HAPTCHA: Eyes-Free Haptic CAPTCHA in Mobile Environments: Design Principle, Security Evaluation, and User Experience

Duhong Kweon
Interaction Lab
Pohang University of Science and Technology (POSTECH)
Pohang, Republic of Korea
duhongk27@postech.ac.kr

Dajin Lee
Interaction Lab
Pohang University of Science and Technology (POSTECH)
Pohang, Republic of Korea
dj1122@postech.ac.kr

Seulbae Kim
Computer Security Lab
Pohang University of Science and Technology (POSTECH)
Pohang, Republic of Korea
seulbae@postech.ac.kr

Seungmoon Choi
Interaction Lab
Pohang University of Science and Technology (POSTECH)
Pohang, Republic of Korea
choism@postech.ac.kr

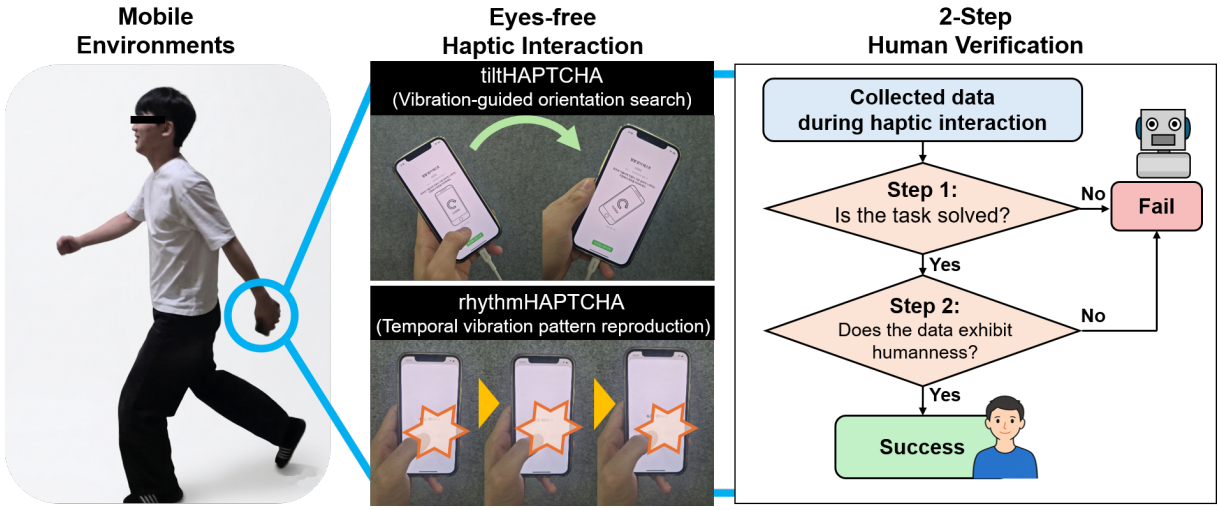


Figure 1: HAPTCHA: Eyes-free haptic CAPTCHA in mobile environments.

Abstract

Traditional visual CAPTCHAs face dual challenges: increasing vulnerability to automated solvers and maintaining usability without imposing excessive visual burden. We propose *Haptic CAPTCHA (HAPTCHA)*, an eyes-free human-verification framework that leverages smartphone vibration as the primary information channel. HAPTCHA grounds security in a human closed-loop sensorimotor process that is inherently difficult for automated agents to emulate. We instantiate this framework through two complementary designs: *tiltHAPTCHA*, an efficiency-oriented spatial search task, and *rhythmHAPTCHA*, a robustness-oriented temporal reproduction task. Our security analysis against sophisticated imitation-learning bots shows that HAPTCHA's structured interaction traces provide a robust signal for discrimination, achieving up to 100% bot rejection rate in unseen attack scenarios. The usability evaluation across sitting, walking, and fast-walking scenarios demonstrates that HAPTCHA maintains high solvability (e.g., 96.7% success rate for tiltHAPTCHA while sitting) and offers superior usability with significantly lower cognitive workload than a visual reCAPTCHA mockup. These findings position HAPTCHA as an explicit haptic challenge layer that complements passive, risk-based verification by moving challenge information beyond the screen.

CCS Concepts

• **Security and privacy** → **Usability in security and privacy**; • **Human-centered computing** → **Interaction design process and methods**; *User interface design*; • **Hardware** → **Tactile and hand-based interfaces**.

Keywords

CAPTCHA, Vibrotactile, Haptic, Mobile, Usability, Security

This work is licensed under a Creative Commons Attribution 4.0 International License.

UIST '26, Detroit, MI, USA

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2856-3/26/11
https://doi.org/10.1145/3830398.3830484

ACM Reference Format:
Duhong Kweon, Dajin Lee, Seulbae Kim, and Seungmoon Choi. 2026. HAPTCHA: Eyes-Free Haptic CAPTCHA in Mobile Environments: Design Principle, Security Evaluation, and User Experience. In *The 39th Annual ACM Symposium on User Interface Software and Technology (UIST '26), November 02–05, 2026, Detroit, MI, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3830398.3830484

1 Introduction

CAPTCHA, "Completely Automated Public Turing test to tell Computers and Humans Apart," was established to distinguish legitimate users from automated programs using tasks that are easy for humans but difficult for machines [67]. Over the past two decades, it has been widely adopted as a standard solution for mitigating automated threats such as spam, credential stuffing, and brute-force attempts [32]. Modern human-verification pipelines strike a balance between security and usability by pairing background risk scoring with explicit fallbacks: systems evaluate passive behavioral or network signals first [25, 26] and then present an explicit interactive challenge only as an indispensable step-up layer when passive evidence is inconclusive or ambiguous [28].

However, the security of this explicit step-up layer is rapidly collapsing because challenges are almost universally rendered on the screen. Advances in artificial intelligence (AI)—particularly agentic vision-language systems—allow automated solvers to directly perceive screen content and execute interface actions across diverse schemes without task-specific adaptation [53, 64]. Simply increasing the visual diversity or complexity of screen-rendered puzzles fails to provide a durable defense while imposing disproportionate effort on legitimate users [7, 76]. While prior mobile CAPTCHA research explored alternative input modalities, including audio [66], gestures [38], and device sensors [22, 36], most still convey the challenge prompt itself visually on the screen (e.g., SenCAPTCHA [22]). Consequently, screen-parsing agents can easily extract the challenge objective and bypass verification by synthesizing artificial sensor events. Whether tactile vibration can instead serve as the primary challenge channel—moving the challenge completely off the screen while leveraging sensorimotor traces—remains underexplored.

In this paper, we present *HAPTCHA (haptic CAPTCHA)*, an explicit human-verification framework that utilizes sensorimotor interaction to complement passive or risk-based verification. HAPTCHA conveys challenges through vibrotactile stimuli and jointly evaluates task correctness and sensorimotor traces, requiring an attacker to intercept the vibration signal and reproduce plausible human execution. We instantiate the framework through *tiltHAPTCHA*, which employs a tactile feedback-guided orientation search task, and *rhythmHAPTCHA*, which utilizes a tactile rhythm perception and reproduction task.

This research aims to address the following core questions:

RQ1: Can haptic stimuli serve as a viable primary channel for human-verification challenges on smartphones?

RQ2: Do the sensorimotor interaction traces generated during HAPTCHA execution provide sufficient signals to distinguish human users from automated bots?

RQ3: To what extent do haptic-centered challenges maintain practical usability across controlled mobile conditions?

We evaluate both HAPTCHA schemes in two steps. First, we assess their security potential via human-bot trace discrimination against diverse automated attackers, including attackers that can observe vibration signals and generate human-like responses. Second, we compare their usability against a reCAPTCHA-style visual mockup [58] under sitting, walking, and fast-walking conditions. We use these mobility conditions as controlled stress tests rather than as the sole deployment scenarios for HAPTCHA. Our results suggest that HAPTCHA generates informative signals for human-bot discrimination while remaining practically usable across the evaluated conditions.

The contributions of this work are as follows:

- We propose HAPTCHA, an explicit human-verification framework that elevates tactile vibration from auxiliary feedback to the primary challenge channel, and jointly evaluates task correctness and sensorimotor execution.
- We introduce two distinct haptic challenge designs, tiltHAPTCHA and rhythmHAPTCHA, representing different trade-offs between interaction efficiency and environmental robustness.
- We evaluate the two HAPTCHA schemes against diverse vibration-aware attackers, including previously unseen attack strategies, and characterize their usability relative to a visual CAPTCHA mockup under controlled mobility conditions.

2 Related Work

2.1 CAPTCHA and Mobile Alternatives

CAPTCHA has traditionally relied on visually structured tasks such as distorted-text recognition [67, 68], image categorization [20], and other interactive visual challenges [30, 42]. Although the design space has expanded to audio [66], gestures [38], and sensor-assisted schemes [22, 36], recent work suggests that static and heavily visual CAPTCHAs remain increasingly vulnerable to modern machine-learning-based attacks [32, 58, 59, 67, 69, 70, 72, 73].

These limitations are especially pronounced on smartphones, where small displays, touch input, and divided attention make desktop-derived CAPTCHA tasks cumbersome [55]. Prior work has therefore explored more mobile-appropriate alternatives. Some approaches replace text entry with smartphone-native interaction such as gestures or device orientation, improving mobile usability but still relying on visually guided tasks [22, 38]. For instance, SenCAPTCHA [22] leverages device orientation sensing to let users tilt the device and guide a ball toward an animal's eye on the screen. In a user study, it outperformed representative text-, audio-, image-, and video-based CAPTCHA baselines in perceived usability and completion time [22]. However, they remain visually guided at their core rather than genuine eyes-free challenges.

Other work has broadened accessibility through non-visual interaction. TapCAPTCHA [3], for example, asks users to listen to a sequence of letters or numbers and tap the screen according to how many times a target item occurs, thereby avoiding visual dependence. Audio CAPTCHAs [21] replace visual puzzles with listening and transcription tasks, but such designs are often slower and less accurate than visual CAPTCHA. However, audio CAPTCHAs have also been successfully bypassed by automated solvers, limiting their viability as an alternative challenge modality [14, 61].

Our study investigates whether tactile stimulation can serve as the primary challenge channel of a mobile human-verification task under attention-constrained conditions.

2.2 Haptics in Security and Authentication

Haptic interaction has been considered useful in security-sensitive contexts because it can support relatively private, discreet, and eyes-free interaction [18]. One line of work focused on observation-resistant password and PIN entry. They used tactile patterns to support password entry while reducing the amount of visually observable information available to shoulder surfers [9–11]. VibroAuth [39] similarly explored non-visual rearranged keypads whose layout is privately conveyed through haptic patterns in order to mitigate shoulder-surfing attacks.

A second line of work emphasized accessibility-centered authentication. In Tactile Authentication System (TAS) [43, 44], users authenticate by recognizing a learned sequence of tactile patterns. OneButtonPIN [40] enables blind or low-vision users to enter PINs by counting vibration pulses on a single large button. Haptic2FA [8] uses haptic patterns as an accessible alternative to visually or auditorily delivered one-time passcodes.

Another direction has used vibration to present information to users while sensing user-specific physical responses during touch interactions [46, 74]. TouchPass [74], for example, actively generates vibration signals when a user touches the smartphone screen and analyzes how those signals are altered as they propagate through the touching finger. Rather than authenticating users based primarily on overt touch behavior, it seeks to identify user-specific physical characteristics reflected in the vibration response.

HAPTCHA builds on these foundations, but targets a different problem: public human verification rather than personal identity authentication. Our work further asks whether the sensorimotor traces produced during tactile challenge solving can themselves serve as evidence for human-bot discrimination.

2.3 Behavioral Interaction Traces for Human-Bot Discrimination

Human verification has been reframed as a matter of whether a task is answered correctly, as well as of how the answer is produced. This perspective is grounded in *behavioral biometrics*, which argues that human actions carry distinctive information in learned skills, style, and motor behavior [75]. Recent surveys on mouse dynamics similarly describe interaction traces as manifestations of the perceptual, cognitive, and motoric regularities that are difficult to imitate perfectly [41]. In other words, temporal structure, movement variability, and online correction can be informative security signals in addition to final correctness.

Practical applications of this behavioral approach are now widespread. Deployed systems like Google's reCAPTCHA v2 [25] no longer rely solely on solving a puzzle; instead, they employ invisible, adaptive risk analysis that scrutinizes mouse trajectories and click timing. Similarly, BeCAPTCHA [2] showed that a simple drag-and-drop task can yield useful human-bot discrimination signals when touchscreen dynamics are analyzed together with smartphone sensor streams, such as accelerometer and gyroscope data. Supporting this direction, HuMIdb [1] introduced a multimodal mobile dataset comprising 14 smartphone sensors from 600 users and argued that such heterogeneous sensor streams capture meaningful properties of human-mobile interaction beyond task outcome alone.

Instead of focusing on touch, mouse, or passive device motion alone, HAPTCHA investigates whether haptic-guided sensorimotor traces generated while solving a tactile challenge can provide informative evidence for human-bot discrimination.

3 HAPTCHA

3.1 Design Principles

Unlike conventional CAPTCHAs that rely on static visual recognition, HAPTCHA requires users to engage in a continuous process of perception and motor response. As illustrated in Figure 1, the verification process consists of a dual-layered validation. First, the system evaluates task accuracy (Step 1) by checking whether the user's final response falls within the predefined task completion criteria. Second, the system analyzes the interaction process (Step 2) to determine whether the behavioral data exhibits the sensorimotor characteristics unique to legitimate human users. A successful human verification is granted only when the user both solves the challenge and produces a plausible human interaction trace.

We defined four primary goals to ensure HAPTCHA is both secure and user-friendly for mobile environments:

(1) **Mobile-first & Commodity-ready:** The system must utilize standard smartphone hardware (e.g., eccentric rotating mass (ERM) motors, linear resonant actuators (LRAs), and standard IMUs) without requiring specialized accessories.
(2) **Human Solvability:** Tasks should be intuitive for general users, requiring minimal cognitive load and no prior training.
(3) **Behavioral Resistance:** Security should be grounded in the interaction process (how the user solves it) rather than solely the final answer, making it difficult for bots to mimic human-specific kinematic and temporal traces.
(4) **Reduced Visual Burden:** Interaction should be driven primarily by touch and motion, minimizing the need for sustained visual attention in mobile settings.

Based on these principles, we propose a framework comprising two complementary variants: tiltHAPTCHA, which focuses on sensorimotor coupling through continuous feedback-guided spatial search, and rhythmHAPTCHA, which emphasizes the temporal motor reproduction of perceived tactile patterns.

3.2 tiltHAPTCHA

Figure 2 illustrates the concept of tiltHAPTCHA. The system presents a search task where the user tilts the smartphone to find a hidden target angle (θ_{target}) that yields the maximum vibration intensity. As the user rotates the device, the system provides continuous haptic feedback based on the angular distance $d(^\circ) = |\theta_{current} - \theta_{target}|$, where the vibration intensity is defined as:

$$I(d) = \begin{cases} 1, & d \le 6^\circ \\ 0.6\cos\left(\frac{\pi d}{180}\right), & 6^\circ < d < 90^\circ \\ 0, & d \ge 90^\circ \end{cases}. \quad (1)$$

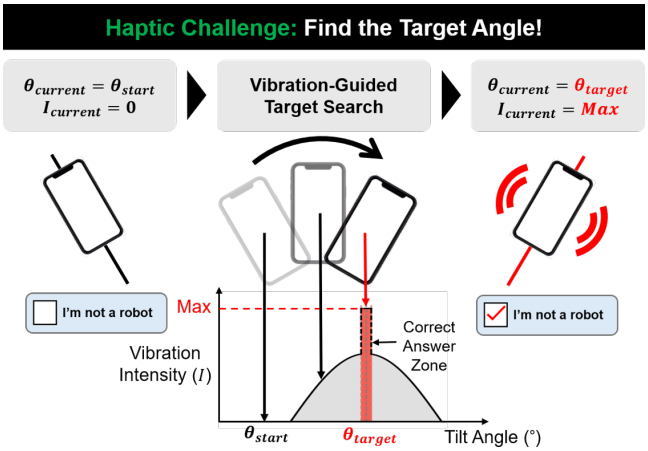


Figure 2: tiltHAPTCHA.

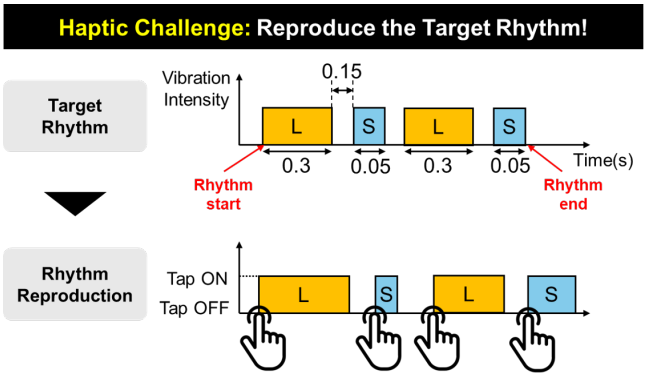


Figure 3: rhythmHAPTCHA.

Note that the flat maximum region for $d \leq 6°$ includes a discontinuity to make the peak more perceptually salient.

This design induces closed-loop sensorimotor behavior. We posit that legitimate users do not reach the target angle in a single, discrete step. Instead, they perform an active search—probing directions, slowing down near the peak, and making fine-grained corrective adjustments. These kinematic features (e.g., velocity profiles and overshoot patterns) can form a behavioral signature that is highly difficult for automated agents to reproduce, even if they know the target angle.

3.3 rhythmHAPTCHA

Figure 3 illustrates the concept of rhythmHAPTCHA, which introduces a perception and reproduction task. The system presents a structured vibration pattern composed of multiple pulses and gaps. The user must perceive this rhythm, retain its temporal structure in short-term memory, and reproduce it by tapping on the screen.

The security of rhythmHAPTCHA relies on the temporal and interactional variability of human motor output. While a bot might produce perfectly timed pulses, human tapping exhibits structured latency and subtle inconsistencies that are nevertheless constrained by the perceived rhythm. By analyzing features such as inter-tap intervals and touch-location consistency, the system distinguishes authentic human rhythmic behavior from rigid, rule-based, or statistically modeled synthetic patterns.

4 Security Evaluation

We evaluate the security of HAPTCHA by framing it as a discrimination task between human and bot traces. A *trace* denotes a time-series record of user actions and sensed signals, including device orientation, IMU data, and touch events captured during task execution. For each HAPTCHA, we collect traces from legitimate users and generate bot traces using diverse automated agents. We then train and evaluate a binary classifier whose objective is to distinguish human traces from bot-generated traces. Our goal is to assess whether the unique sensorimotor characteristics of haptic interaction provide a robust basis for distinguishing human behavior from automated imitations, even when attackers have full knowledge of the system.

For evaluation, we implemented HAPTCHA in Swift using Xcode. Haptic stimuli were generated using Apple's Core Haptics framework. We used an iPhone 12 Pro, and the maximum vibration intensity was 2.3 m/s^2 RMS with a frequency of 123 Hz (measured using an accelerometer). While this setup provides a baseline for evaluating HAPTCHA's security performance, cross-device generalizability remains an important direction, as discussed in Section 6.

4.1 Threat Model

We assume a physically augmented, high-capability attacker model to establish a conservative upper bound on HAPTCHA's security. The attacker's capabilities and constraints are defined as follows:

(1) **Standard OS Constraints:** We assume the attacker operates within the security boundaries of modern mobile operating systems (e.g., iOS and Android). These platforms only expose read-only APIs for motion sensors, such as Apple's Core Motion framework [5] and Android's SensorManager [29]. Consequently, in the absence of OS-level privileges, adversaries cannot directly inject forged sensor values into the system.
(2) **Augmented Attacker Capabilities:** To ensure an evaluation under realistic adversarial capabilities, we consider an attacker with capabilities beyond the standard application sandbox. We assume that the attacker can automate touch interactions (e.g., via Accessibility Services [4] or external hardware) and measure vibration signals—not only through a co-located on-device accelerometer, as motivated by prior accelerometer side-channel attacks [50], but also via external sensors placed in the physical environment. By granting such augmented capabilities, we ensure that HAPTCHA's security does not rely merely on restricting access to auxiliary hardware or input automation.
(3) **Open Design Principle:** In accordance with the Open Design principle [57], we assume that the attacker has the full knowledge of the algorithms, implementation details, and task structure of HAPTCHA. The security of HAPTCHA thus relies not on "security through obscurity," but on the fundamental difficulty of reproducing the complex, structured sensorimotor process of a human user.
(4) **No Direct Event Access:** Reflecting the isolation enforced by mobile operating systems, we assume that the attacker does not

Table 1: Security performance of tiltHAPTCHA.

Setting	Attacker	Feature	AUC	Acc.	Pre.	HAR	BRR
In-Domain	All (Mean)	Kinem.	1.00	0.99	1.00	0.99	100.0
		K-Sens.	1.00	1.00	1.00	0.99	100.0
LOBO	**Heuristic**	Kinem.	0.91	0.54	0.18	0.99	**6.93**
		K-Sens.	1.00	0.99	1.00	0.99	**100.0**
	ARX	Kinem.	0.99	0.96	0.90	0.99	92.38
		K-Sens.	1.00	0.99	1.00	0.99	100.0
	LSTM	Kinem.	1.00	0.99	1.00	0.99	100.0
		K-Sens.	1.00	0.99	1.00	0.99	100.0
	GAIL	Kinem.	0.99	0.92	0.81	0.99	86.79
		K-Sens.	1.00	0.99	1.00	0.99	100.0
LOHO	All	Kinem.	1.00	1.00	1.00	0.99	100.0
		K-Sens.	1.00	1.00	1.00	0.99	100.0

* Kinem.: Kinematic-only, K-Sens.: Kinematic and sensor fusion.
* AUC (Area Under the ROC Curve): Discriminative ability between classes (0–1).
* HAR (Human Acceptance Rate): Proportion of humans correctly accepted.
* BRR (Bot Rejection Rate): Proportion of bots correctly rejected.

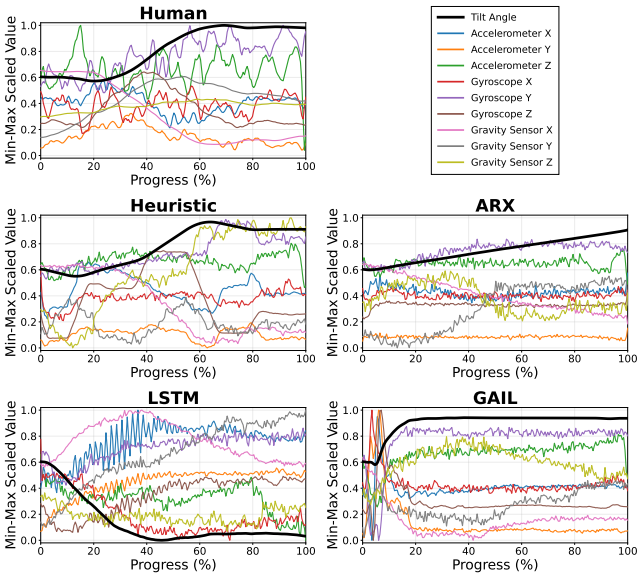


Figure 4: Comparison of feature trajectories between human users and bots (Target angle: 30°).

have access to internal OS-level events and application states (e.g., vibration-trigger timestamps). Instead, the attacker must infer the challenge state through externally observable physical cues. This constraint aligns the attacker's perception with that of a human user, who relies on sensorimotor feedback rather than privileged access to internal system states.

4.2 tiltHAPTCHA

4.2.1 Human Data Collection. We collected behavioral traces from 12 participants (6 females, 6 males; aged 19–35 years) under three mobility conditions: sitting, walking (Preferred Walking Speed, PWS), and fast walking (130% of PWS). This data-collection study was approved by the Institutional Review Board at POSTECH (**PIRB-2025-E010**).

We recorded time-series signals, including device orientation, inertial data (gyroscope, accelerometer, and gravity), vibration, and the confirmation touch event. We set the target yaw angles from −30° to 90° in 5° increments, which reflects the functional asymmetry of wrist radial and ulnar deviation during handheld use [52, 56]. Each condition was measured five times.

4.2.2 Attacker Modeling. To simulate a sophisticated adversary capable of human-like sequential interaction, we modeled the attacker as a state-based agent that determines the next rotational action at runtime. Each bot takes the current tilt angle and observed vibration value as input and computes the direction and magnitude of its next movement as output. This closed-loop approach ensures that the bots actively respond to the system's haptic feedback, mimicking the human search process. To instantiate this model, we implemented four distinct bot architectures, spanning a broad and representative spectrum of automated attack strategies:

- **Heuristic (Feedback-Driven):** A handcrafted search bot that uses real-time vibration feedback to approach the target. It mimics human search strategies by probing both directions, moving toward stronger feedback, slowing down near the peak, and performing minor corrective recentering before confirming.
- **ARX (Autoregressive with Exogenous Variables) [47]:** A linear predictive bot that determines the next yaw increment based on recent movement history and exogenous vibration feedback (past intensities and differences). Trajectories are generated recursively from the model's own previous output values.
- **LSTM (Long Short-Term Memory) [35]:** A recurrent neural network-based bot that predicts the next yaw movement and response probability from the current state and accumulated temporal context. Unlike ARX, it can capture non-linear sequential patterns and longer-range dependencies.
- **GAIL (Generative Adversarial Imitation Learning) [34]:** An imitation learning bot that learns a policy to generate action sequences mimicking observed human search behavior. A discriminator is trained to distinguish generated trajectories from human ones, forcing the policy to reproduce human-like kinematic structures.

We also synthesized corresponding motion-sensor streams for each bot by applying a regression mapping trained on human data to the bot-generated rotational trajectories. The mapping was learned from the relationship between rotational trajectories and the accompanying motion-sensor measurements in the human traces. The resulting sensor streams were combined with the bot-generated trajectories for the sensor-augmented evaluation.

4.2.3 Evaluation Protocol. To select the best performing binary classifier for the human-versus-bot discrimination task [2], we conducted a preliminary experiment using episode-level behavioral features. We evaluated the performance of K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF) classifiers. Consistent with prior work on sensor and activity recognition using engineered features [6, 16], RF, a practical non-linear

model for feature-based behavioral discrimination, outperformed the other classifiers and was therefore selected as the final model.

Then, we evaluated the RF classifier under three settings designed to probe both in-domain performance and two forms of generalization: (1) *in-domain*, where bot types seen during training could also appear in testing; (2) *leave-one-bot-out (LOBO)*, which tests generalization to an unseen bot type [19]; and (3) *leave-one-human-out (LOHO)*, which tests generalization to an unseen human participant, analogous to leave-one-subject-out evaluation [15]. Decision thresholds were selected on the validation set to maintain a ≥99% Human Acceptance Rate (HAR)[1]. The observed HAR on the test set may deviate from the initial target due to sampling variance. This deviation can be more pronounced in the LOBO and LOHO settings, as these evaluations reflect the model's generalization to unseen bot types or human participants.

We also compared two feature sets to analyze which aspects of the interaction provide the strongest security signals. The first set is *kinematic-only*, focusing on the temporal and trajectory characteristics of the rotation. The second set is *kinematic and sensor fusion*, which additionally incorporates motion-sensor features derived from accelerometer, rotation rate, and gravity signals.

4.2.4 Results and Discussion. Table 1 summarizes the security performance of tiltHAPTCHA. In the known-bot setting (In-Domain and LOHO), both feature configurations achieved near-ceiling performance, indicating that kinematic features alone are sufficient to distinguish human traces from bot-driven attack patterns represented during training.

However, a critical divergence emerged in the unseen-bot setting (LOBO). While the kinematics-only configuration rejected only 6.93% of the previously unseen Heuristic bot, the kinematic-sensor fusion configuration achieved 100.00% BRR (Bot Rejection Rate) against the same bot. This result highlights that authentic human performance involves a tight cross-modal consistency between physical rotation and motion-sensor responses, which automated agents struggle to synthesize even with regression-based sensor generation. This observation is consistent with BeCAPTCHA [2], which demonstrated that multimodal data from touchscreen interactions (e.g., drag and drop) combined with accelerometer data can effectively model human behavior for bot detection.

In the unseen-human setting (LOHO), both configurations maintained high overall performance, confirming that the system does not over-fit to idiosyncratic patterns of training participants. However, the kinematic-sensor fusion configuration exhibited a more conservative decision boundary, with one participant's HAR dropping to 89.07%. This suggests that the primary contribution of motion-sensor features lies in rejection robustness against novel attacks, potentially at a marginal cost to the acceptance of users with less common interaction patterns.

Figure 4 compares the rotational trajectories and other sensor traces of human users and bots when the target angle is 30°. To facilitate visual comparison across features with disparate scales, we applied min-max normalization to the ensemble-averaged trajectories (see Figure 9 for original y-axis scales). This comparison further reveals the behavioral flaws of the bot architectures. Notably, the LSTM bot begins to diverge in the incorrect direction as early as the 25% progress mark. This behavior is a clear manifestation of exposure bias in autoregressive generation; once an initial directional error occurs, it is recursively reinforced by subsequent predictions. This creates a feedback loop of incorrect trajectories that drift further from the target, whereas humans utilize haptic feedback to execute real-time corrections. In contrast, the ARX model's linear structure prevents such non-linear divergence. However, its resulting paths remain overly simplistic and lack the nuanced kinematic signatures of human search.

[1]Human Acceptance Rate (HAR) is the proportion of human instances correctly classified as human. A higher value indicates better usability for legitimate users.

Table 2: Security performance of rhythmHAPTCHA.

Setting	Attacker	Feature	AUC	Acc.	Pre.	HAR	BRR
In-Domain	All (Mean)	Temporal	0.87	0.79	0.77	0.96	61.98
		Spatiotemporal	1.00	0.97	1.00	0.95	98.87
LOBO	**Naive**	Temporal	1.00	0.99	0.99	0.96	99.85
		Spatiotemporal	1.00	0.99	0.99	0.95	99.87
	Gaussian	Temporal	0.96	0.85	0.60	0.96	82.04
		Spatiotemporal	1.00	0.99	0.99	0.95	99.60
	KDE	Temporal	0.51	0.24	0.22	0.96	**3.42**
		Spatiotemporal	0.99	0.96	0.87	0.94	**96.14**
	GMM	Temporal	0.62	0.26	0.22	0.95	**5.78**
		Spatiotemporal	0.99	0.94	0.82	0.95	**94.19**
	GAN	Temporal	0.90	0.67	0.40	0.96	58.50
		Spatiotemporal	1.00	0.98	0.98	0.95	99.54
	Adv. Opt.	Temporal	0.99	0.96	0.88	0.96	96.37
		Spatiotemporal	0.99	0.98	0.96	0.95	98.81
LOHO	All	Temporal	0.84	0.67	0.10	0.88	66.06
		Spatiotemporal	0.99	0.99	0.87	0.94	99.80

4.3 rhythmHAPTCHA

We applied the same evaluation protocol to rhythmHAPTCHA by collecting human traces, constructing multiple attack bots, and measuring human-versus-bot discrimination performance.

4.3.1 Human Data Collection. Similar to the case of tiltHAPTCHA, we recruited another 12 participants (6 males, 6 females; aged 20–34 years). We also used the same three mobility conditions: sitting, walking, and fast walking. This data-collection study was approved by the Institutional Review Board at POSTECH (**PIRB-2024-E005**).

A total of 30 rhythm patterns, ranging from 1 to 4 beats in length, were presented to the participants. Each pattern consisted of Long (L, 0.3 s) or Short (S, 0.05 s) pulses with a fixed inter-beat interval of 0.15 s. In each trial, participants were tasked to perceive a target vibration rhythm and then reproduce it by tapping the screen. During reproduction, we recorded touch onset times, touch durations, and inter-touch intervals to capture temporal characteristics. We also recorded normalized touch coordinates to capture spatial interaction characteristics.

4.3.2 Attacker Modeling. Under our threat model, the attacker infers the rhythm pattern from physical vibrations sensed via an accelerometer and generates a corresponding touch sequence. To simulate diverse imitation strategies, we implemented six bots:

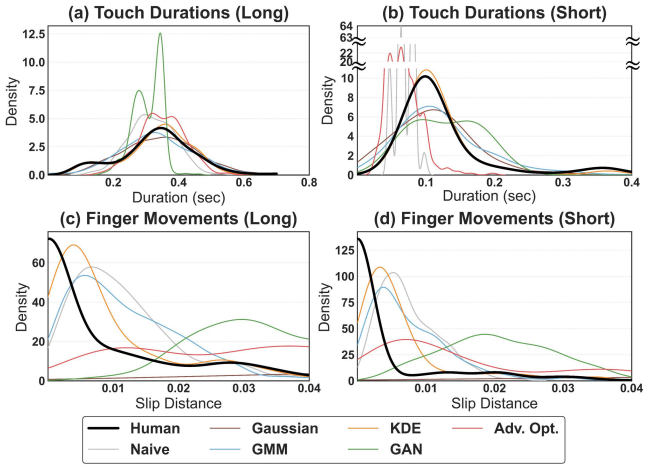


Figure 5: Comparison of touch durations and movements between human users and bots (Target rhythm: Long-Short).

- **Naive:** A rule-based bot that directly maps detected vibration durations and gaps to touch timing while sampling coarse touch-down and touch-up locations from human-derived coordinate statistics.
- **Statistical (Gaussian, KDE, GMM):** Bots that sample interaction features from distributions estimated from human data, including temporal variables (e.g., latency, duration, and inter-touch interval) and, when available, normalized touch-location features, using Gaussian, Kernel Density Estimation (KDE) [60], or Gaussian Mixture Models (GMM) [13].
- **GAN (Generative Adversarial Network) [24]:** A generative model that learns the joint distribution of human interaction features to synthesize novel touch sequences that capture complex dependencies across both temporal and spatial patterns.
- **Adversarial Optimization:** Starting from naive bot samples, this bot iteratively updates interaction features using gradients from a learned discriminator so that the samples receive a higher human-like score. In other words, it performs a white-box search in feature space to make bot traces harder to distinguish from human traces [12].

All bot timing values were quantized to a 60 Hz grid to ensure that the bot's temporal precision aligns with the discrete sampling behavior of standard smartphone touch-sampling hardware.

4.3.3 Evaluation Protocol. We apply the same evaluation protocol described in Section 4.2.3 to rhythmHAPTCHA. We report the performance under the same *in-domain*, *LOBO*, and *LOHO* settings. Decision thresholds are selected to maintain a ≥95% Human Acceptance Rate (HAR). We consider two feature configurations: *Temporal-only*, using timing statistics, and *Spatiotemporal*, which additionally incorporates touch location features.

4.3.4 Results and Discussion. Table 2 summarizes the comprehensive security performance of rhythmHAPTCHA. A key result is that temporal-only features were particularly vulnerable to KDE- and GMM-based bots (3.42% and 5.78% BRR, respectively). This highlights that when security relies solely on coarse temporal statistics, it is susceptible to bots capable of modeling the empirical shape of human timing variability.

Table 3: Security performance against attacks constructed from the expanded human corpus.

CAPTCHA	Attacker	Feature	AUC	Acc.	Pre.	HAR	BRR
Tilt	Heuristic	Kinem.	0.82	0.52	0.53	0.97	**1.36**
		K-Sens.	1.00	0.98	1.00	0.96	**100.0**
	ARX	Kinem.	1.00	0.98	1.00	0.97	100.0
		K-Sens.	1.00	0.98	1.00	0.96	100.0
	LSTM	Kinem.	1.00	0.98	1.00	0.97	100.0
		K-Sens.	1.00	0.98	1.00	0.96	100.0
	GAIL	Kinem.	1.00	0.98	1.00	0.97	100.0
		K-Sens.	1.00	0.98	1.00	0.96	100.0
Rhythm	Naive	Temporal	1.00	0.94	1.00	0.81	99.82
		Spatiotemp.	1.00	0.98	1.00	0.95	99.95
	Gaussian	Temporal	0.95	0.89	0.85	0.81	93.14
		Spatiotemp.	1.00	0.98	0.99	0.95	99.44
	KDE	Temporal	0.65	0.53	0.39	0.81	**40.44**
		Spatiotemp.	0.99	0.95	0.90	0.95	**95.30**
	GMM	Temporal	0.72	0.61	0.43	0.81	**51.21**
		Spatiotemp.	0.99	0.95	0.90	0.95	**95.15**
	GAN	Temporal	0.89	0.83	0.71	0.81	84.36
		Spatiotemp.	1.00	0.97	0.96	0.95	97.95
	Adv. Opt.	Temporal	0.99	0.92	0.93	0.81	97.25
		Spatiotemp.	1.00	0.98	0.98	0.95	99.21

In contrast, the spatiotemporal configuration restored high robustness across all attack vectors, consistently achieving >94% BRR even against previously unseen (LOBO) bots. This indicates that human rhythm reproduction is not merely a timing task but a coordinated motor act with a distinct spatial signature.

To shed light on this robustness, Figure 5 compares the tapping durations and finger movements between human users and bots when the target rhythm is Long-Short. While bots can be tuned to match temporal marginals (Figures 5a and 5b), replicating the spatiotemporal coordination (Figures 5c and 5d) remains a non-trivial challenge for the current agents. Legitimate users exhibited biomechanical constraints—specific relationships between "when" they tap and "where" they tap relative to the rhythm's tempo.

It is worth noting that even in the unseen-human setting (LOHO), the spatiotemporal configuration maintained a strong average HAR of 93.61%, significantly outperforming the temporal-only baseline (88.00% HAR) in overall discrimination (99.80% BRR). This reconfirms that integrating spatial interaction provides essential complementary information that is more consistent across different human users than timing alone.

4.4 Robustness to Expanded Attacker Training Data

The preceding security evaluation constructed attacks using human traces collected from 12 participants. To examine the effect of

increased attacker training data, we collected traces from 12 additional participants and expanded the attacker-accessible human corpus from 12 to 24 participants. This additional data collection was approved by the Institutional Review Board at POSTECH (**PIRB-2025-E047**). The classifiers and decision thresholds obtained using the original 12-participant corpus were retained without retraining or recalibration. We then trained or parameterized the same attack families using the traces from all 24 participants and evaluated them against the fixed classifiers. HAR was also evaluated on the 12 new participants, who were unseen by the fixed classifiers. Table 3 summarizes the results.

The expanded corpus affected the attack families differently: several rhythm attacks showed higher BRRs than in the original in-domain evaluation, whereas the Heuristic tilt attack produced a lower BRR against the kinematic-only configuration. Expanding the corpus increased both the amount and behavioral diversity of the available human data and produced different changes in attack effectiveness. Nevertheless, the combined-feature configurations achieved BRRs of 100% for tiltHAPTCHA and 97.84% for rhythmHAPTCHA. These results provide additional evidence of robustness to a broader attacker training corpus, while larger-scale evaluations remain necessary to characterize how performance changes with substantially larger datasets.

5 Usability Evaluation

We conducted a user study to evaluate the usability of tiltHAPTCHA and rhythmHAPTCHA across multiple mobility conditions. We included walking conditions to approximate mobile use scenarios involving divided visual attention. The HAPTCHAs were implemented on the same device (iPhone 12 Pro) used in Section 4.

5.1 Methods

5.1.1 Participants. A total of 24 right-handed participants (12 females, 12 males; aged 18–33 years) took part in this study. They did not participate in the human data collection described in Section 4. This usability study was approved by the Institutional Review Board at POSTECH (**PIRB-2025-E047**).

5.1.2 Experimental Conditions. The experiment employed a within-subjects design with two factors: Captcha and Mobility. We evaluated three CAPTCHAs: *reCAPTCHA* (Baseline), *tiltHAPTCHA*, and *rhythmHAPTCHA*. To ensure consistency, we used a high-fidelity mockup of reCAPTCHA, as the live version's adaptive difficulty is unsuitable for controlled repeated-measures comparisons. Accordingly, our study evaluates the user-facing explicit challenges presented when verification is triggered, rather than the end-to-end production reCAPTCHA pipeline.

The CAPTCHAs were tested under three mobility conditions: *Sitting*, *Walking* (Preferred Walking Speed, PWS), and *Fast walking* (130% of PWS). For the walking conditions, treadmill speed was individually calibrated to ensure consistent exertion levels. To mitigate order effects, we employed a 4×6 counterbalanced design. We defined four CAPTCHA-order schemes—positioning the baseline either first or last while counterbalancing the two HAPTCHAs—and crossed them with all six permutations of the three mobility conditions. Each of the 24 participants was assigned a unique sequence.

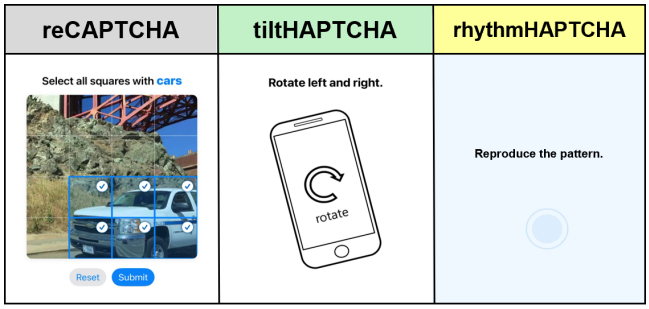


Figure 6: User interfaces of the three CAPTCHA tasks.

5.1.3 Task and Procedure. The study was conducted in a quiet indoor space equipped with a treadmill. Participants were first introduced to the three CAPTCHA tasks shown in Figure 6 and were briefed on the rules for each task. They then completed a short practice session prior to the main experiment. Participants then completed 9 blocks (3 Captchas × 3 Mobilities), presented in the assigned order combination, with 15 trials each. In each trial, an auditory cue (beep sound) prompted the participant to begin the task. After each block, participants completed a corresponding subjective questionnaire. At least a 1-minute break was provided between blocks to reduce fatigue.

5.1.4 Measures. We employed objective and subjective measures to evaluate the effectiveness and user experiences. Success rate and completion time were automatically logged as objective measures.

Subjective measures were answered on a 0–100 scale across three dimensions: usability, cognitive workload, and mobile burden. First, usability was assessed via *Usefulness*, *Ease of Use*, *Ease of Learning*, *Satisfaction*, and *Enjoyment* (the first four adapted from the USE questionnaire [49]). Second, cognitive workload was evaluated using *Task Difficulty*, *Mental Effort*, *Performance*, and *Frustration*; the latter three are standard NASA-TLX sub-scales [33]. Finally, for the two walking conditions, mobile burden was scored via *Visual Demand* (effort to monitor the screen), *Gait Interference* (disruption to walking rhythm), and *Safety Risk* (perceived danger).

After experiencing all CAPTCHAs within a mobility condition, participants ranked their preferences assuming real-world deployment. Optional free-text comments were also collected.

5.2 Hypotheses

We formulated the following hypotheses:

- **H1 (Solvability):** HAPTCHA will achieve a high success rate while sitting, demonstrating fundamental human solvability.
- **H2 (Mobile Practicality):** HAPTCHA will maintain practical performance during locomotion, proving effective in divided-attention scenarios.
- **H3 (Distinct Usability):** The HAPTCHA variants will exhibit distinct usability profiles, representing complementary design trade-offs.

5.3 Results

We conducted the Shapiro-Wilk test to check the normality assumptions, but all measures exhibited significant ceiling effects or were

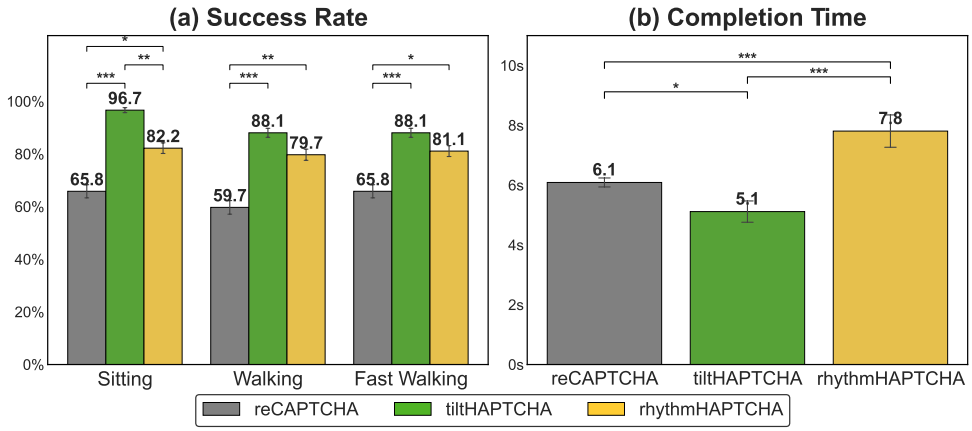


Figure 7: Objective measures of (a) Success rate and (b) Completion time. The error bars represent the standard errors.

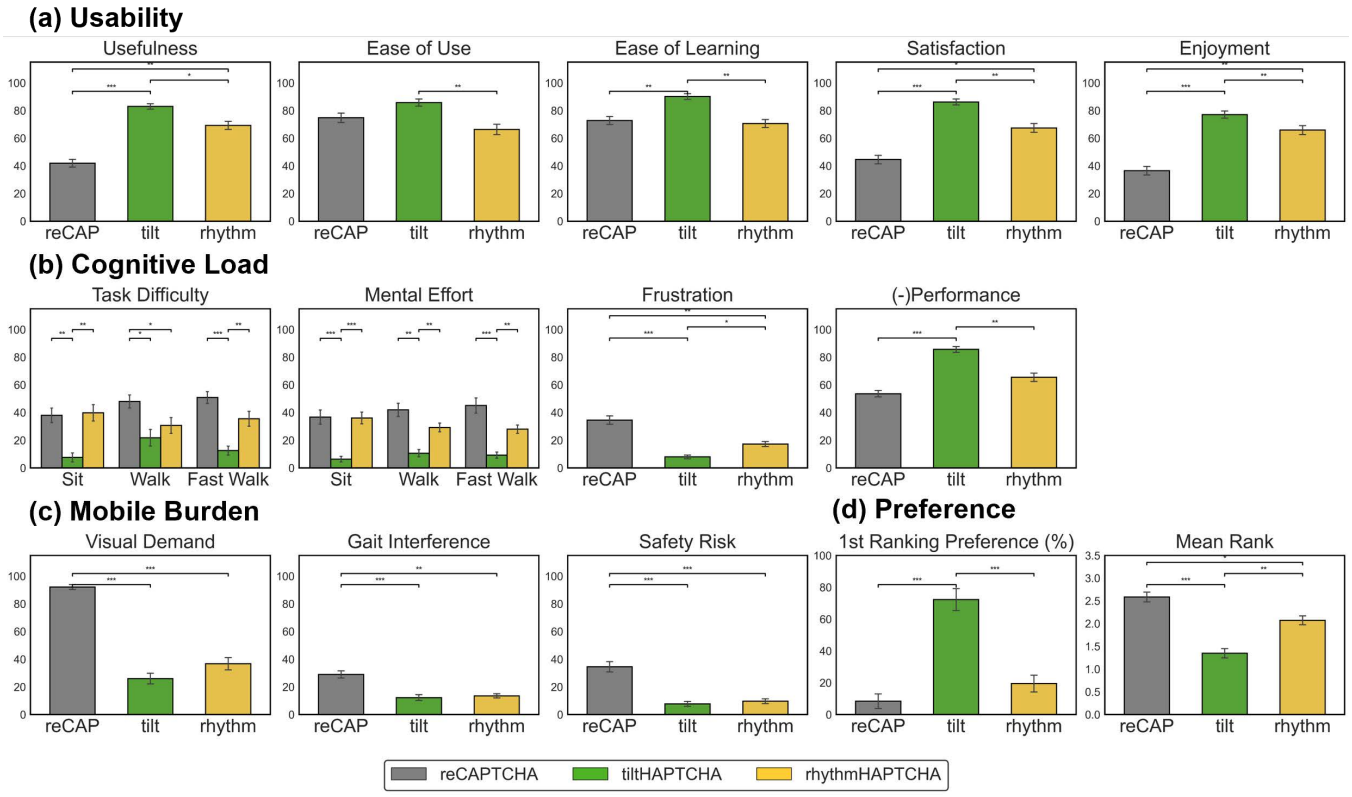


Figure 8: Subjective measures of (a) Usability, (b) Cognitive load, (c) Mobile burden, and (d) Preference. The error bars represent the standard errors.

highly skewed ($p < .05$). We therefore applied an Aligned Rank Transform (ART) ANOVA [71] for a full within-subjects test with two independent factors of Captcha and Mobility for each metric. Significant effects were further analyzed using post-hoc pairwise comparisons with Bonferroni correction.

5.3.1 Objective Performance. Figure 7 shows the means and standard errors of the objective measures. Captcha effects were significant on both *Success Rate* ($F(2, 46) = 45.27, p < .001$) and *Completion Time* ($F(2, 46) = 39.48, p < .001$). Post-hoc comparisons for *Success Rate* revealed significant differences among all three CAPTCHA types: reCAPTCHA < tiltHAPTCHA ($p < .001$), reCAPTCHA < rhythmHAPTCHA ($p = .012$), and tiltHAPTCHA >

rhythmHAPTCHA ($p = .005$). For *Completion Time*, all pairwise comparisons were also significant: reCAPTCHA > tiltHAPTCHA ($p = .016$), reCAPTCHA < rhythmHAPTCHA ($p < .001$), and tiltHAPTCHA < rhythmHAPTCHA ($p < .001$). tiltHAPTCHA achieved both the highest success rate and the shortest completion time. rhythmHAPTCHA also outperformed reCAPTCHA in success rate, but required the longest completion time. These results support H1 (Solvability).

Mobility had a significant main effect on *Success Rate* ($F(2, 46) = 8.07, p = .001$), but not on *Completion Time* ($F(2, 46) = 1.05, p = .358$). Post-hoc analysis showed a significant difference in *Success Rate* only between *Sitting* and *Walking* ($p = .006$).

The interaction effect of Captcha × Mobility was significant for *Success Rate* only ($F(4, 92) = 3.32, p = .014$). Under *Sitting*, all pairwise differences among the three CAPTCHAs were significant. Under *Walking* and *Fast Walking*, both tiltHAPTCHA and rhythmHAPTCHA significantly outperformed reCAPTCHA, whereas the difference between the two HAPTCHA variants was not significant. This pattern indicates that both HAPTCHAs maintained better performance than the baseline even under locomotion, supporting H2 (Mobile Practicality).

Regarding H3 (Distinct Usability), the results did not indicate a balanced trade-off between the two HAPTCHAs. Rather, tiltHAPTCHA consistently showed the strongest overall objective performance, achieving both the highest success rate and the shortest completion time. Although rhythmHAPTCHA also outperformed reCAPTCHA in success rate, particularly under locomotion, it required significantly longer completion times than tiltHAPTCHA. Thus, from an objective-performance perspective, tiltHAPTCHA emerged as the most effective design overall.

5.3.2 Subjective Usability. As shown in the mean ratings in Figure 8a, Captcha significantly affected all usability dimensions: *Usefulness* ($F(2, 46) = 33.55, p < .001$), *Ease of Use* ($F(2, 46) = 9.20, p < .001$), *Ease of Learning* ($F(2, 46) = 14.76, p < .001$), *Satisfaction* ($F(2, 46) = 35.59, p < .001$), and *Enjoyment* ($F(2, 46) = 41.25, p < .001$). Post-hoc comparisons revealed that tiltHAPTCHA achieved the highest ratings on all dimensions. rhythmHAPTCHA outperformed reCAPTCHA in *Usefulness*, *Satisfaction*, and *Enjoyment* but was less effective than reCAPTCHA in *Ease of Use*.

Mobility showed a significant main effect only on *Ease of Use* ($F(2, 46) = 3.31, p = .045$). No significant Captcha × Mobility interactions were found for the usability measures.

5.3.3 Cognitive Workload. As shown in Figure 8b, Captcha significantly affected all workload dimensions, including *Task Difficulty* ($F(2, 46) = 33.46, p < .001$), *Mental Effort* ($F(2, 46) = 37.66, p < .001$), *Frustration* ($F(2, 46) = 21.90, p < .001$), and *Performance* ($F(2, 46) = 29.06, p < .001$). Post-hoc tests indicated that tiltHAPTCHA yielded the lowest workload and the highest perceived performance, while rhythmHAPTCHA significantly reduced frustration relative to reCAPTCHA.

Mobility had a significant main effect only on *Frustration* ($F(2, 46) = 3.83, p = .029$). Interestingly, the interaction effects of Captcha × Mobility were significant for *Task Difficulty* ($F(4, 92) = 3.31, p = .014$) and *Mental Effort* ($F(4, 92) = 3.47, p = .011$). reCAPTCHA and rhythmHAPTCHA showed comparable workload levels in the *Sitting* condition. However, rhythmHAPTCHA became significantly less demanding than reCAPTCHA as mobility intensity increased.

5.3.4 Mobile Burden. Figure 8c summarizes the mobile burden ratings under the two walking conditions. Captcha significantly affected *Visual Demand* ($F(2, 46) = 91.66, p < .001$), *Gait Interference* ($F(2, 46) = 26.73, p < .001$), and *Safety Risk* ($F(2, 46) = 38.45, p < .001$). Post-hoc comparisons showed that reCAPTCHA was rated significantly higher than both rhythmHAPTCHA and tiltHAPTCHA on all three measures. This suggests that our HAPTCHA designs were perceived as less disruptive and as posing lower safety risk during the tested walking conditions.

The effects of Mobility and Captcha × Mobility were not significant for all measures.

5.3.5 User Preference. Figure 8d shows the proportion of the first rank in preference and the mean rank (MR; 1.0 indicates the highest preference). For the first rank proportion, the effect of Captcha was significant ($F(2, 46) = 24.03, p < .001$). tiltHAPTCHA was the most preferred. However, Mobility ($F(2, 46) = 0.00, p = 1.000$) and the interaction effect ($F(4, 92) = 1.70, p = .157$) was not significant.

For the mean rank, Captcha again showed a significant main effect ($F(2, 46) = 24.29, p < .001$), whereas neither Mobility ($F(2, 46) = 0.00, p = 1.000$) nor their interaction effect ($F(4, 92) = 1.70, p = .157$) was significant. tiltHAPTCHA was the most preferred, while rhythmHAPTCHA was preferred over reCAPTCHA.

5.4 Discussion

5.4.1 Haptic Solvability: Beyond Visual Recognition. Our findings demonstrate that haptic-motor interaction is a highly viable alternative to traditional visual CAPTCHAs. While prior orientation-based schemes (e.g., SenCAPTCHA [22]) used mobile sensors, they relied primarily on visual feedback. In contrast, tiltHAPTCHA's high success rate (96.7%) and efficiency (4.64 s) indicate that closed-loop haptic feedback can effectively replace visual guidance.

This aligns with theories of embodied cognition [23, 63], where haptic cues offload cognitive tasks from the visual channel without performance loss [17, 45]. Furthermore, we posit that this sensorimotor search process provides a more robust signal of human presence than static visual tasks. It requires a tight coupling between perception and action, which is inherently difficult for automated bots to emulate in a non-visual domain, as discussed in Section 4.

5.4.2 Alleviating Visual Dependency During Locomotion. This study provides empirical evidence regarding mobility's impact on verification. The baseline, reCAPTCHA, imposes strong visual dependencies: as locomotion increases, the visual attention required to monitor the screen may compete with attention to the surrounding environment. In contrast, HAPTCHA can mitigate this by offloading the challenge to the underutilized haptic channel. Our data suggest that HAPTCHA maintains practical performance during locomotion while imposing lower visual demand, gait interference, and perceived safety risk than the visual baseline. This indicates that eyes-free embodied interaction can provide a promising approach to mobile verification when visual attention is divided.

5.4.3 Design Profiles: Efficiency and Stability Across Mobility Conditions. The two HAPTCHA variants exhibited distinct usability profiles. tiltHAPTCHA supports a direct sensorimotor search where the mapping between orientation and feedback facilitates rapid alignment. Conversely, rhythmHAPTCHA maintained stable accuracy across locomotion levels, though completion times were inherently longer (7.67–7.97 s) due to the necessity of perceiving and reproducing a temporal sequence. Our findings suggest that internal temporal processing is highly resilient to the physical tremors of walking, remaining stable across varying activities [37, 54].

5.4.4 Security, Experience, and Familiarity. The pronounced preference and high enjoyment scores for HAPTCHA suggest a potential pathway to mitigate security fatigue (psychological burden often associated with repetitive verification tasks [62]). Unlike passive behavioral modeling operating in the background, HAPTCHA transforms verification into a brief, active sensorimotor task. This engagement strengthens the security signal while providing a salient confirmation of human presence.

Importantly, despite participants' extensive prior familiarity with reCAPTCHA, the HAPTCHA variants—specifically tiltHAPTCHA—demonstrated superior overall usability. This indicates the advantages are not simply novelty effects; rather, tactile-motor coupling is inherently more harmonious with the natural affordances of modern smartphones.

6 Limitations and Future Work

While this study demonstrates the potential of mobile HAPTCHAs, several limitations inform future research directions.

First, our security evaluation demonstrates meaningful human-bot separability under a realistic threat model [2], but does not exhaust all possible zero-day attack vectors. Furthermore, our baseline comparison against reCAPTCHA focused on user-facing interaction flow and perceived burden, rather than its proprietary backend risk-analysis engines [27, 28]. Future work should extend attack modeling and empirically compare HAPTCHA against other sensor- or gesture-based mobile verification schemes [22, 31].

Second, tiltHAPTCHA and rhythmHAPTCHA represent only two specific instances within a wide configurable design space of haptic-enabled CAPTCHAs. Parameters such as frequency, envelope, and feedback mapping can be highly varied, requiring future exploration to map the broader usability-security trade-offs. Additionally, cross-device validation is particularly important because actuator dynamics differ: LRA-based devices generally support more responsive amplitude and timing control, whereas slower ERM spin-up and braking may affect rapid haptic transitions and the resulting sensorimotor traces used for Step 2 verification.

Third, HAPTCHA shifts some accessibility requirements from vision to haptic perception and motor interaction. While this may benefit users who experience difficulty with visual challenges, its accessibility may vary for users with reduced vibrotactile sensitivity or motor impairments. Future work should examine how HAPTCHA can support a broader range of users with different sensory and motor capabilities.

Fourth, the current implementation primarily targets haptic-capable mobile devices. In general web services that retain conventional desktop fallbacks, challenge selection based solely on user-agent information may allow bots to bypass the haptic challenge. More robust deployment may be possible through mobile-app or cross-device verification flows, while the underlying concept can also be explored on haptic-capable peripherals such as haptic mice or VR controllers [48, 51].

Finally, although we tested mobile-relevant conditions (e.g., walking on a treadmill), real-world deployment introduces unconstrained variables like one-handed use, environmental distractions, and longitudinal fatigue [65]. Longitudinal "in-the-wild" studies are necessary to fully capture these ecological factors.

7 Conclusion

In this paper, we presented HAPTCHA, a haptic-centered human-verification framework designed to provide an eyes-free, secure experience in mobile environments. Our research confirms that tactile vibration is a viable primary channel for human-verification (RQ1), as demonstrated by HAPTCHA's high success rate and fast completion time. The unique sensorimotor interaction traces of humans provide sufficient signals to differentiate them from automated agents (RQ2), achieving up to 100% bot rejection. HAPTCHA also maintains practical usability under various mobile conditions (RQ3) by offloading cognitive and visual demands to the haptic channel during locomotion. Empirical results show that the two variants exhibit different usability profiles: tiltHAPTCHA emphasizes efficiency and solvability, whereas rhythmHAPTCHA maintains relatively stable accuracy across the tested mobility conditions, albeit with longer completion times. Ultimately, HAPTCHA demonstrates that embodied sensorimotor interaction can effectively bridge the gap between robust defense against automated agents and a positive, low-workload user experience. We believe this work serves as a promising first step toward establishing eyes-free interaction as a fundamental design requirement for next-generation mobile and wearable security protocols.

Acknowledgments

This work was supported by an ITRC support program (IITP-2026-RS-2024-00437866) and a Soft Haptics research program (IITP-RS-2025-02214780) of IITP, and a Pioneer research program (RS-2024-00451947) of NRF, all funded by the Korean government.

References

[1] Alejandro Acien, Aythami Morales, Julian Fierrez, Ruben Vera-Rodriguez, and Ivan Bartolome. 2020. HuMIdb: A Multimodal Mobile Database for Human Identification and Impersonation Detection in the Wild. arXiv:2002.00918 [cs.CV] doi:10.48550/arXiv.2002.00918

[2] Alejandro Acien, Aythami Morales, Julian Fierrez, Ruben Vera-Rodriguez, and Oscar Delgado-Mohatar. 2021. BeCAPTCHA: Behavioral bot detection using touchscreen and mobile sensors benchmarked on HuMIdb. *Engineering Applications of Artificial Intelligence* 98 (2021), 104058.

[3] Mrim Alnfiai and Fawaz Alassery. 2022. TapCAPTCHA: Non-visual CAPTCHA on Touchscreens for Visually Impaired People. *Journal on Multimodal User Interfaces* 16 (2022), 385–398. doi:10.1007/s12193-022-00394-2

[4] Android Developers. 2026. AccessibilityService | API reference. https://developer.android.com/reference/android/accessibilityservice/AccessibilityService. Accessed: 2026-03-26.

[5] Apple Inc. 2026. Core Motion Framework. https://developer.apple.com/documentation/coremotion Accessed: 2026-03-21.

[6] Ferhat Attal, Samer Mohammed, Mariam Dedabrishvili, Faicel Chamroukhi, Latifa Oukhellou, and Yacine Amirat. 2015. Physical Human Activity Recognition Using Wearable Sensors. *Sensors* 15, 12 (2015), 31314–31338. doi:10.3390/s151229858

[7] Marios Belk, Christos A. Fidas, Panagiotis Germanakos, and George Samaras. 2015. Do Human Cognitive Differences in Information Processing Affect Preference and

Performance of CAPTCHA? *International Journal of Human-Computer Studies* 84 (2015), 1–18. doi:10.1016/j.ijhcs.2015.07.002

[8] Palavi V. Bhole, Ziming Li, Shivang Bokolia, Tae Oh, Garreth W. Tigwell, and Roshan L. Peiris. 2024. Haptic2FA: Haptics-Based Accessible Two-Factor Authentication for Blind and Low Vision People. *Proceedings of the ACM on Human-Computer Interaction* 8, MHCI, Article 264 (2024), 20 pages. doi:10.1145/3676509

[9] Andrea Bianchi, Ian Oakley, Vassilis Kostakos, and Dong Soo Kwon. 2011. The Phone Lock: Audio and Haptic Shoulder-Surfing Resistant PIN Entry Methods for Mobile Devices. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '11)*. ACM, 197–200. doi:10.1145/1935701.1935740

[10] Andrea Bianchi, Ian Oakley, and Dong Soo Kwon. 2010. The Secure Haptic Keypad: A Tactile Password System. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. Association for Computing Machinery, New York, NY, USA, 1089–1092. doi:10.1145/1753326.1753488

[11] Andrea Bianchi, Ian Oakley, Jong Keun Lee, and Dong Soo Kwon. 2010. The Haptic Wheel: Design & Evaluation of a Tactile Password System. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI EA '10)*. Association for Computing Machinery, New York, NY, USA, 3625–3630. doi:10.1145/1753846.1754029

[12] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion Attacks against Machine Learning at Test Time. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 387–402.

[13] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

[14] Kevin Bock, Daven Patel, George Hughey, and Dave Levin. 2017. unCaptcha: A Low-Resource Defeat of reCaptcha's Audio Challenge. In *11th USENIX Workshop on Offensive Technologies (WOOT 17)*. USENIX Association, Vancouver, BC. https://www.usenix.org/conference/woot17/workshop-program/presentation/bock

[15] Hendrio Bragança, Juan G. Colonna, Horácio A. B. F. Oliveira, and Eduardo Souto. 2022. How Validation Methodology Influences Human Activity Recognition Mobile Systems. *Sensors* 22, 6 (2022), 2360. doi:10.3390/s22062360

[16] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32. doi:10.1023/A:1010933404324

[17] Stephen A. Brewster and Lorna M. Brown. 2004. Tactons: Structured Tactile Messages for Non-Visual Information Display. In *Proceedings of the Fifth Australasian User Interface Conference (AUIC '04)*. Australian Computer Society, 15–23.

[18] Seungmoon Choi and Katherine J. Kuchenbecker. 2013. Vibrotactile Display: Perception, Technology, and Applications. *Proc. IEEE* 101, 9 (2013), 2093–2104.

[19] Juan Echeverría, Emiliano De Cristofaro, Nicolas Kourtellis, Ilias Leontiadis, Gianluca Stringhini, and Shi Zhou. 2018. LOBO: Evaluation of Generalization Deficiencies in Twitter Bot Classifiers. In *Proceedings of the 34th Annual Computer Security Applications Conference*. 137–146. doi:10.1145/3274694.3274738

[20] Jeremy Elson, John R. Douceur, Jon Howell, and Jared Saul. 2007. Asirra: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*. Association for Computing Machinery, New York, NY, USA, 366–374. doi:10.1145/1315245.1315291

[21] Valerie Fanelle, Sepideh Karimi, Aditi Shah, Bharath Subramanian, and Sauvik Das. 2020. Blind and Human: Exploring More Usable Audio CAPTCHA Designs. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*. USENIX Association, 111–125. https://www.usenix.org/conference/soups2020/presentation/fanelle

[22] Yunhe Feng, Qing Cao, Hairong Qi, and Scott Ruoti. 2020. SenCAPTCHA: A Mobile-First CAPTCHA Using Orientation Sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2, Article 43 (2020), 26 pages. doi:10.1145/3397312

[23] Arthur M Glenberg. 2010. Embodiment as a unifying perspective for psychology. *Wiley interdisciplinary reviews: Cognitive science* 1, 4 (2010), 586–596. doi:10.1002/wcs.55

[24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Vol. 27.

[25] Google. 2014. Are You a Robot? Introducing "No CAPTCHA reCAPTCHA". https://developers.google.com/search/blog/2014/12/are-you-robot-introducing-no-captcha. Accessed: 2026-03-23.

[26] Google. 2025. Choosing the Type of reCAPTCHA. https://developers.google.com/recaptcha/docs/versions. Accessed: 2026-03-23.

[27] Google Developers. 2018. Introducing reCAPTCHA v3: the new way to stop bots. https://developers.google.com/search/blog/2018/10/introducing-recaptcha-v3-new-way-to. Accessed: 2026-03-23.

[28] Google Developers. 2024. reCAPTCHA v3. https://developers.google.com/recaptcha/docs/v3. Accessed: 2026-03-23.

[29] Google LLC. 2026. Sensors Overview | Android Developers. https://developer.android.com/guide/topics/sensors/sensors_overview Accessed: 2026-03-21.

[30] Rich Gossweiler, Maryam Kamvar, and Shumeet Baluja. 2009. What's Up CAPTCHA? A CAPTCHA Based on Image Orientation. In *Proceedings of the 18th International Conference on World Wide Web*. Association for Computing Machinery, New York, NY, USA, 841–850. doi:10.1145/1526709.1526822

[31] Meriem Guerar, Alessio Merlo, and Mauro Migliardi. 2018. Completely Automated Public Physical Test to Tell Computers and Humans Apart: A Usability Study on Mobile Devices. *Future Generation Computer Systems* 82 (2018), 617–630. doi:10.1016/j.future.2017.03.012

[32] Meriem Guerar, Luca Verderame, Mauro Migliardi, Francesco Palmieri, and Alessio Merlo. 2021. Gotta CAPTCHA 'Em All: A Survey of 20 Years of the Human-or-Computer Dilemma. *Comput. Surveys* 54, 9, Article 192 (2021), 33 pages. doi:10.1145/3477142

[33] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, Amsterdam, Netherlands, 139–183. doi:10.1016/S0166-4115(08)62386-9

[34] Jonathan Ho and Stefano Ermon. 2016. Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., Red Hook, NY, USA, 4565–4573.

[35] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. doi:10.1162/neco.1997.9.8.1735

[36] Thomas Hupperich, Katharina Krombholz, and Thorsten Holz. 2016. Sensor CAPTCHAs: On the Usability of Instrumenting Hardware Sensors to Prove Liveliness. In *Trust and Trustworthy Computing (Lecture Notes in Computer Science, Vol. 9824)*. Springer International Publishing, Cham, Switzerland, 40–59. doi:10.1007/978-3-319-45572-3_3

[37] Petr Janata and Scott T Grafton. 2003. Swinging in the brain: shared neural substrates for behaviors related to sequencing and music. *Nature neuroscience* 6, 7 (2003), 682–687. doi:10.1038/nn1081

[38] Nan Jiang and Feng Tian. 2013. A Novel Gesture-based CAPTCHA Design for Smart Devices. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*. BCS Learning & Development Ltd., Swindon, United Kingdom, Article 49, 5 pages. doi:10.14236/EWIC/HCI2013.57

[39] Manisha Varma Kamarushi, Stacey Watson, Garreth W. Tigwell, and Roshan L. Peiris. 2022. VibroAuth: Authentication with Haptics Based Non-Visual, Rearranged Keypads to Mitigate Shoulder Surfing Attacks. In *HCI for Cybersecurity, Privacy and Trust*. Lecture Notes in Computer Science, Vol. 13320. Springer, 297–316. doi:10.1007/978-3-031-05563-8_19

[40] Manisha Varma Kamarushi, Stacey L. Watson, Garreth W. Tigwell, and Roshan L. Peiris. 2022. OneButtonPIN: A Single Button Authentication Method for Blind or Low Vision Users to Improve Accessibility and Prevent Eavesdropping. *Proceedings of the ACM on Human-Computer Interaction* 6, MHCI, Article 212 (2022), 22 pages. doi:10.1145/3546747

[41] Simon Khan, Charles Devlen, Michael Manno, and Daqing Hou. 2024. Mouse Dynamics Behavioral Biometrics: A Survey. *Comput. Surveys* 56, 6 (2024). doi:10.1145/3640311

[42] Kurt Alfred Kluever and Richard Zanibbi. 2009. Balancing Usability and Security in a Video CAPTCHA. In *Proceedings of the 5th Symposium on Usable Privacy and Security*. Association for Computing Machinery, New York, NY, USA, Article 14, 11 pages. doi:10.1145/1572532.1572551

[43] Ravi Kuber and Sandeep Sharma. 2010. Toward Tactile Authentication for Blind Users. In *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '10)*. Association for Computing Machinery, New York, NY, USA, 295–296. doi:10.1145/1878803.1878875

[44] Ravi Kuber, Wai Yu, Graham McAllister, and Eamonn Murphy. 2010. Feasibility Study of Tactile-Based Authentication. *International Journal of Human-Computer Studies* 68, 12 (2010), 913–923. doi:10.1016/j.ijhcs.2009.11.001

[45] Dajin Lee, Daehyeon Nam, and Seungmoon Choi. 2025. Augmenting outdated museum exhibits with embodied and tangible interactions for prolonged use and learning enhancement. *International Journal of Human-Computer Studies* 198 (2025), 103470. doi:10.1016/j.ijhcs.2025.103470

[46] Sangwon Lee, Jisu Lee, Hyunsu Choi, and Taesoo Kim. 2023. The Vibration Knows Who You Are! A Further Analysis on Vibration-Based User Authentication on Smartwatches. *Computers & Security* 124 (2023), 102977. doi:10.1016/j.cose.2022.102977

[47] Lennart Ljung. 1999. *System Identification: Theory for the User* (2 ed.). Prentice Hall PTR, Upper Saddle River, NJ.

[48] Logitech. 2026. MX Master 4 Wireless Mouse. https://www.logitech.com/en-us/shop/p/mx-master-4. Accessed: 2026-07-13.

[49] Arnold M Lund. 2001. Measuring usability with the use questionnaire. *Usability interface* 8, 2 (2001), 3–6.

[50] Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. 2011. (sp)iPhone: Decoding Vibrations from Nearby Keyboards Using Mobile Phone Accelerometers. In *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS '11)*. Association for Computing Machinery, New York, NY, USA, 551–562. doi:10.1145/2046707.2046771

[51] Meta Platforms, Inc. 2026. Haptics SDK. https://developers.meta.com/horizon/resources/haptics-sdk/. Accessed: 2026-07-13.

[52] A. Keith Palmer, Frank W. Werner, Douglas Murphy, and Ronald Glisson. 1985. Functional Wrist Motion: A Biomechanical Study. *The Journal of Hand Surgery*

10, 1 (1985), 39–46. doi:10.1016/S0363-5023(85)80246-X
[53] Minfeng Qi, Dongyang He, Qin Wang, and Lefeng Zhang. 2026. VIPER Strike: Defeating Visual Reasoning CAPTCHAs via Structured Vision–Language Inference. In *35th USENIX Security Symposium (USENIX Security 26)*. USENIX Association. https://www.usenix.org/conference/usenixsecurity26/presentation/qi-minfeng To appear; prepublication version available.
[54] Bruno H Repp. 2005. Sensorimotor synchronization: A review of the tapping literature. *Psychonomic bulletin & review* 12, 6 (2005), 969–992. doi:10.3758/BF03206433
[55] Gerardo Reynaga, Sonia Chiasson, and Paul C. van Oorschot. 2015. Exploring the Usability of CAPTCHAs on Smartphones: Comparisons and Recommendations. In *NDSS Workshop on Usable Security (USEC)*. Internet Society, Reston, VA, USA, 10 pages. doi:10.14722/usec.2015.23006
[56] J. Y. Ryu, W. P. Cooney, L. J. Askew, K. N. An, and E. Y. S. Chao. 1991. Functional Ranges of Motion of the Wrist Joint. *The Journal of Hand Surgery* 16, 3 (1991), 409–419. doi:10.1016/0363-5023(91)90006-W
[57] Jerome H Saltzer and Michael D Schroeder. 1975. The protection of information in computer systems. *Proc. IEEE* 63, 9 (1975), 1278–1308.
[58] Andrew Searles, Yoshimichi Nakatsuka, Ercan Ozturk, Andrew Paverd, Gene Tsudik, and Ai Enkoji. 2023. An Empirical Study & Evaluation of Modern CAPTCHAs. In *32nd USENIX Security Symposium (USENIX Security 23)*. USENIX Association, Anaheim, CA, USA, 3081–3097. https://www.usenix.org/conference/usenixsecurity23/presentation/searles
[59] Chenghui Shi et al. 2020. Text CAPTCHA Is Dead? A Large Scale Deployment and Empirical Study. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, New York, NY, USA, 1391–1406. doi:10.1145/3372297.3417258
[60] B. W. Silverman. 1986. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall.
[61] Saumya Solanki, Gautam Krishnan, Varshini Sampath, and Jason Polakis. 2017. In (Cyber)Space Bots Can Hear You Speak: Breaking Audio CAPTCHAs Using OTS Speech Recognition. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec '17)*. Association for Computing Machinery, New York, NY, USA, 69–80. doi:10.1145/3128572.3140443
[62] Brian Stanton, Mary F Theofanos, Sandra Spickard Prettyman, and Susanne Furman. 2016. Security fatigue. *It Professional* 18, 5 (2016), 26–32. doi:10.1109/MITP.2016.84
[63] John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive science* 12, 2 (1988), 257–285. doi:10.1016/0364-0213(88)90023-7
[64] Xiwen Teoh, Yun Lin, Siqi Li, Ruofan Liu, Avi Sollomoni, Yaniv Harel, and Jin Song Dong. 2025. Are CAPTCHAs Still Bot-hard? Generalized Visual CAPTCHA Solving with Agentic Vision Language Model. In *34th USENIX Security Symposium (USENIX Security 25)*. USENIX Association, Seattle, WA, USA, 3747–3766. https://www.usenix.org/conference/usenixsecurity25/presentation/teoh
[65] Garreth W. Tigwell, Zhanna Sarsenbayeva, Benjamin M. Gorman, David R. Flatla, Jorge Goncalves, Yeliz Yesilada, and Jacob O. Wobbrock. 2019. Addressing the Challenges of Situationally-Induced Impairments and Disabilities in Mobile Interaction. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems (CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, Article W30, 8 pages. doi:10.1145/3290607.3299029
[66] Erkam Uzun, Simon Pak Ho Chung, Irfan Essa, and Wenke Lee. 2018. rtCaptcha: A Real-Time CAPTCHA Based Liveness Detection System. In *Network and Distributed System Security Symposium (NDSS)*. The Internet Society, San Diego, CA, USA, 15 pages. doi:10.14722/ndss.2018.23260
[67] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. 2003. CAPTCHA: Using Hard AI Problems for Security. In *Advances in Cryptology — EUROCRYPT 2003 (Lecture Notes in Computer Science, Vol. 2656)*. Springer, Berlin, Heidelberg, 294–311. doi:10.1007/3-540-39200-9_18
[68] Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. 2008. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science* 321, 5895 (2008), 1465–1468. doi:10.1126/science.1160379
[69] Ping Wang, Haichang Gao, Xiaoyan Guo, Chenxuan Xiao, Fuqi Qi, and Zheng Yan. 2023. An Experimental Investigation of Text-based CAPTCHA Attacks and Their Robustness. *Comput. Surveys* 55, 9, Article 196 (2023), 38 pages. doi:10.1145/3559754
[70] Yijun Wang, Ziyi Zhou, Weiqi Bai, Ruijie Zhao, and Xianwen Deng. 2024. CaptchaSAM: Segment Anything in Text-based CAPTCHAs. In *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 1070–1076. doi:10.1109/TrustCom63139.2024.00153
[71] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only ANOVA Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 143–146. doi:10.1145/1978942.1978963
[72] Wan Xing, Mohd Rizman Sultan Mohd, Juliana Johari, and Fazlina Ahmat Ruslan. 2023. A Review on Text-based CAPTCHA Breaking Based on Deep Learning Methods. In *2023 International Conference on Computer Engineering and Distance Learning (CEDL)*. 171–175. doi:10.1109/CEDL60560.2023.00040
[73] Xin Xu, Lei Liu, and Bo Li. 2020. A Survey of CAPTCHA Technologies to Distinguish between Human and Computer. *Neurocomputing* 408 (2020), 292–307. doi:10.1016/j.neucom.2019.08.109
[74] Xiangyu Xu, Jiadi Yu, Yingying Chen, Qin Hua, Yanmin Zhu, Yi-Chao Chen, and Minglu Li. 2020. TouchPass: Towards Behavior-Irrelevant On-Touch User Authentication on Smartphones Leveraging Vibrations. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom '20)*. ACM, 24:1–24:13. doi:10.1145/3372224.3380901
[75] Roman V. Yampolskiy and Venu Govindaraju. 2008. Behavioural Biometrics: A Survey and Classification. *International Journal of Biometrics* 1, 1 (2008), 81–113. doi:10.1504/IJBM.2008.018665
[76] Jeff Yan and Ahmad Salah El Ahmad. 2008. Usability of CAPTCHAs or Usability Issues in CAPTCHA Design. In *Proceedings of the 4th Symposium on Usable Privacy and Security (SOUPS '08)*. Association for Computing Machinery, New York, NY, USA, 44–52. doi:10.1145/1408664.1408671

8 Appendices

8.1 Feature Trajectories in tiltHAPTCHA

Figure 9 visualizes the ensemble-averaged (black, thick lines) and individual (colored, thin lines) trajectories of rotation angles and sensor traces when the target angle is 30°.

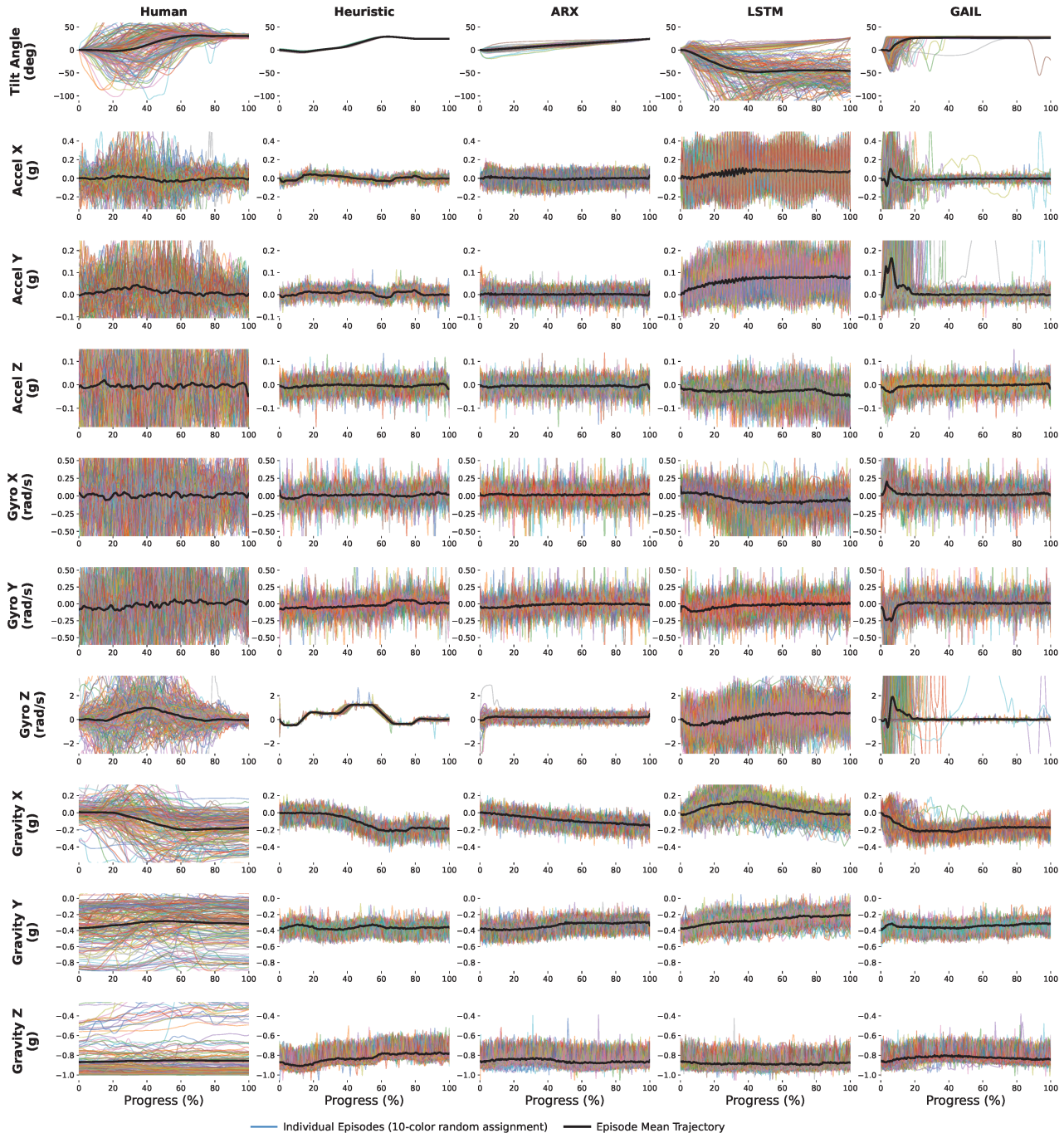


Figure 9: Comparison of average and individual feature trajectories between human users and bots (Target angle: 30°).