytic functions of W and A and construct them from a pool of simple terms such as W, A, W^2, A^{-1}, $\log_2 A$, and their interaction terms (e.g., W^2A^2, W^2A^{-1}). Each term corresponds to a width, amplitude, or interaction component, so that any candidate model is a linear combination (Detailed see Table. 5).

Given the observed endpoints distribution, we first treat the Gaussian parameters as unknown functions and let different analytic combinations of terms from the pool act as candidate formulas for μ_x and σ_x^2. Concretely, for each candidate pair of formulas, we fit their coefficients to the empirical mean and variance of the endpoints For every fitted candidate model, we then compute the BIC of the mean and variance using Eq. 10. The best model is chosen as the one whose analytic formulas for μ_x and σ_x^2 achieve the lowest BIC. Finally, we apply Shapley values to this BIC-optimal model to quantify the contribution of each selected term and prune terms with negligible contribution, yielding an expression for μ_x and σ_x^2. More details are provided in Appendix B, where we use concrete data and report the key steps of our pipeline.

4 Model Evaluation

We evaluate the method through a benchmark comparison with published models, examining its performance across various scenarios. In datasets where new attributes were introduced, we further investigated the model's ability to explain the influence of these attributes on endpoint distribution using Shapley values.

4.1 Dataset Description

The datasets shown in Table 2 cover traditional pointing tasks across various dimensions, including 1D, 2D, 3D, and different contexts (encompassing static, dynamic, and spatial-temporal target selection). In some datasets, we considered cross-display device usage (Dataset 2D-Static) and cross-input modalities (Dataset 2D-Moving). Most tasks involve 2-3 attributes, while more complex tasks incorporate 4-7 attributes (Dataset 1D-Moving-Temporal, Dataset 2D-Moving-Temporal, Dataset 3D-Moving). We recruited participants and collected data to form three datasets: Dataset 1D-Static, Dataset 2D-Static, Dataset 2D-Moving-Tempora. This study was conducted in accordance with the ethical guidelines of the local ethics committee. Prior to participation, all subjects provided informed consent, and their confidentiality and anonymity were strictly maintained throughout the study.

Dataset 1D-Static represents 1D static target selection tasks. It was collected through a customized experiment conducted on a desktop computer equipped with an Intel i7-7700 CPU, 16 GB of RAM, and an NVIDIA GTX 1060 6GB graphics card. A Samsung 23.6" monitor (1920 × 1080 resolution) was used for display. 12 participants (8 males, 4 females; aged 20–28 years, mean = 23.3, SD = 2.98) took part in the study. The experimental design included 4 target widths (W: 24, 48, 96, and 144 pixels) and 5 distances (A: 384, 480, 576, 672, and 768 pixels).

During the experiment, participants were first required to click on a white strip on the left side of the screen (48 pixels wide). A target strip then appeared on the right, with its width and distance determined by the experimental design. Participants were instructed to select the target strip as quickly and accurately as possible, following a similar approach as described by Bi et al. in their study [5]. Each condition was repeated 12 times, resulting in a total of 240 trials, presented in a randomized order. This resulted in a total of 2880 data points collected under randomized conditions.

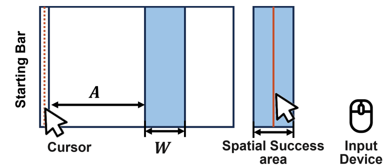
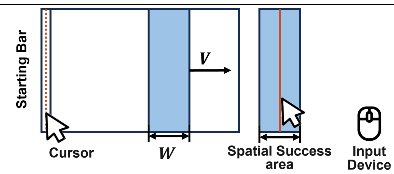
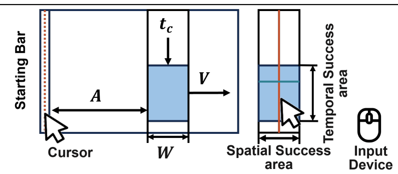
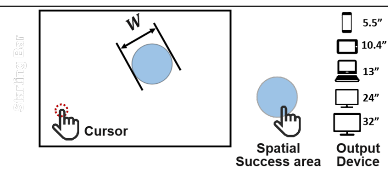
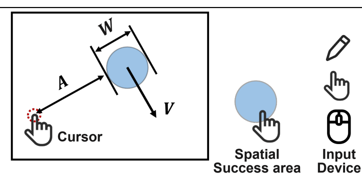
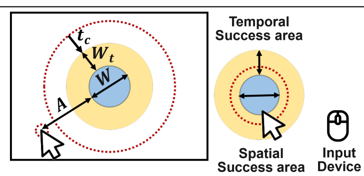
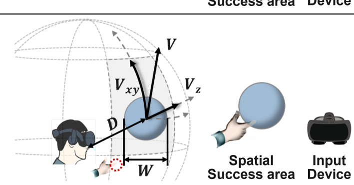
Dataset 1D-Moving represents 1D moving target selection tasks. This Dataset is derived from Huang et al. [27], in which the experiment was conducted on a Dell OptiPlex 9020 laptop computer, with an Intel Core i7 4 Quad core CPU at 3.6 GHz and a 23-inch (533.2 × 312mm) LED display at 1,920 × 1,080 resolution. The pointing device was a Dell MS111 mouse (1000 dpi). A total of 12 participants (6 males, 6 females; mean age = 27) × 4 target width (W: 24, 48, 96, and 144 pixels) × velocity (V: 96, 192, 288, and 384 pixels/second) 60 repeats resulted in 11520 endpoints.

Dataset 1D-Moving-Temporal represents 1D spatial-temporal target selection tasks. This Dataset is derived from Huang et al. [23]. The same hardware setup as Dataset 1D-Moving was used. As reported, a total of 12 participants × 2 distances (A: 384 and 768 pixels) × 3 target sizes (W: 24, 48 and 144 pixels) × 3 velocities (V: 96, 192 and 384 pixels/second) × 4 temporal widths (W_t: 0.08, 0.16, 0.24 and 0.32 second) × 3 temporal distances (t_c: 0.7, 1.1 and 1.5 second) × 5 repeats resulted in 12960 endpoints. In each trial, participants clicked the "start" button on the left side of the window. After a 1000 ms interval, a moving target appeared, and participants were asked to control the 1D cursor to acquire it. Each target could only be acquired once per trial.

Dataset 2D-Static represents 2D static target selection tasks. This Dataset was collected through a customized experiment using 5 output devices with varying screen sizes and resolutions: a Wacom Cintiq Pro 32" display (79.6 cm diagonal, 1920 × 1080 resolution, PPI = 70.29), [2] a Dell 24" monitor (60.4 cm diagonal, 1920 × 1080 resolution, PPI = 92.6), a Microsoft Surface Pro 13" tablet (32.8 cm diagonal, 2880 × 1920 resolution, PPI = 268.04), a Huawei MatePad 10.4" tablet (26.1 cm diagonal, 2000 × 1200 resolution, PPI = 226.98), and a ZTE 5.5" smartphone (14.6 cm diagonal, 1920 ×

[2]The pixel densities (PPI) of the devices, were calculated using the formula PPI = $\frac{\sqrt{W^2+H^2}}{O}$ where W and H are the screen resolution dimensions in pixels, and O is the screen size in inches. The same calculation method applies to the other devices mentioned.

Table 2: An illustration of the task setups of datasets, with the key task attributes (A: target amplitude, W: target width, V: target velocity, O: output display, I: input modalities W_t: temporal widths, t_c: temporal distances, D depth,W_p: perceived target width, V_{px}: perceived target velocity in the XOY plane, V_{pz}: perceived target velocity along z-axis), as well as successful selection cases.

Dataset Name	Task	Attribute	Dimension	Dynamic	Temporal
1D-Static		W, A	1D	⊗	⊗
1D-Moving [26]		W, V	1D	●	⊗
1D-Moving-Temporal [23]		A, W, V, W_t, t_c	1D	●	●
2D-Static		W, O	2D	⊗	⊗
2D-Moving [27]		W, V, I	2D	●	⊗
2D-Moving-Temporal		W, V, W_t, t_c	2D	●	●
3D-Moving [70]		W, V, V_x, V_z, D W_p, V_{px}, V_{pz}	3D	●	⊗

1080 resolution, PPI = 383.25). 16 participants (9 males and 7 females, aged 19 to 29, with an average age of 22.0 and a standard deviation of 2.32) took part in the study. The static target selection task included 4 different sizes of circular targets (72, 144, 216, and 288 pixels), as shown in Table 2.

During the experiment, participants performed a typical 2D static target selection task. Each trial began with a start button at the center of the screen; clicking it caused a circular target to appear. To mitigate potential learning effects, the target appeared at a random location on an invisible circle centered on the start button. The circle's radius was fixed at 0.375 times the shorter screen dimension. Participants were instructed to select the target as quickly and accurately as possible. Endpoint data were recorded for every trial, regardless of whether the target was successfully hit or missed. A trial was considered successful if the recorded endpoint fell within the target boundary. A total of 16 participants $\times$ 4 sizes(W) $\times$ 16 repeats $\times$ 5 output displays(O) = 5120 endpoints.

Dataset 2D-Moving represents 2D moving target selection tasks. This Dataset utilizes data from Huang et al. [27], in which the experiment was conducted on a HiteVision X7 interactive system with a built-in computer and a 70-inch LED display at a resolution of 1920×1080, supporting three input modalities. In the original study, separate models were built for each input modality(I), i.e., finger touch, stylus, and mouse. Dataset 2D-Moving, however, incorporates input modality as an attribute in the model. To quantify the impact of each modality, Berard et. al. [4] proposed using Device's Human Resolution (DHR), which suggests that mouse DHR typically ranges between 700-1400 dpi, pen devices range between 100-200 dpi and free-space devices, analogous to hand-touch interactions, range between 5-10 dpi. In addition to the original study's [27] reported mouse dpi of 1000, we used the average values reported in [4] to approximate the parameters of the input devices: mouse at 1000 dpi, stylus at 150 dpi, and finger touch at 7.5 dpi.

Dataset 2D-Moving-Temporal represents 2D spatial-temporal target selection tasks. This dataset was collected via a customized experiment involving 16 participants (9 males, 7 females; aged 20–33 years, mean age = 22.8, SD = 3.03). Target widths (W) were set at 3 levels (144, 24, and 48 pixels), and movement velocities (V) were set at 3 levels (384, 192, and 96 pixels/second). Instead of the temporal cue and bar-like temporal target prompts used in [51], we adopted a more 2D-appropriate shrinking cue form (see Table 2). A yellow region surrounding each target represented the time available for participants to select the target (W_t), with temporal widths set to 0.08, 0.16, 0.24, and 0.32 seconds. A red dished circle began shrinking at a constant rate from the outer edge of the target toward the yellow region, representing the duration from the appearance of the time circle to its arrival at the selection region. When the red dished circle reached the boundary of the yellow region, it marked the temporal distances (t_c) of 1.5, 1.1, and 0.7 seconds.

During the experiment, once participants pressed the start button, a blue target appeared randomly on a circle's boundary and moved in a randomly chosen direction at a constant velocity (see Table 2). Participants were instructed to select the blue target as quickly and accurately as possible while the red dots was within the yellow region. Selections made during this period were considered successful, whereas selections made before or after this window were considered failures. A total of 16 participants $\times$ 3 target sizes (W) $\times$ 3 velocities (V) $\times$ 4 temporal widths (W_t) $\times$ 3 temporal distances (t_c) $\times$ 5 repeats resulted in 8640 endpoints.

Dataset 3D-Moving represents 3D moving target selection tasks. This Dataset is derived from Zheng et al. [70]. In the original study, the 3D target motion was decomposed into two components: motion in the XOY plane (surface motion) and motion in depth. The combined motion condition integrates these two components, such that the target simultaneously varies in depth (D), polar angle, and azimuth angle, thereby approximating free motion in 3D space. The target surface velocities (V_x) were set to 0.2, 0.4, 0.6, and 0.8 m/s, while the depth velocities (V_z) were 0.10, 0.15, 0.20, and 0.25 m/s, resulting in overall velocities (V) of 0.22, 0.43, 0.63, and 0.84 m/s. Four target sizes (W: 0.02, 0.04, 0.06, and 0.08 m) and three depth levels (D: 0.3, 0.4, and 0.5 m) were included in the experimental design. In total, 15 participants $\times$ 4 target sizes (W) $\times$ 4 velocities (V) $\times$ 3 depths (D) $\times$ 12 repeats = 8,640 trials. The original study [70] also considered perceived target width (W_p) and perceived target velocity (V_p decomposed into V_{px} in the XOY plane and V_{pz} in depth) as perceptual attributes.

4.2 Comparing with White-box Models

The majority of samples in each dataset passed the normality test using Kolmogorov-Smirnov tests, with a confidence level of 95%. Following these tests, we assessed the performance of the proposed N-ary Gaussian Model against previously published baseline models in Human-Computer Interaction (HCI). The comparison was based on the Mean Wasserstein Distance (MWD) [1], which measures the average distance between actual and predicted distributions across various conditions. Smaller MWD values indicate a closer fit between the predicted and actual distributions, with values ranging from 0 to positive infinity. The N-ary Gaussian Model was evaluated across all datasets using the complete data for both the baseline and our model to ensure sufficient data points for regression. Cross-validation is applied in subsequent sections to validate the model's generalization capabilities. Note that previous studies separately modeled temporal and spatial endpoints for spatial-temporal target selection tasks (Dataset 1D-Moving-Temporal, Dataset 2D-Moving-Temporal). Consequently, the models employed in these datasets were the 1D [26] and 2D [27] Ternary Gaussian models.

The comparison results are provided in Table 3, where N represents the number of non-constant parameters in each model. The table also presents the detailed model forms for μ and σ, along with additional comparisons. Overall, in all the 7 datasets the N-ary Gaussian Model achieved a lower Mean Wasserstein Distance (MWD) compared to the corresponding baseline model. Overall, our model reduced MWD by an average of 6.074, with a standard deviation of 6.180. On average, our model achieved a 28.759% reduction in MWD compared to the baseline models. Notably, in datasets where new task attributes were introduced (e.g., Dataset 2D-Static with displays, Dataset 2D-Moving with input modalities, and Dataset 2D-Moving-Temporal with time constraints), our model automatically identified the optimal representation of these new attributes based on the Multi-Gaussian framework.

Across datasets, the number of parameters increased by an average of 1.714 (STD = 4.786) compared to the baseline, and the mean MWD decreased by 38.19%. The largest parameter increases occurred in Dataset 2D-Static and Dataset 2D-Moving, with six additional parameters each. These increases were justified by MWD reductions of 54.403% and 38.904%, respectively. In Dataset 3D-Moving, our model reduced the number of parameters by 8, yet still improved MWD by 17.940%. This suggests that, in scenarios with greater task complexity, it is possible to simplify the baseline model while still capturing the most critical attribute.

4.3 Error Rate Prediction

The ultimate metric for evaluating a pointing model in practical scenarios is often its ability to predict error rates [59]. This section assesses the practical utility of the N-ary Gaussian model and the baseline models on this critical task. We compute the predicted error

Table 3: N-ary Gaussian Model against previously published models in HCI. N refers to the number of parameters in the model. Results for the models that performed better are highlighted in bold.

Dataset Name	Term	Baseline						N-ary Gaussian Model					
		N	Term Format	$Adj.R^2$	MAE	Total N	MWD	N	Term Format	$Adj.R^2$	MAE	Total N	MWD
1D-Static	μ_x	1	α_0	-	2.628	3	1.129	2	$\alpha_0 + \alpha_1 W$	0.991	0.445	5	**0.523**
	σ_x^2	2	$\alpha_0 + \alpha_1 W^2$	0.958	4.977			3	$\alpha_0 + \alpha_1 W^4 + \alpha_2 [log_2(A)]^2$	0.989	3.613		
1D-Moving	μ_x	3	$\alpha_0 + \alpha_1 W + \alpha_2 V$	0.957	1.341	7	3.970	3	$\alpha_0 + \alpha_1 W + \alpha_2 V$	0.957	1.341	7	**2.615**
	σ_x^2	4	$\alpha_0 + \alpha_1 V^2 + \alpha_2 W^2 + \alpha_3 V/W$	0.971	39.628			4	$\alpha_0 + \alpha_1 W^2 + \alpha_2 V^2 + \alpha_3 WV$	0.988	25.887		
2D-Moving-Temporal	μ_x	3	$\alpha_0 + \alpha_1 W + \alpha_2 V$	0.976	3.307	7	2.501	3	$\alpha_0 + \alpha_1 W + \alpha_2 V$	0.976	3.307	10	**1.341**
	σ_x^2	4	$\alpha_0 + \alpha_1 W^2 + \alpha_2 V^2 + \alpha_3 V/W$	0.670	149.917			7	$\alpha_0 + \alpha_1 W^2 + \alpha_2 V^2 + \alpha_3 V/t_c + \alpha_4 WV + \alpha_5 A^2 + \alpha_6 W_t^2$	0.814	117.422		
2D-Static	μ_x	1	α_0	-	3.180	5	6.972	3	$\alpha_0 + \alpha_1 W + \alpha_2 1/O$	0.895	0.997	11	**3.179**
	σ_x^2	2	$\alpha_0 + \alpha_1 W^2$	0.884	136.567			4	$\alpha_0 + \alpha_1 W^2 + \alpha_2 1/O^2 + \alpha_3 WO$	0.964	70.316		
	σ_y^2	2	$\alpha_0 + \alpha_1 W^2$	0.903	159.002			4	$\alpha_0 + \alpha_1 W^2 + \alpha_2 [log_2(O)]^2 + \alpha_3 WO$	0.970	85.004		
2D-Moving	μ_x	3	$\alpha_0 + \alpha_1 W + \alpha_2 V$	0.861	2.470	10	48.291	4	$\alpha_0 + \alpha_1 W^2 + \alpha_2 V + \alpha_3 I^2$	0.897	2.115	17	**29.504**
	σ_x^2	4	$\alpha_0 + \alpha_1 W^2 + \alpha_2 V^2$	0.394	90.503			7	$\alpha_0 + \alpha_1 WI + \alpha_2 [log_2(I)]^2 + \alpha_3 V^4 + \alpha_4 V/W + \alpha_5 VI + \alpha_6 W^4$	0.946	28.764		
	σ_y^2	3	$\alpha_0 + \alpha_1 W^2 + \alpha_2 V^2$	0.396	42.682			6	$\alpha_0 + \alpha_1 WI + \alpha_2 VI + \alpha_3 [log_2(I)]^2 + \alpha_4 W^2 + \alpha_5 V^4$	0.962	10.369		
2D-Moving-Temporal	μ_x	3	$\alpha_0 + \alpha_1 W + \alpha_2 V$	0.913	2.184	10	39.751	4	$\alpha_0 + \alpha_1 W + \alpha_2 V + \alpha_3 t_c^2$	0.918	2.087	13	**31.047**
	σ_x^2	4	$\alpha_0 + \alpha_1 W^2 + \alpha_2 V^2 + \alpha_3 V/W$	0.818	79.509			5	$\alpha_0 + \alpha_1 W^4 + \alpha_2 V^2 + \alpha_3 [log_2(t_c)]^2 + \alpha_4 V/t_c$	0.865	65.121		
	σ_y^2	3	$\alpha_0 + \alpha_1 W^2 + \alpha_2 V^2$	0.838	29.106			4	$\alpha_0 + \alpha_1 W^2 + \alpha_2 V^2 + \alpha_3 1/(t_c^4)$	0.896	22.970		
3D-Moving	μ_x	6	$\alpha_0 + \alpha_1 W + \alpha_2 V_x + \alpha_3 D + \alpha_4 V_{px} + \alpha_5 W_p$	0.917	1.992	33	45.234	4	$\alpha_0 + \alpha_1 W + \alpha_2 V_x + \alpha_3 \frac{1}{D^2}$	0.926	1.942	25	**37.119**
	μ_z	6	$\alpha_0 + \alpha_1 W + \alpha_2 V_z + \alpha_3 D + \alpha_4 V_{pz} + \alpha_5 W_p$	0.775	2.781			6	$\alpha_0 + \alpha_1 W^2 + \alpha_2 log_2(V_z) + \alpha_3 D^2 + \alpha_4 V_{pz}^2 + \alpha_5 W_p^2$	0.813	2.703		
	σ_x^2	8	$\alpha_0 + \alpha_1 V_x^2 + \alpha_2 V^2 + \alpha_3 W^2 + \alpha_4 D^2 + \alpha_5 \frac{V_x}{W} + \alpha_6 V_{px}^2 + \alpha_7 W_p^2$	0.908	2.781			5	$\alpha_0 + \alpha_1 \frac{1}{W^2} + \alpha_2 V_x^4 + \alpha_3 V_z^4 + \alpha_4 V_{px}$	0.911	2.652		
	σ_y^2	5	$\alpha_0 + \alpha_1 V^2 + \alpha_2 W^2 + \alpha_3 D^2 + \alpha_4 W_p^2$	0.854	1.711			4	$\alpha_0 + \alpha_1 \frac{1}{W^4} + \alpha_2 V_x^4 + \alpha_3 V_{px}$	0.885	1.560		
	σ_z^2	8	$\alpha_0 + \alpha_1 V_z^2 + \alpha_2 V^2 + \alpha_3 W^2 + \alpha_4 D^2 + \alpha_5 \frac{V_z}{W} + \alpha_6 V_{pz}^2 + \alpha_7 W_p^2$	0.758	2.598			6	$\alpha_0 + \alpha_1 \frac{1}{W^4} + \alpha_2 D^4 + \alpha_3 V_{pz}^2 + \alpha_4 \frac{W}{V_{pz}^2} + \alpha_5 \frac{V_x}{V_z}$	0.816	2.117		

rates based on the estimated μ and σ from each model across the 7 datasets. The accuracy of these predictions is then systematically evaluated by comparing them against the empirically observed error rates.

4.3.1 Error Rate Model. The error rate of a pointing task is defined as the proportion of selection attempts in which the user's endpoint falls outside the target region. With endpoint distribution predicted by N-ary Gaussian Model, we can further calculate the error rate for pointing a target via multivariate normal cumulative distribution function (CDF) [17]. For a 1D pointing task, the error rate corresponds to the probability that the endpoint falls outside the target range, which can be computed via the corresponding CDF [26]. For 2D circular targets, the probability of hitting a circle with radius $W/2$ centered at the target location is given by the integral of the bivariate normal distribution over that circular area [27].

Extending to 3D, for a spherical target of radius $R = W/2$. Let (X_t, X_n, X_z) denotes the 3D endpoint distribution, where $X_t \sim \mathcal{N}(\mu_t, \sigma_t^2)$, $X_n \sim \mathcal{N}(\mu_n, \sigma_n^2)$, and $X_z \sim \mathcal{N}(\mu_z, \sigma_z^2)$. The joint probability density function is:

$$f_{X_t,X_n,X_z}(x_t, x_n, x_z) = \frac{1}{(2\pi)^{3/2}\,\sigma_t\,\sigma_n\,\sigma_z} \exp\left(-\frac{1}{2}\left[\frac{(x_t-\mu_t)^2}{\sigma_t^2} + \frac{(x_n-\mu_n)^2}{\sigma_n^2} + \frac{(x_z-\mu_z)^2}{\sigma_z^2}\right]\right). \tag{13}$$

Using explicit integration limits, the probability that an endpoint falls inside this spherical target is given by:

$$P_{\text{in}}(R) = \int_{-R}^{R} \int_{-\sqrt{R^2-x_t^2}}^{\sqrt{R^2-x_t^2}} \int_{-\sqrt{R^2-x_t^2-x_n^2}}^{\sqrt{R^2-x_t^2-x_n^2}} f_{X_t,X_n,X_z}(x_t, x_n, x_z)\, \mathrm{d}x_z\, \mathrm{d}x_n\, \mathrm{d}x_t. \tag{14}$$

The corresponding error rate for a target is then:

$$E_{\text{err}}(R) = 1 - P_{\text{in}}(R). \tag{15}$$

4.3.2 Results. Based on the error rates predicted by the N-ary Gaussian model and the baseline model, we compute the Mean Absolute Error (MAE) between the predicted and actual error rates. The results are summarized in Fig. 4. Across all datasets, the N-ary Gaussian model consistently achieves a lower MAE than the baseline. Overall, our model reduces MAE by an average of 2.88 (SD = 6.18), corresponding to an average relative reduction of 19.07%. Notably, the model delivers its best error rate prediction on the 1D-Moving dataset, attaining an MAE of only 1.88. The most substantial improvement occurs on the 3D-Moving dataset, where the MAE drops sharply from 30.58 to 13.69. This pronounced gain suggests that the N-ary Gaussian framework is particularly effective in capturing the underlying error structure in more complex, high-dimensional task settings. These results align with the observed reduction in MWD reported earlier, reinforcing that the proposed model not only better approximates the distribution of human performance but also provides more accurate point estimates of error rates.

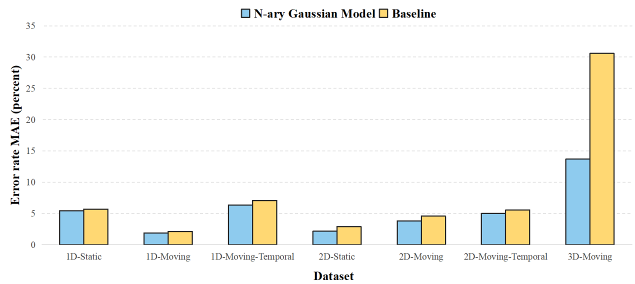


Figure 4: Comparison of error rate prediction MAE across 7 Datasets.

4.4 Comparison with Black-box Models

Black-box modeling approaches, through flexible adjustment of input features, offer the potential for automated modeling across diverse task scenarios. However, to our knowledge, no prior framework has been explicitly designed to model endpoint distributions in target pointing tasks using black-box methods. To establish a comparative baseline, we implemented two representative approaches—Support Vector Machine (SVM) regression with a polynomial kernel and a Feedforward Neural Network (FNN)—to predict Gaussian distribution parameters (μ and σ). The SVM utilized a 4th-degree polynomial kernel to capture non-linear interactions, matching the maximum term complexity of our N-ary model. The FNN architecture consisted of an input layer aligned with feature dimensionality, two hidden layers (64 and 32 neurons, respectively), and an output layer corresponding to the number of predicted parameters. Dropout regularization (rate = 20%) was applied to mitigate overfitting. Both models were trained to minimize mean squared error (MSE) between predicted and ground-truth Gaussian parameters, following the same fitting protocols as our method.

Table 4 summarizes the comparative performance. The SVM achieved the lowest MWD in Dataset 1D-Moving-Temporal, Dataset 2D-Static, Dataset 2D-Moving-Temporal, and Dataset 3D-Moving, while the FNN performed better in the remaining datasets. The N-ary Gaussian Model has the lowest model complexity (mean: 12.429, STD: 6.630), requiring at most 25 parameters (Dataset 3D-Moving) across all datasets, in sharp contrast to FNN, which consistently uses more than 2300 parameters (mean: 2467.429, STD: 91.460), and SVM (mean: 778.714, STD: 638.003), which varies widely from 26 to over 1600. Despite its lightweight structure, the N-ary Gaussian Model remains competitive in MWD performance. For example, in Dataset 1D-Static, N-ary Gaussian Model achieves its result with only 5 parameters, in contrast to the SVM's 26 and the FNN's staggering 2338 neurons. And the decrease in MWD from 0.523 to 0.338 offers limited practical benefit in many contexts, especially given that it requires nearly 500 times more parameters.

4.5 Cross-Validation

To evaluate the robustness of the proposed model, as well as to assess the potential for overfitting, we conducted cross-validation tests on each dataset. Given the need for a sufficient number of data points to fit the parameters of the Gaussian distribution, we employed a bootstrap resampling technique. For each iteration, we randomly selected 70% of participants (rounded up to the next number) with replacement as the training and testing sets. This procedure was repeated 10 times for each dataset to ensure comprehensive evaluation.

Across all 7 datasets, the N-ary Gaussian Model demonstrated an average reduction in Mean Wasserstein Distance (MWD) of 5.523 (STD = 5.247). On average, our model reduced MWD by 27.173% (STD = 0.0752) compared to baseline models. The black-box models continue to deliver the best overall performance, consistently achieving the lowest MWD values. However, the SVM revealed signs of overfitting, attaining the best results only in Dataset 2D-Moving and Dataset 3D-Moving. In contrast, the FNN demonstrated more stable generalization, securing optimal performance in other 5 datasets and reducing MWD by an average of 16.611% compared to our model, albeit at the cost of a substantially larger parameter footprint (mean : 2492.57, SD : 89.08). These results indicate that the model's predictive performance remains robust, supporting its generalizability and resistance to overfitting.

5 Discussion

5.1 Generalization of N-ary Gaussian Model in Diverse Datasets

To comprehensively evaluate the generalization capability of the proposed N-ary Gaussian Model, this section presents a detailed comparative analysis across the 7 diverse datasets. This section focuses on interpreting the automatically generated model forms, contrasting them with established baseline models from the literature, and examining how both traditional and novel task attributes affect pointing uncertainty.

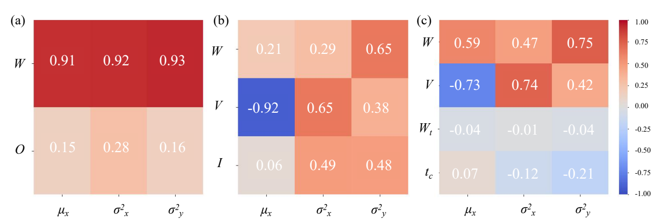
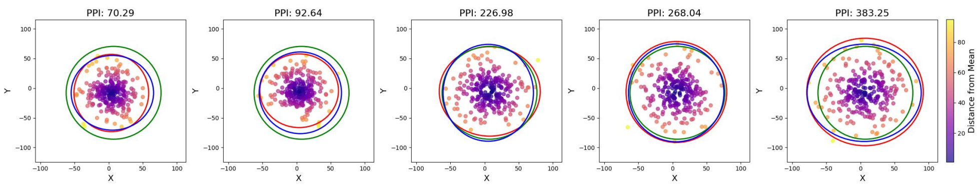


Figure 5: The Spearman correlation for three datasets: Dataset 2D-Static (added display device), Dataset 2D-Moving (added input modality), and Dataset 2D-Moving-Temporal (added time constraint).

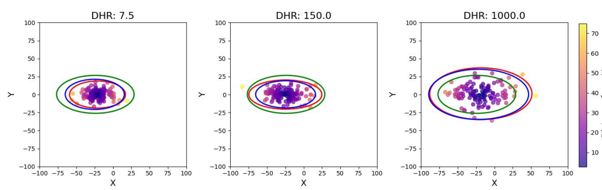
Dataset 1D-Static. Although existing research widely suggests that amplitude (A) has negligible influence in 1D static target selection tasks, our model incorporates the effect of target amplitude through a logarithmic function to achieve more precise modeling. The logarithmic form, widely adopted in foundational models such as Fitts' Law where index of difficulty is defined as $log_2(D/W + 1)$, not only captures positive correlations but also reflects a limiting

Table 4: N-ary Gaussian Model against black-box models. N refers to the number of parameters in the model.

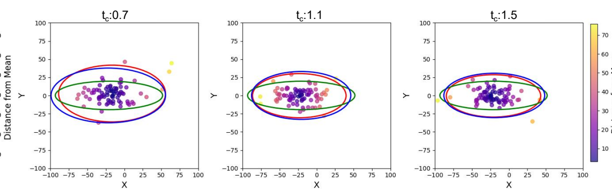
Dataset Name	SVM		FNN		N-ary Gaussion Model	
	N	MWD	N	MWD	N	MWD
1D-Static	26	0.395	2338	**0.338**	5	0.523
1D-Moving	450	1.990	2402	**1.929**	7	2.615
1D-Moving-Temporal	1418	**1.293**	2530	1.344	10	1.341
2D-Static	199	**2.498**	2404	2.860	11	3.179
2D-Moving	508	23.741	2468	**21.315**	16	29.504
2D-Moving-Temporal	1664	**28.829**	2532	30.087	13	31.047
3D-Moving	1186	**34.515**	2598	37.073	25	37.119



(a) Dataset 4: Bivariate Gaussian 3-Sigma Contour Plot with Gradient Across Display, for Target Width (W) of 144



(b) Dataset 5: Bivariate Gaussian 3-Sigma Contour Plot with Gradient Across Input Modality, for Target Width (W) of 144 and Velocity (V) of 256



(c) Dataset 6: Bivariate Gaussian 3-Sigma Contour Plot with Gradient Across Temporal Constraints, for Target Width (W) of 24, Velocity (V) of 384, and Temporal Widths (W_t) of 0.32"

Figure 6: Comparison of Bivariate Gaussian 3-Sigma Contour Plots across different display, input modalities, and temporal constraints. The color bar reflects the distance from the center. The three 3-sigma contour plots represent the real distribution (red), the N-ary Gaussian Model (blue), and the Baseline model (green).

effect: the influence of amplitude diminishes as its value increases. For small amplitudes, even minor changes significantly affect endpoint spread, while at larger amplitudes, distribution variability becomes increasingly constrained by motor control precision or perceptual limits.

Dataset 1D-Moving. Our approach can reduce the reliance on specialized expertise and prior experience in the modeling process. As shown in the model format for Dataset 1D-Moving, our method captured a model structure largely consistent with that of the original model [26], including an identical number of terms, while delivering comparable performance. However, an unexpected term, WV, emerged in the resulting model, which lacks a clear interpretation within conventional pointing paradigms. We argue that the emergence of this term offers a novel perspective, suggesting that endpoint variance in dynamic target selection is not solely determined by geometric difficulty (e.g., V/W) but is also influenced by a multiplicative interaction between velocity and target tolerance. We consider that the WV term captures a coupling between the inherent movement noise induced by high velocity and the subjective allocation of effort modulated by target width. Traditional indices of difficulty focus primarily on geometric constraints but often fail to account for the behavioral strategies users adopt in high-tolerance, high-dynamic environments. When the target width W is large, it provides substantial tolerance space, which may induce an effort-minimizing strategy [34]. This strategy can reduce the motor system's ability to suppress the intrinsic movement noise generated at high velocity V. Furthermore, this effect is moderated by individual risk sensitivity [43]. In scenarios with both high V and large W, the resulting high outcome variability represents increased risk. Risk-seeking individuals may perceive this variability as acceptable or even beneficial, thereby not adjusting their strategy, which allows variance to rise. Thus, the WV term may represent the observed velocity-tolerance interaction and suggests a novel cognitive and motor control interpretation for the increase in endpoint variance σ^2.

Dataset 2D-Static. We explicitly incorporate the physical pixel density (PPI) of the display device as a task attribute into the computational model.Unlike approaches that treat device differences as a black-box variable, our model captures how PPI, as a quantifiable physical factor, systematically influences pointing uncertainty during interaction.

Previous work has indeed investigated the effect of physical scale on selection performance [8], typically by varying the actual display area while keeping the resolution constant on a single device. However, in real-world scenarios where interactive devices vary greatly in both physical dimensions and resolution, PPI serves as a more comprehensive metric that inherently accounts for both factors.

On low-PPI displays, the minimal movement unit of an input device (such as one pixel) corresponds to a larger physical displacement. This amplifies motor control noise and leads to more dispersed endpoint distributions. As illustrated in Fig. 6 (a), for the same target width, the endpoint distribution of real data (plotted in red) becomes increasingly dispersed with higher PPI values. This effect explains why the baseline model performs poorly. These results show that PPI may be a worthy modeling attribute in pointing tasks.

Dataset 2D-Moving. While prior research has acknowledged the influence of input modality on pointing behavior, it has largely stopped short of developing quantitative models that integrate this factor into a unified predictive framework. Addressing this gap, our work quantifies input modality using the Device's Human Resolution (DHR) and rigorously evaluates whether incorporating this attribute is necessary through our automated modeling pipeline. As demonstrated using Dataset 2D-Moving from a 2D moving target selection task, a clear positive correlation exists between DHR and endpoint distribution variance (Fig. 5(b)). Specifically, higher-DHR devices such as the mouse (DHR = 1000) result in greater endpoint variability compared to the stylus (DHR = 150) and finger touch (DHR = 7.5), a finding consistent with existing literature [27]. Crucially, unlike earlier studies [27] that constructed separate models per input modality, our approach incorporates DHR directly as a continuous task attribute within a single, extensible model. This not only enhances predictive accuracy and generalizability across interaction contexts, but also provides a mathematically explicit representation—enabling deeper insight into how input resolution shapes pointing uncertainty.

Dataset 1D-/2D-Moving-Temporal. Modeling results from Dataset 2D-Moving-Temporal reveal that temporal constraints in spatiotemporal selection tasks exert a notable influence on spatial endpoint distributions and should be incorporated into the model. As shown in Fig. 6 (c), as time constraints become more stringent (i.e., with lower t_c values), reduced reaction time leads to increased variability in the endpoint distribution. Prior methods for modeling endpoint distribution often overlooked the role of temporal constraints [23, 51] and typically treated temporal and spatial factors in isolation. Our work underscores their interdependence. Our model identifies the V/t_c^4 term (in Dataset 1D-Moving-Temporal) and the $1/t_c^4$ term (in Dataset 2D-Moving-Temporal) as effective mechanisms for integrating temporal constraints with spatial uncertainty. These terms capture how the influence of time pressure intensifies as the temporal distance interval (t_c) decreases. This finding aligns with the view that shorter decision windows limit users' ability to fine-tune their movements, thereby increasing endpoint variability [55]. Such an effect may be especially critical in fast-paced interactive scenarios, such as first-person shooter games [9], where users must perform rapid and precise movements under extreme time pressure. In the original Dual Gaussian and Ternary Gaussian models, σ was typically described by linear functions (i.e., σ^2 was expressed quadratically). We extended this approach by introducing terms up to a quadratic function, which corresponds to 4th-order terms in σ^2. In our model, the majority of these 4th-order terms appear with W, suggesting that higher-order terms capture the user's increasing sensitivity to changes in this attribute. Importantly, this expansion retains the interpretability of the original positive correlation between task attributes and endpoint distribution. Our framework is designed for modeling endpoint distributions in spatial pointing tasks, as long as the endpoints of interest can be treated as approximately Gaussian. The present experiments focus on spatial endpoints without explicit temporal success criteria.

Dataset 3D-Moving. Unlike conventional 1D and 2D user interfaces, pointing in 3D environments presents considerably greater challenges due to the need to process depth perception cues. This difficulty is further exacerbated when the target is in motion. Consequently, modeling the endpoint distribution for 3D moving target selection is particularly demanding, as it requires accounting for up to eight attributes—significantly more than in 1D or 2D scenarios. This complexity inherently leads to more sophisticated models, such as the one proposed in the original paper [70], which employs 33 parameters. Notably, our approach achieves superior performance in the 3D task, outperforming the baseline model across key metrics. Furthermore, it offers a more streamlined model architecture with the number of parameters reduced to 25. This reduction not only maintains predictive capability but also provides valuable insights for future research in refining 3D modeling techniques. Compared to the baseline model [70], our N-ary Gaussian model for σ_z^2 incorporates two additional compound terms. The first, $\frac{W}{V_{pz}^2}$, relates to depth and time-to-collision (TTC) perception. Humans often estimate TTC using optical variables such as τ—the ratio of an object's angular size to its rate of expansion [36]. Since $V_{pz} = V_z/D$, the term expands to $\frac{W}{V_{pz}^2} = \frac{WD^2}{V_z^2} \propto W\tau^2$, suggesting that participants may rely on a τ-like cue rather than processing target width, depth, and speed independently. This interpretation aligns with work showing that TTC-related variables systematically affect depth judgment precision [62]. The model also includes the term $\frac{V_x}{V_z}$, whose functional role warrants further investigation. We speculate that this term may capture the anisotropy of endpoint uncertainty relative to the target's motion direction of 3D moving target selection. Overall, these terms indicate that the N-ary Gaussian formulation captures perceptual and geometric constraints not explicitly represented in the baseline model, pointing to promising directions for investigating the mechanisms underlying 3D moving target selection. We believe that further experimental and theoretical studies are needed to test the generality of this interpretation.

Based on the experimental results and analysis across diverse datasets, the following key findings are derived:

- **Display pixel density (PPI) systematically influences pointing precision,** with lower PPI displays leading to more dispersed endpoint distributions due to amplified human motor control noise.
- **Input modality, quantified by Device Human Resolution (DHR), significantly affects pointing uncertainty,** where higher-DHR devices like mice exhibit greater distribution variance than stylus or touch input.
- **Temporal constraints (t_c) interact with spatial precision,** where stricter time limits increase pointing uncertainty due to reduced movement correction opportunities. This demonstrates the

necessity of integrated spatial-temporal modeling for time-sensitive applications such gaming or real-time decision systems.

• **Potential for modeling pointing uncertainty in complex scenarios,** not only reducing parameter complexity but also maintaining predictive accuracy in challenging tasks. N-ary Gaussian model provides a tool for identifying and validating previously overlooked factors that influence interaction performance, making it particularly valuable for exploring sophisticated interactive environments where multiple task attributes coexist and interact.

5.2 Impact of BIC-Based Model Selection

In this section, we discuss how Bayesian Information Criterion (BIC) guided model selection within our model. Taking Dataset 2D-Moving-Temporal as an illustrative case, we recorded the top 20 model combinations for μ_x, σ_x^2, and σ_y^2 based on their BIC values. And, we collected the corresponding $Adj.R^2$ and Mean Absolute Error (MAE) values. The results for Dataset 2D-Moving-Temporal are shown in Fig. 7. Each row corresponds to μ_x, σ_x^2, and σ_y^2, with plots showing their associated $Adj.R^2$ and MAE values. The x-axis in each plot represents the number of parameters, while the y-axis indicates the respective $Adj.R^2$ or MAE values. Data points are represented as circles, and the models with the best BIC scores are marked with yellow stars. In the $Adj.R^2$ and MAE columns, models achieving higher $Adj.R^2$ values and lower MAE scores are emphasized in red, providing a clear visual distinction of their performance.

As shown in Fig. 7, models with more parameters tend to achieve better performance, as evidenced in the figure where the highest $Adj.R^2$ and the lowest MAE values are often associated with the most parameterized models. However, this improvement diminishes beyond a certain point. For instance, as shown in subplot (b), when the number of parameters increases from 5 to 6, the $Adj.R^2$ value only improves marginally, from 0.865–0.866 to 0.867, indicating that adding more parameters does not significantly enhance the model's predictive ability. BIC values, however, consistently favor models with fewer parameters, as highlighted by the yellow stars indicating the best BIC scores. The BIC-selected models retain relatively high $Adj.R^2$ values and competitive MAE scores despite having fewer parameters, keeping simplicity while preserving predictive accuracy. This ability to prioritize parameters with meaningful contributions ensures that the models generalize well across datasets, demonstrating that BIC balances model simplicity and performance by penalizing unnecessary complexity.

5.3 Impact of Low-Contribution Term Filter

This section focuses on validating the effectiveness of the Shapley value-based term filter. For each dataset, based on the model forms selected by BIC, we calculated the contribution ratio of each term to the model's fit using Eq. 11 and 12. This was applied in cases where the number of terms was two or more. Additionally, we computed the percentage reduction in $Adj.R^2$ when each term was removed, relative to the original $Adj.R^2$. A strong positive correlation was observed between the two measures (Spearman's $r = 0.975$, $p < 0.001$), indicating that terms with higher Shapley contributions generally led to larger reductions in model fit when excluded.

By setting a threshold of 5%, we successfully filtered out 3 low-contribution terms. These terms include the t_c term (contribution: 4.902%, $Adj.R^2$ reduction: 0.389%) in the Dataset 1D-Moving-Temporal for μ_x, the A term (contribution: 3.723%, $Adj.R^2$ reduction: 0.128%) for μ_x in the Dataset 1D-Static, and the WA interaction term (contribution: 3.991%, $Adj.R^2$ reduction: 0.032%) for σ_x^2. These results demonstrate the feasibility of using Shapley values to effectively quantify the contribution of each term to the model, while the term-filtering approach successfully removes low-impact terms with minimal effect on predictive accuracy. This confirms the validity of our term-filtering method in maintaining model simplicity without sacrificing performance.

5.4 Relationship between BIC Model Selection and Shapley Value Term Filtering

The relationship between BIC model selection and Shapley value term filtering is both complementary and collaborative. While both methods aim to reduce model complexity, they operate at different levels of the modeling process. BIC serves as a global selection criterion, guiding the choice of the overall model structure by balancing model fit and complexity. By introducing a penalty for the number of parameters, BIC ensures that simpler models are favored unless the inclusion of additional terms leads to a significant improvement in predictive performance. In contrast, Shapley values operate at a more granular level within the selected model, quantifying the contribution of each term to the overall prediction. This term-level analysis enables a finer degree of control, allowing for the removal of terms that make negligible contributions to model performance, even if they were initially retained under the BIC-based selection. BIC alone is not sufficient for our goal. BIC tells us which combination of terms gives the best trade-off between fit and complexity, but it does not say how important each single term is inside that model. When the sample size is large, several terms can each give a small improvement in likelihood that is enough to pass the BIC penalty, even though their unique effect is very small compared to the main terms. Shapley values complement this by providing a term-level measure of contribution within a fixed BIC-selected model, so that we can see which terms are doing most of the work and which ones only do small improvement.

Despite their shared objective of reducing model complexity, conflicts can arise between BIC model selection and Shapley value filtering. We have a linear workflow, which involves first executing BIC and then performing term filtering. Such conflicts typically occur when BIC retains a term that marginally improves model fit but is later identified by Shapley analysis as having a low contribution. For instance, interaction terms (e.g., W/V) may be selected by BIC due to their slight improvement in model fit, but their Shapley values may indicate minimal contribution if the individual effects of W and V are already strong. In this case, our model may favor filtering such a low contribution term. Importantly, using Shapley values in our pipeline does not override the decision made by BIC; it only simplifies the BIC-selected model at the term level when the effect on goodness of fit is negligible. Across the three terms that were filtered by the 5% threshold in our datasets, the average Shapley contribution was 4.644% (SD 0.504%), while their removal reduced $Adj.\ R^2$ by only 0.137 % on average (SD 0.151%), indicating

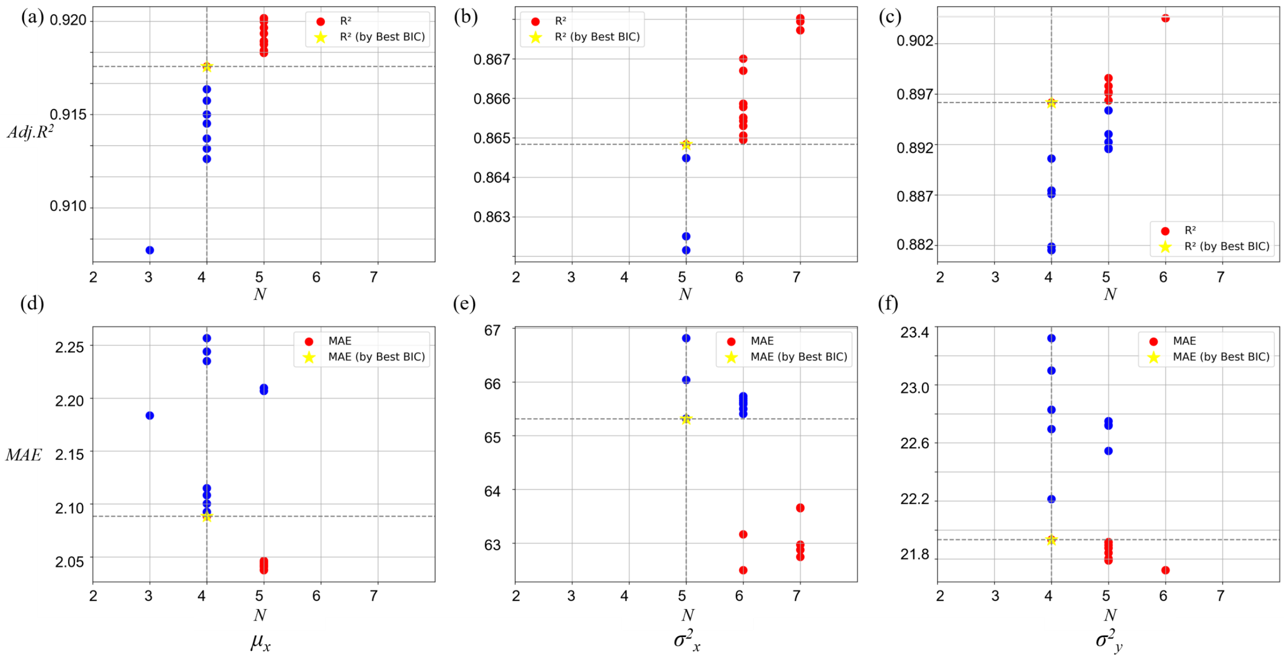


Figure 7: This figure highlights the effectiveness of BIC-based model selection in balancing simplicity and accuracy. Each row corresponds to μ_x, σ^2_x, and σ^2_y, with plots showing their associated $Adj.R^2$ and MAE values. The x-axis in each plot represents the number of parameters, while the y-axis indicates the respective $Adj.R^2$ or MAE values. Data points are represented as circles, and the models with the best BIC scores are marked with yellow stars. In the $Adj.R^2$ and MAE columns, models achieving higher $Adj.R^2$ values and lower MAE scores are emphasized in red, providing a clear visual distinction of their performance.

that Shapley-based pruning removes only terms with very small relative contribution and almost no impact on predictive performance. Besides, the order of applying BIC and Shapley is important. Shapley values are defined with respect to a given model and are costly to compute, so using Shapley first would require computing them for many unstable candidate models and the importance ranking of terms would change whenever the model structure changes. In contrast, using BIC first stabilizes the model structure and restricts the search space, after which Shapley values provide a clear and efficient term-level view.

5.5 Runtime Analysis of N-ary Gaussian Model

The runtime of automated modeling is a key concern for researchers, as also noted in the study by Oulasvirta [44]. In their experiment on a multitouch rotations dataset with five controlled factors, while satisfactory performance was achieved, the automated search for a nonlinear regression model required 96 hours to complete. To quantify the computational cost, we measured the runtime of an exhaustive search for constructing the N-ary Gaussian model across different numbers of attributes (Fig. 8). All programs were executed on a desktop computer equipped with an Intel Core i9-10900K CPU @ 3.70 GHz and 32 GB of RAM, running 64-bit Windows 11.

The computational cost remains practical for small-to-medium attribute sets: with up to 4 attributes, the model can be fitted within a few minutes (mean = 132.53 seconds), which is fully acceptable for offline analysis in typical experimental settings. Beyond 4 attributes, the runtime increases with the dimensionality of the attribute space, reaching approximately 27.5 minutes for 5 attributes and about 2.5 hours for 8 attributes. These results indicate that the proposed N-ary Gaussian Model is practically usable for small to medium numbers of attributes, but scaling to higher-dimensional attribute sets will require further optimisation. In future work, we plan to parallelise the core computations and port them to a GPU implementation, which should substantially reduce runtime and make exhaustive modelling with larger attribute numbers feasible.

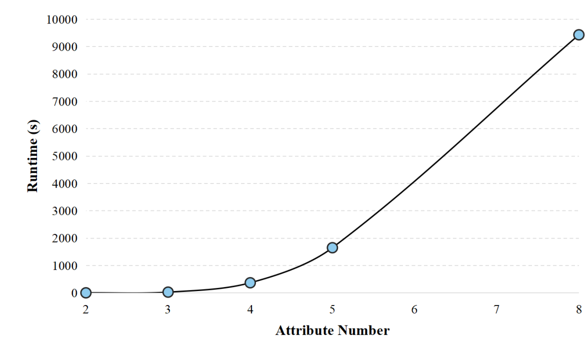


Figure 8: Growth in runtime with increasing attribute numbers.

5.6 Potential Extensions of N-ary Gaussian Model

The N-ary Gaussian Model introduces a systematic, theoretically grounded approach for automating endpoint distribution modeling. It enables the rapid identification of whether a specific task attribute is "modelable" and establishes a lower bound for the model's performance under that attribute. Traditionally, this process required researchers to hypothesize potential task attributes [23, 26], iteratively formulate multiple models [64], and evaluate their performance using metrics such as $Adj.R^2$, MAE, AIC, and BIC, etc [61]. This trial-and-error process is labor-intensive and often relies on subjective decisions about which task attributes to include or exclude from the model. Our approach offers a more systematic solution by employing BIC-based model selection and Shapley value analysis to systematically assess and exclude low-contribution terms. By automating the process of determining which attributes to include, our approach accelerates the modeling process, minimizes manual intervention, and produces models that retain only the most relevant task attributes. We do not propose that these results should supersede existing models but rather aim to show how the method can facilitate the exploration and refinement of model spaces. This tool can also help us gain an initial understanding of the impact of unexplored but potentially significant attributes on endpoint distribution, such as walking or running [38], emotional states [48] for a broader range of applications.

The ability to automatically derive models can accelerate research pursuits in adaptive UI and UI optimization. For theory-driven researchers who previously regarded modeling as a complex and specialized endeavor, this approach lowers the entry barrier, enabling them to obtain models in scenarios where theoretical rigor is not a primary concern. For instance, based on the Multi-Gaussian distribution hypothesis, our model offers a structured and transparent modeling approach. Leveraging the final Gaussian form of the model, we can utilize the likelihood function from Bayesian inference to infer the user's intended target during interaction [27], forming interaction techniques that focus more on model efficiency. Furthermore, in the field of UI design, if designers seek to understand how task attributes influence error rates, the functional forms for each task attribute enable the estimation of error rates using the cumulative distribution function (CDF) [25]. This capability may guide interface design, allowing designers to predict user performance and optimize UI design parameters for enhanced the usability of user interfaces.

6 Limitations

This study has four primary limitations. First, the target shapes are limited to regular geometric shapes, and the endpoint distributions are assumed to follow a normal distribution. However, given that most target selection tasks in current interactive interfaces involve regular shapes, and previous research has shown that the endpoint distributions for such shapes typically follow a normal distribution, our work remains broadly applicable to the design of mainstream user interfaces.

Second, as an extension of the Dual- and Ternary-Gaussian models, our multi-Gaussian framework builds on these foundational models for endpoint distribution. In our white-box model baseline comparisons, we primarily focus on these two models and their extensions. This choice is justified by the fact that Dual- and Ternary-Gaussian models have demonstrated strong adaptability and robustness across a wide range of target selection tasks, making them a solid foundation for further development. Therefore, despite the focus on these models, our approach provides a reasonable and scalable solution for modeling endpoint distributions in various task scenarios.

Third, our model lies in its reliance on an automated search process, which makes it highly dependent on data quality. This dependence can lead to the inclusion of terms that lack practical interpretability (e.g., 4th-order terms) or are inconsistent with established findings from prior research (e.g., WV in this study differ from W/V proposed in Ternary-Gaussian Model [27]). Such issues pose challenges, particularly in modeling scenarios where interpretability is a key requirement. To address this limitation, one potential solution is to incorporate theory-driven constraints on the model's functional forms. Since users in these scenarios are often researchers with domain knowledge, it is feasible to predefine or place strict constraints on the range and form of attribute terms. This approach may limit the model exploration space only to new attributes while enhancing the interpretability of existing attributes.

A further limitation of our framework is that it can only search over parameters that were explicitly instrumented and logged in the original experiment. In practice, this means that the hypothesis space explored by the N-ary Gaussian pipeline is bounded not only by the chosen function set, but also by the experimental logging scheme: unrecorded factors cannot appear in the resulting models, except indirectly through their correlations with logged variables. In the future, studies that intend to use automated model search should be encouraged to log a richer set of task- and context-level attributes, so that the resulting models can more fully capture the key features of endpoint distribution.

7 Conclusion

This paper introduced the N-ary Gaussian Model, an automated approach to describe endpoint distribution across various pointing tasks. We evaluated the model through a benchmark comparison with published models, examining its performance across various scenarios. Our results demonstrated that the model's generalizability across a range of task settings outperforming white-box baselines, and keeping competitive accuracy comparing with black-box models while using fewer parameters. It further contributes to the field by explicitly incorporating cross output device, input modality, and temporal constraint factors into spatial pointing uncertainty modeling, thereby offering new insights into spatial-temporal target selection. This work opens avenues for future research in complex interaction environments such as virtual reality and multimodal systems. This work opens avenues for future research in complex interaction environments where modeling demands are particularly high, especially in scenarios characterized by numerous interacting factors that traditionally require extensive domain knowledge to model accurately.

Acknowledgments

We thank the Associate Chairs and anonymous reviewers for their constructive comments that helped improve this paper. This work was supported by the National Natural Science Foundation of China (62332017, 62402486), Key Project of the Institute of Software Chinese Academy of Sciences(ISCAS-ZD-202401), Key Project of Key Laboratory of System Software (Chinese Academy of Sciences) (ZD202502), and the CAS Project for Young Scientists in Basic Research (Grant No. YSBR-040).

References

[1] Johnny Accot and Shumin Zhai. 1997. Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '97)*. Association for Computing Machinery, New York, NY, USA, 295–302. doi:10.1145/258549.258760

[2] Johnny Accot and Shumin Zhai. 2002. More than dotting the i's—foundations for crossing-based interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 73–80.

[3] Hirotugu Akaike. 1974. A new look at the statistical model identification. *IEEE transactions on automatic control* 19, 6 (1974), 716–723.

[4] François Bérard, Guangyu Wang, and Jeremy R Cooperstock. 2011. On the limits of the human motor control precision: the search for a device's human resolution. In *Human-Computer Interaction–INTERACT 2011: 13th IFIP TC 13 International Conference, Lisbon, Portugal, September 5-9, 2011, Proceedings, Part II 13*. Springer, 107–122.

[5] Xiaojun Bi, Yang Li, and Shumin Zhai. 2013. FFitts law: modeling finger touch with fitts' law. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1363–1372.

[6] Xiaojun Bi and Shumin Zhai. 2013. Bayesian touch: a statistical criterion of target selection with finger touch. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 51–60.

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

[8] Graeme Browning and Robert J. Teather. 2014. Screen scaling: effects of screen scale on moving target selection. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI EA '14)*. Association for Computing Machinery, New York, NY, USA, 2053–2058. doi:10.1145/2559206.2581227

[9] Brad J Bushman. 2019. "Boom, Headshot!": Violent first-person shooter (FPS) video games that reward headshots train individuals to aim for the head when shooting a realistic firearm. *Aggressive behavior* 45, 1 (2019), 33–41.

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

[11] Xiuli Chen, Gilles Bailly, Duncan P Brumby, Antti Oulasvirta, and Andrew Howes. 2015. The emergence of interactive behavior: A model of rational menu search. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 4217–4226.

[12] Xiuli Chen, Sandra Dorothee Starke, Chris Baber, and Andrew Howes. 2017. A cognitive model of how people make decisions through interaction with visual displays. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 1205–1216.

[13] Jeremy Chu, Yan Ma, Shumin Zhai, Xianfeng David Gu, and Xiaojun Bi. 2023. TouchType-GAN: Modeling Touch Typing with Generative Adversarial Network. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 29, 13 pages. doi:10.1145/3586183.3606760

[14] Mark Claypool. 2018. Game input with delay—moving target selection with a game controller thumbstick. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14, 3s (2018), 1–22.

[15] Mark Claypool, Andy Cockburn, and Carl Gutwin. 2019. Game input with delay: Moving target selection parameters. In *Proceedings of the 10th ACM Multimedia Systems Conference*. 25–35.

[16] Mark Claypool, Ragnhild Eg, and Kjetil Raaen. 2016. Modeling user performance for moving target selection with a delayed mouse. In *International Conference on Multimedia Modeling*. Springer, 226–237.

[17] Zvi Drezner. 1994. Computation of the trivariate normal integral. *Math. Comput.* 62, 205 (Jan. 1994), 289–294. doi:10.2307/2153409

[18] Brett R Fajen and William H Warren. 2007. Behavioral dynamics of intercepting a moving target. *Experimental Brain Research* 180, 2 (2007), 303–319.

[19] Tovi Grossman and Ravin Balakrishnan. 2005. A probabilistic approach to modeling two-dimensional pointing. *ACM Transactions on Computer-Human Interaction (TOCHI)* 12, 3 (2005), 435–459.

[20] John Paulin Hansen, Vijay Rajanna, I. Scott MacKenzie, and Per Bækgaard. 2018. A Fitts' law study of click and dwell interaction by gaze, head and mouse with a head-mounted display. In *Proceedings of the Workshop on Communication by Gaze Interaction* (Warsaw, Poland) *(COGAIN '18)*. Association for Computing Machinery, New York, NY, USA, Article 7, 5 pages. doi:10.1145/3206343.3206344

[21] Khalad Hasan, Tovi Grossman, and Pourang Irani. 2011. Comet and target ghost: techniques for selecting moving targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 839–848.

[22] ERROL R. HOFFMANN. 1991. Capture of moving targets: a modification of Fitts' Law. *Ergonomics* 34, 2 (1991), 211–220. arXiv:https://doi.org/10.1080/00140139108967307 doi:10.1080/00140139108967307

[23] Jin Huang and Byungjoo Lee. 2019. Modeling error rates in spatiotemporal moving target selection. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.

[24] Jin Huang, Xiaolan Peng, Feng Tian, Hongan Wang, and Guozhong Dai. 2018. Modeling a target-selection motion by leveraging an optimal feedback control mechanism. *Sci. China Inf. Sci.* 61, 4 (2018), 044101–1.

[25] Jin Huang, Feng Tian, Xiangmin Fan, Huawei Tu, Hao Zhang, Xiaolan Peng, and Hongan Wang. 2020. Modeling the endpoint uncertainty in crossing-based moving target selection. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[26] Jin Huang, Feng Tian, Xiangmin Fan, Xiaolong Zhang, and Shumin Zhai. 2018. Understanding the uncertainty in 1D unidirectional moving target selection. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–12.

[27] Jin Huang, Feng Tian, Nianlong Li, and Xiangmin Fan. 2019. Modeling the uncertainty in 2D moving target selection. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*. 1031–1043.

[28] Dugald Hutchings. 2012. An investigation of Fitts' law in a multiple-display environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 3181–3184. doi:10.1145/2207676.2208736

[29] Faustina Hwang, Nic Hollinworth, and Nitin Williams. 2013. Effects of target expansion on selection performance in older computer users. *ACM Transactions on Accessible Computing (TACCESS)* 5, 1 (2013), 1–26.

[30] Shunichiro Ikeno, Chia-Ming Chang, and Takeo Igarashi. 2023. Sticky Cursor: A Technique for Facilitating Moving Target Selection. In *International Conference on Human-Computer Interaction*. Springer, 467–483.

[31] Richard J. Jagacinski, Daniel W. Repperger, Sharon L. Ward, and Martin S. Moran. 1980. A Test of Fitts' Law with Moving Targets. *Human Factors* 22, 2 (1980), 225–233. arXiv:https://doi.org/10.1177/001872088002200211 doi:10.1177/001872088002200211 PMID: 7390506.

[32] Jussi Jokinen, Aditya Acharya, Mohammad Uzair, Xinhui Jiang, and Antti Oulasvirta. 2021. Touchscreen typing as optimal supervisory control. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–14.

[33] Alvin Jude, Darren Guinness, and G. Michael Poor. 2016. Reporting and Visualizing Fitts's Law: Dataset, Tools and Methodologies. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 2519–2525. doi:10.1145/2851581.2892364

[34] Feng-Yang Kuo, Tsai-Hsin Chu, Meng-Hsiang Hsu, and Hong-Ssu Hsieh. 2004. An investigation of effort–accuracy trade-off and the impact of self-efficacy on Web searching behaviors. *Decision Support Systems* 37, 3 (2004), 331–342. doi:10.1016/S0167-9236(03)00032-0

[35] Byungjoo Lee, Sunjun Kim, Antti Oulasvirta, Jong-In Lee, and Eunji Park. 2018. Moving target selection: A cue integration model. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[36] David Lee. 1976. A Theory of Visual Control of Braking Based on Information about Time-to-Collision. *Perception* 5 (02 1976), 437–59. doi:10.1068/p050437

[37] Injung Lee, Sunjun Kim, and Byungjoo Lee. 2019. Geometrically compensating effect of end-to-end latency in moving-target selection games. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[38] Yang Li, Juan Liu, Jin Huang, Yang Zhang, Xiaolan Peng, Yulong Bian, and Feng Tian. 2024. Evaluating the effects of user motion and viewing mode on target selection in augmented reality. *International Journal of Human-Computer Studies* 191 (2024), 103327.

[39] Zhi Li, Yu-Jung Ko, Aini Putkonen, Shirin Feiz, Vikas Ashok, Iv Ramakrishnan, Antti Oulasvirta, and Xiaojun Bi. 2023. Modeling Touch-based Menu Selection Performance of Blind Users via Reinforcement Learning. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 357, 18 pages. doi:10.1145/3544548.3580640

[40] Yuexing Luo and Daniel Vogel. 2014. Crossing-based selection with direct touch input. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 2627–2636.
[41] I. Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Hum.-Comput. Interact.* 7, 1 (mar 1992), 91–139. doi:10.1207/s15327051hci0701_3
[42] Jörg Müller, Antti Oulasvirta, and Roderick Murray-Smith. 2017. Control theoretic models of pointing. *ACM Transactions on Computer-Human Interaction (TOCHI)* 24, 4 (2017), 1–36.
[43] Arne J. Nagengast, Daniel A. Braun, and Daniel M. Wolpert. 2011. Risk-sensitivity and the mean-variance trade-off: decision making in sensorimotor control. *Proceedings of the Royal Society B: Biological Sciences* 278, 1716 (01 2011), 2325–2332. arXiv:https://royalsocietypublishing.org/rspb/article-pdf/278/1716/2325/567843/rspb.2010.2518.pdf doi:10.1098/rspb.2010.2518
[44] Antti Oulasvirta. 2014. Automated nonlinear regression modeling for hci. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3899–3902.
[45] Antti Oulasvirta, Sunjun Kim, and Byungjoo Lee. 2018. Neuromechanics of a button press. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
[46] Antti Oulasvirta, Per Ola Kristensson, Xiaojun Bi, and Andrew Howes. 2018. *Computational interaction*. SIPRI/Oxford University Press. doi:10.1093/oso/9780198799603.001.0001 Publisher Copyright: © Oxford University Press 2018. All rights reserved. Copyright: Copyright 2020 Elsevier B.V., All rights reserved..
[47] Yong S Park, Sung H Han, Jaehyun Park, and Youngseok Cho. 2008. Touch key design for target selection on a mobile phone. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*. 423–426.
[48] Xiaolan Peng, Xurong Xie, Jin Huang, Chutian Jiang, Haonian Wang, Alena Denisova, Hui Chen, Feng Tian, and Hongan Wang. 2023. Challengedetect: Investigating the potential of detecting in-game challenge experience from physiological measures. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–29.
[49] Adrian Ramcharitar and Robert J. Teather. 2017. A Fitts' Law Evaluation of Video Game Controllers: Thumbstick, Touchpad and Gyrosensor. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 2860–2866. doi:10.1145/3027063.3053213
[50] David A. Rosenbaum and Markus Janczyk. 2019. Who is or was E. R. F. W. Crossman, the champion of the Power Law of Learning and the developer of an influential model of aiming? *Psychonomic Bulletin & Review* 26, 4 (2019), 1449–1463. doi:10.3758/s13423-019-01583-z
[51] Adrian L Jessup Schneider and TC Nicholas Graham. 2023. Supporting aim assistance algorithms through a rapidly trainable, personalized model of players' spatial and temporal aiming ability. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–17.
[52] Immo Schuetz, T. Scott Murdison, Kevin J. MacKenzie, and Marina Zannoli. 2019. An Explanation of Fitts' Law-like Performance in Gaze-Based Selection Tasks Using a Psychophysics Approach. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3290605.3300765
[53] Gideon Schwarz. 1978. Estimating the dimension of a model. *The annals of statistics* (1978), 461–464.
[54] Siti Rohkmah Mohd Shukri and Andrew Howes. 2019. Children adapt drawing actions to their own motor variability and to the motivational context of action. *International Journal of Human-Computer Studies* 130 (2019), 152–165.
[55] Katrin Sutter, Leonie Oostwoud Wijdenes, Robert J van Beers, and W Pieter Medendorp. 2021. Movement preparation time determines movement variability. *Journal of Neurophysiology* 125, 6 (2021), 2375–2383.
[56] Huawei Tu, Susu Huang, Jiabin Yuan, Xiangshi Ren, and Feng Tian. 2019. Crossing-based selection with virtual reality head-mounted displays. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–14.
[57] Eyal Winter. 2002. The shapley value. *Handbook of game theory with economic applications* 3 (2002), 2025–2054.
[58] Jacob O. Wobbrock, Edward Cutrell, Susumu Harada, and I. Scott MacKenzie. 2008. An error model for pointing based on Fitts' law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 1613–1622. doi:10.1145/1357054.1357306
[59] Jacob O. Wobbrock, Edward Cutrell, Susumu Harada, and I. Scott MacKenzie. 2008. An error model for pointing based on Fitts' law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 1613–1622. doi:10.1145/1357054.1357306
[60] Jacob O. Wobbrock, Kristen Shinohara, and Alex Jansen. 2011. The effects of task dimensionality, endpoint deviation, throughput calculation, and experiment design on pointing measures and models. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 1639–1648. doi:10.1145/1978942.1979181
[61] Shota Yamanaka and Hiroki Usuba. 2020. Rethinking the Dual Gaussian Distribution Model for Predicting Touch Accuracy in On-screen-start Pointing Tasks. *Proc. ACM Hum.-Comput. Interact.* 4, ISS, Article 205 (Nov. 2020), 20 pages. doi:10.1145/3427333
[62] Jing-Jiang Yan, Bailey Lorv, Hong Li, and Hong-Jin Sun. 2011. Visual processing of the impending collision of a looming object: Time to collision revisited. *Journal of Vision* 11, 12 (10 2011), 7–7. doi:10.1167/11.12.7
[63] Chuang-Wen You, Yung-Huan Hsieh, Wen-Huang Cheng, and Yi-Hsuan Hsieh. 2014. AttachedShock: Design of a crossing-based target selection technique on augmented reality devices and its implications. *International journal of human-computer studies* 72, 7 (2014), 606–626.
[64] Difeng Yu, Brandon Victor Syiem, Andrew Irlitti, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2023. Modeling Temporal Target Selection: A Perspective from Its Spatial Correspondence. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 577, 14 pages. doi:10.1145/3544548.3581011
[65] Shumin Zhai, Jing Kong, and Xiangshi Ren. 2004. Speed–accuracy tradeoff in Fitts' law tasks—on the equivalency of actual and nominal pointing precision. *International journal of human-computer studies* 61, 6 (2004), 823–856. doi:10.1016/j.ijhcs.2004.09.007
[66] Hao Zhang, Jin Huang, Huawei Tu, and Feng Tian. 2023. Shape-Adaptive Ternary-Gaussian Model: Modeling Pointing Uncertainty for Moving Targets of Arbitrary Shapes. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.
[67] Hao Zhang, Jin Huang, Huawei Tu, Feng Tian, Guozhong Dai, and Hongan Wang. 2023. A unified user behavior model for trajectory-based tasks with different types of path constraints. *Science China Information Sciences* 66, 10 (2023), 204101.
[68] Xinyong Zhang, Hongbin Zha, and Wenxin Feng. 2012. Extending Fitts' law to account for the effects of movement direction on 2d pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 3185–3194. doi:10.1145/2207676.2208737
[69] Ziyue Zhang, Jin Huang, and Feng Tian. 2020. Modeling the Uncertainty in Pointing of Moving Targets with Arbitrary Shapes. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–7.
[70] Yawen Zheng, Jin Huang, Hao Zhang, Yulong Bian, Juan Liu, Chenglei Yang, Feng Tian, and Xiangxu Meng. 2025. 3D Ternary-Gaussian model: Modeling pointing uncertainty of 3D moving target selection in virtual reality. *International Journal of Human-Computer Studies* 198 (2025), 103454. doi:10.1016/j.ijhcs.2025.103454

A Velocity Coordinate System

The velocity coordinate system serves as a bridge integrating both dynamic and static factors, ensuring that the static and dynamic models share the same structural form by using a velocity-based local coordinate system [27]. We have applied this approach in our N-ary Gaussian model, where the local coordinate system is defined with the x-axis tangent to the direction of target motion and the y-axis perpendicular to it. Specifically, for a target moving in direction θ, the endpoint $[x, y]^T$ indicated by the user in the world coordinate system is transformed as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta) & \cos(-\theta) \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} tx \\ ty \end{bmatrix} \right), \tag{16}$$

Equivalently, this can be rewritten as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} tx \\ ty \end{bmatrix} \right), \tag{17}$$

where $[x', y']^T$ represents the transformed endpoint position, and $[tx, ty]^T]$ is the center coordinate of the target when the user lands on it.[3]

[3] In practice, for one-dimensional target selection tasks, the rotation matrix described above can be applied to convert coordinates into a single x-axis data. This approach allows the focus to remain solely on the x-axis during the subsequent modeling process.

B Specific Running Example

To illustrate how the proposed pipeline operates in practice, we take a 1D static target selection task as an example. This task involves two task attributes: target width W and the distance between start button and the target A. Our goal is to automatically construct models for the Gaussian parameters μ_x and σ_x of the endpoint distribution ($X \sim N(\mu_x, \sigma_x^2)$).

Step 1: Gaussian Components Automatic Generator. In the multi-Gaussian framework, the endpoint distribution is modeled as a sum of latent Gaussian components, each associated with a specific task attribute or an interaction between attributes. The Gaussian Components Automatic Generator first maps task attributes to Gaussian components. For the 1D static target selection task, our model construct one component X_W associated with W and one component X_A associated with A for both μ_x and σ_x (follow Eq. 7), and construct an interaction component $Cov(W, AV)$ for σ_x to capture their joint effect:

$$\mu_x = \mu_W + \mu_A = f_1(W) + g_1(A),$$
$$\sigma_x^2 = \sigma_W^2 + \sigma_A^2 + Cov(W, A) = f_2(W)^2 + g_2(A)^2 + h_2(WA).$$

Each $f(\cdot)$ and $g(\cdot)$ is a single-attribute term, $h(\cdot)$ is an interaction term. The mappings $f(\cdot)$ and $g(\cdot)$ are instantiated using the functions in Table 1, and $h(\cdot)$ is calculated following Eq. 8. We define the candidate function pools for W, A, and their interaction as follows:

$$\mathcal{F}_W = \{\, W,\ W^2,\ 1/W,\ \log_2 W,\ \}, \quad f(\cdot) \in \mathcal{F}_W,$$
$$\mathcal{G}_A = \{\, A,\ A^2,\ 1/A,\ \log_2 A,\ \}, \quad g(\cdot) \in \mathcal{G}_V,$$
$$\mathcal{H}_{WA} = \{\, W^2A^2,\ W^2A,\ W^2A^{-1},\ W^2A^{-2}, \dots \}, \quad h(\cdot) \in \mathcal{H}_{WA}.$$

Through the above procedure, we complete the transition from task attributes to instantiated model terms: the multi-Gaussian framework first links each attribute to a conceptual component (X_W, X_A, and Cov(W, A)), and the function pools then turn each component into a specific term (Table. 5). Once a function is selected from each pool, X_W, X_A, and Cov(W, A) are realized as specific terms, and μ_x and σ_x^2 in Eq. 7 become sums of these instantiated terms, which will be further estimated and selected in the subsequent steps.

Step 2: BIC-based model selection. With the function pools defined in Step 1, we next decide which specific forms should be used. We exhaustively enumerate all valid combinations of terms from $\mathcal{F}_W$, $\mathcal{G}_A$, and $\mathcal{H}_{WA}$, fit the corresponding coefficients to the observed endpoints, and compute the BIC value for each candidate using Eq. 10. The combination that yields the lowest BIC is retained as the optimal choice, giving a specific functional form for μ_x and σ_x^2 (Top 5 models ranked by BIC see Table. 6), which is then passed to Step 3 for Shapley-based term filtering.

Step 3: Shapley-based term filter. After Step 2, the BIC-based selection has fixed the specific functional forms of μ_x and σ_x^2. In this step, we quantify how much each instantiated term contributes to the overall model using Shapley values. Each non-constant term is treated independently. The final form of μ_x and σ_x^2 obtained in Step 2 is

$$\mu_x = \alpha_0 + \alpha_1 W + \alpha_2 A, \sigma_x = \alpha_0 + \alpha_1 W^4 + \alpha_2\ [log_2 A]^2 + \alpha_3 WA.$$

Table 5: Examples of function pools for μ_x and σ_x^2.

Index	Candidate term combinations for μ_x
1	$\alpha_0 + \alpha_1 W$
2	$\alpha_0 + \alpha_1 W + \alpha_2 A$
3	$\alpha_0 + \alpha_1 W + \alpha_2 A^2$
4	$\alpha_0 + \alpha_1 W + \alpha_2 \log_2 A$
⋮	⋮
33	$\alpha_0 + \alpha_1 W^{-2} + \alpha_2 \log_2 A$
34	$\alpha_0 + \alpha_1 W^{-2} + \alpha_2 A^{-1}$
35	$\alpha_0 + \alpha_1 W^{-2} + \alpha_2 A^{-2}$
Index	**Candidate term combinations for σ_x^2**
1	$\alpha_0 + \alpha_1 W^2$
2	$\alpha_0 + \alpha_1 W^2 + \alpha_2 A^2 + \alpha_3 W^2 A^2$
3	$\alpha_0 + \alpha_1 W^2 + \alpha_2 A^2 + \alpha_3 W^2 A^1$
4	$\alpha_0 + \alpha_1 W^2 + \alpha_2 A^2 + \alpha_3 W^2 A^{-1}$
⋮	⋮
558	$\alpha_0 + \alpha_1 W^{-4} + \alpha_2 A^{-4} + \alpha_3 W^{-2} A^1$
559	$\alpha_0 + \alpha_1 W^{-4} + \alpha_2 A^{-4} + \alpha_3 W^{-2} A^{-1}$
560	$\alpha_0 + \alpha_1 W^{-4} + \alpha_2 A^{-4} + \alpha_3 W^{-2} A^{-2}$

Table 6: Top-5 models ranked by BIC for μ_x and σ_x^2.

Index	Top-5 BIC for μ_x	Top-5 BIC for σ_x
1	$\alpha_0 + \alpha_1 W + \alpha_2 A$	$\alpha_0 + \alpha_1 W^4 + \alpha_2\ [log_2]^2 + \alpha_3 WA$
2	$\alpha_0 + \alpha_1 W + \alpha_2 A^{-1}$	$\alpha_0 + \alpha_1 W^4 + \alpha_2 A^2 + \alpha_3 WA$
3	$\alpha_0 + \alpha_1 W + \alpha_2 A^{-2}$	$\alpha_0 + \alpha_1 W^4 + \alpha_2 A^4 + \alpha_3 WA$
4	$\alpha_0 + \alpha_1 W + \alpha_2 A^2$	$\alpha_0 + \alpha_1 W^4 + \alpha_2 A^{-2} + \alpha_3 WA$
5	$\alpha_0 + \alpha_1 W + \alpha_2 \log_2 A$	$\alpha_0 + \alpha_1 W^4 + \alpha_2 A^{-2}$

Following Eq. 11, the contribution of each term for μ_x is: ϕ_W = 96.277%, ϕ_A = 3.990%; the contribution of each term for σ_x is: ϕ_{W^4} = 80.358%, $\phi_{[log_2 A]^2}$ = 15.651%, ϕ_{WA} = 3.991%. Terms whose normalized contribution is below 5% (i.e., A in μ_x and WA in σ_x) are removed, yielding the final simplified endpoint model:

$$\mu_x = \alpha_0 + \alpha_1 W, \sigma_x = \alpha_0 + \alpha_1 W^4 + \alpha_2\ [log_2 A]^2.$$