CalibRead: Unobtrusive Eye Tracking Calibration Through Natural Reading Behavior

ANONYMOUS AUTHOR(S)

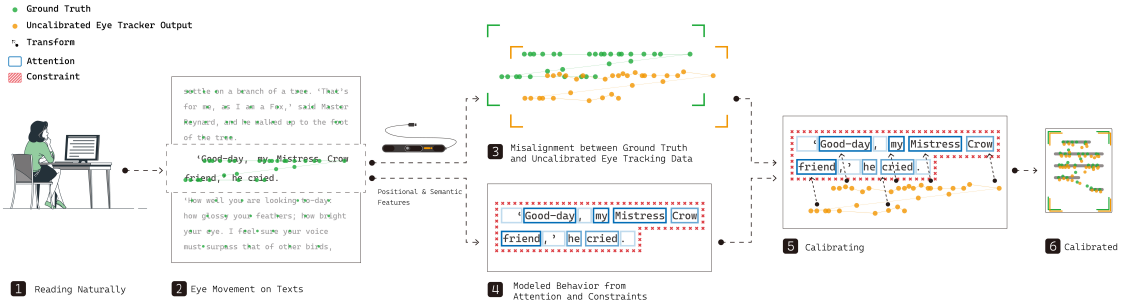


Fig. 1. (1) User reads texts naturally. (2) An unobservable "ground truth" exists for eye movements during reading, known only to the user themselves. (3) The output from the uncalibrated eye-tracking device exhibits misalignment with this ground truth. (4) Our model predicts reading behavior by identifying areas of the text where the user is likely to focus or ignore. (5) We apply a transformation to the uncalibrated data to minimize the discrepancy between it and the behavior predicted by our model. (6) The optimal transformation yields the calibrated eye movement data.

In this paper, we present a novel, unobtrusive calibration method that leverages the association between eye-movement and text to calibrate eye-tracking devices during natural reading. The calibration process involves an iterative sequence of 3 steps: (1) matching the points of eye-tracking data with the text grids and boundary grids, (2) computing the weight for each point pair, and (3) optimizing the calibration parameters that best align point pairs through gradient descent. During this process, we assume that, from a holistic perspective, the gaze will cover the text area, effectively filling it after sufficient reading. Meanwhile, on a granular level, the gaze duration is influenced by the semantic and positional features of the text. Therefore, factors such as the presence of empty space, the positional features of tokens, and the depth of constituency parsing play important roles in calibration. Our method achieves accuracy error comparable to traditional 7-point mehtod after naturally reading 3 texts, which takes about 51.75 seconds. Moreover, we analyse the impact of different holistic and granular features on the calibration results.

CCS Concepts: • **Human-centered computing → Interaction techniques**.

Additional Key Words and Phrases: Eye Tracker Calibration, Reading Model, Unobtrusive Calibration, Implicit Calibration

ACM Reference Format:

Anonymous Author(s). 2024. CalibRead: Unobtrusive Eye Tracking Calibration Through Natural Reading Behavior. 1, 1 (July 2024), 29 pages. https://doi.org/XXXXXXX.XXXXXXX

Manuscript submitted to ACM

1 Introduction

Eye-tracking devices depend on traditional explicit calibration, a process that minimizes the accuracy error between output of gaze coordinates and the ground truth of where the user looked. However this process could notably degrade the user experience [35, 49, 56]. In this approach, users are required to focus on a series of pre-set points on the screen. The necessity for frequent recalibration amplifies the inconvenience. Although there are existing unobtrusive calibration techniques, such as those that use saliency maps generated from various sources, these methods can suffer from a subjective definition of saliency and imprecise saliency region generation [53], compromising calibration accuracy.

We introduce CalibRead, a calibration method that is both unobtrusive and accurate. Picture this scenario: a user connects a new eye tracker to the computer. Without any explicit calibration, the initial accuracy error is unacceptable. However, the user has a daily habit of reading news. Within just a few minutes of reading, our method enables the eye tracker to reach accuracy levels comparable to those traditional explicit calibration methods.

Our approach leverages the association between eye movement and text to calibrate eye tracking device during routine reading activities. The effectiveness of reading as a calibration clue is rooted in the basic principle of eye-tracking calibration: establishing a reliable association between the user's input signal, such as the angular position of the pupil center relative to corneal reflection in PCCR (Pupil Center Corneal Reflection) [9, 11, 18], and the true gaze point on the display screen. Unlike explicit calibration, which imposes stringent constraints on user behavior, reading provides a more natural and less intrusive form of guidance. The textual content itself, being small and surrounded by white empty space, naturally constrains our eye movement, enabling higher precision.

The target of our approach is to identify the affine transformation matrix which, when applied to the raw gaze coordinates from eye-tracking device, minimizes the accuracy error between the transformed gaze coordinates and the ground truth of where the user looked. To implement our approach, two key research questions (RQs) were identified:

RQ1: What are the characteristics of eye movements during reading, and which can be used for calibration?

RQ2: How to build an effective calibration algorithm based on these characteristics, and how well does it perform?

To address RQ1, we carried out a user study to collect eye movement data during natural reading. We also interviewed participants to share insights on their reading behaviors especially those noticeable patterns we observed, such as why did they focus on or neglect certain words. Our observation revealed that: (1) Empty space acts as boundary, which significantly shaped the distribution of gaze. (2) The distribution of gaze is spatially uneven. Readers tend to pay more attention on the top-left corner and less attention on bottom-right corner. (3) Readers tend to pay much less attention on punctuations. (4) Gaze duration on each text grid (Chinese character) within a sentence was influenced by positional, semantic and syntactic features. Details are in Section 3.

For RQ2, we adapted ICP (iterative closet point) algorithm [5] using the findings of user study. Similar to the original ICP algorithm, we first match gaze points with the center points of text grid or boundary grid based on their proximity in space, resulting in multiple pairs of points. Next, we will compute a weight for each point pair according to the features of the Chinese character in the text grid or the location of the boundary grid. Subsequently, we employ gradient descent to determine the optimal affine transformation matrix. This matrix, when applied to gaze points and repositions them, minimizes the weighted distance between corresponding point pairs. Details are in Section 4.

We evaluate our method using an eye tracker with an inherent accuracy error of 0.20°(15.62 px). Each text in our experiment consists of less than 180 Chinese characters, and the average reading time for each text is 17.25 seconds. Before calibreation, the accuracy error is 1.21°(93.76 px). Results show that after reading 3 texts (about 51.75 seconds),

our method's accuracy error does not significant differ from the traditional 7-point method; after reading 11 (about 189.75 seconds) or more texts, our method exhibits significantly lower accuracy errors. Upon reading 22 texts (about 379.5 seconds), our method achieves a minimum accuracy error of 0.29°(22.32 px), outperforming the 7-point accuracy error of 0.39°(29.75 px) by 24.9 %. Details are in Section 5.

Further evaluations reveal that boundary grids significantly enhance calibration, as users tend not to look at the beginning and end of each row, thus providing effective constraints. Additionally, granular features related to tokens, such as the length of the token a Chinese character belongs to and the character's position within the token, are much more influential to the calibration result than others.

Our contributions are as follows:

(1) We proposed an implicit calibration method using natural reading.
(2) Our method achieves lower accuracy error while being unobtrusive to users.
(3) We evaluated different holistic and granular features to determine their impact on calibration result.

The rest chapters include related works (Section 2), limitations and future work (Section 6), and conclusion (Section 7).

2 Related Works

2.1 Explicit Eye-Gaze Calibration

Calibration is a necessary step in eye gaze tracking. Based on the anatomy eye model, the 9-point calibration [10, 34] is well-known and widely used in eye trackers. The model can be constructed using cameras (RGB [25, 55, 59], infrared [27, 31] or depth [17]), eye trackers (Tobii [1], etc.) and head-mounted devices (VR headset, etc). Users are required to fix the gaze at several targets (usually organized 3 by 3 or 4 by 4) to finish the calibration. However, this calibration method is only based on fixed-position targets, resulting in a tedious calibration process and optimizable accuracy.

Beyond the traditional stationary calibration targets, pursuit methods offer an alternative for achieving more efficient and accurate eye tracking by requiring users to follow a moving object. Drewes et al. [14] proposed a pursuit-based method with circular trajectories, achieving the best result with an accuracy of 19 px (0.38°). Similar designs like different target trajectories [43], different targets [37, 44], and combinations of the two [30] are also developed. Game-like calibrations [16, 47] have also been explored to reduce boredom.

Although these methods improve the interest and accuracy of calibration, they still require users to complete the calibration during the interaction process actively. Explicit method causes inconsistency and inadequate scalability due to differences in equipment[60]. It also interferes with the user's interaction process, causing discontinuity and inconvenience. Therefore, we hope to integrate the calibration process into the user's interaction process in an implicit way, maintaining the coherence of the interaction while ensuring the accuracy and effectiveness of the calibration.

2.2 Implicit Calibration

Implicit calibration methods are steps that adjust and optimize gaze estimators during the usage of the devices. Previous work has explored implicit calibration during interactions with saliency maps, which shows that the method and accuracy are affected by the content the user sees. Sugano et al. [53] provides a calibration algorithm using visual saliency in video clips. The algorithm extracts six features to build saliency maps and feeds them into weight optimization, thus obtaining a transformation from the camera images to the gaze points. Besides video clips, there are other

sources to generate saliency maps, including mouse-click behaviors in desktop scenario [54], everyday user interactions (clicking, dragging, typing, etc.) [23], mobile phone usages [36] and head-mounted devices [52]. A probabilistic and incremental algorithm [6] is also set to fit visual saliency.

Implicit calibration relies on the user's eye movement patterns and behavior when paying attention to content. Text is also an important component in user interactions and has been explored in explicit calibration [30]. Khamis et al. [30] introduced a calibration method similar to pursuit [14] by partially displaying text, prompting readers to follow and read the "partially displayed" content. However, there is a lack of research on leveraging natural reading behavior for eye-tracking calibration, and the special patterns and features of reading behavior remained unexplored in implicit calibration. We hope to fill this gap by understanding reading behavior and achieving implicit calibration of eye movements.

2.3 Eye Movement Behavior While Reading

Reading has been wildly explored in the past decades in the field of cognitive psychology. An overview of eye movements was discussed, including fixations, saccades, visual acuity, etc.[45]. Based on the cognitive identifications of eye movements, many models were proposed to explain why fixations occur for certain words and why they take a certain time. A dynamic model, SWIFT(Saccade-generation with inhibition by foveal targets) [15], was proposed to predict eye movement events. E-Z Reader [46], accounted for the eye movement control comprehensively. Reading behavior in different languages has also been explored. Apart from letter-based reading, there are several findings working on Chinese characters. Li et al. [32] established an integrated computational model for Chinese reading, providing methods for Chinese word processing, eye movement control, and visual processing.

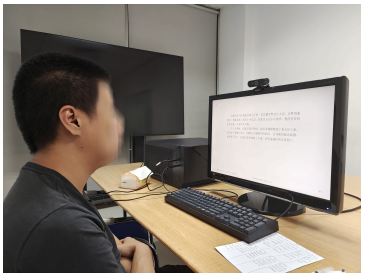
Based on the rich features during reading, previous work has achieved prediction of fixations and saccades[12, 19, 20, 38, 58]. The language model also provides a reference for eye movement prediction. BERT [13] is utilized for word embeddings. Bensemann et al. [4] found a strong correlation between human eye movements and early layers of pre-trained transformers. The relationship between eye movements and reading behavior provides assurance for calibration. However, which features can be used for implicit calibration have not been explored. In this paper, we focus on extracting key reading behavior features and building a reading-based implicit calibration method.

3 User Study 1: Understanding Reading Behavior

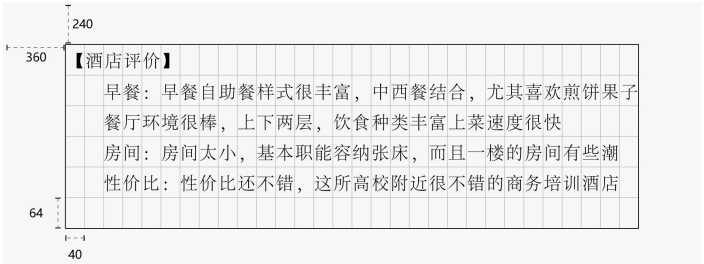
The purpose of this user study is two-fold. First, we seek to identify specific features of user's reading behavior through eye-tracking data and interviews. Second, we aim to develop an algorithm capable of leveraging features for effective calibration. We hope the insights gained from this could also offer valuable perspectives for future research.

3.1 Apparatus

The study was conducted on a 24-inch 1920 px × 1200 px 52.0 cm × 32.7 cm monitor where participants read multiple texts that we prepared. Texts are displayed in a 6×30 grid layout (check Figure 2b), allowing for Chinese, punctuation, Arabic numbers, and alphabets. Each grid is 40 px in width and 64 px in height. A keyboard is prepared for participants to conduct operations during reading. Eye movements were recorded using a Tobii Eye Tracker 5, which is mounted at the bottom of the computer screen. Initial calibration was conducted using the eye tracker's built-in calibration. We captured the entire experiment on video using a camera and simultaneously recorded the monitor's display.



(a) User Setup



(b) Grid Layout in Screen

Fig. 2. Experiment Setup

3.2 Procedure and Participants

We recruited 24 participants (16 males and 8 females, aged from 19 to 33, $Mean = 23.81, SD = 4.13$). We offered a compensation of $15 USD per hour.

Participants were asked to sit 60 cm in front of the computer and maintain a comfortable posture (Figure 2a). Before the experiment begins, we will calibrate the eye tracker using its built-in calibration program. Experiments were conducted in multiple rounds, each consisting of three phases: reading multiple texts, interview, and manual calibration.

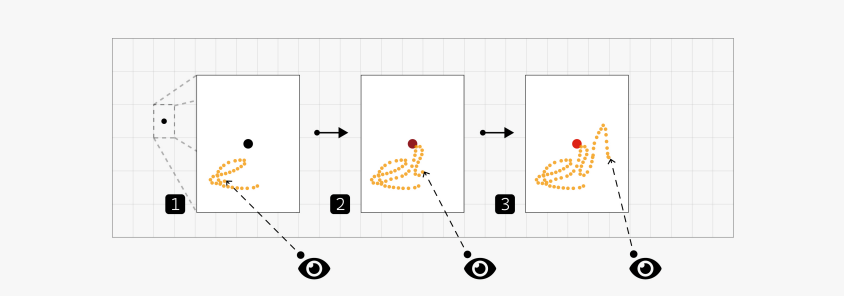
Reading Phase. Participants read the text displayed on the screen naturally and navigated to the next text using the 'page down' key.

Interview Phase. After each participant finished reading a text, we inquired about the elements that left an impression on him/her. Subsequently, we visualized the gaze points of last text, and ask them to explain their own eye-movements. For example, we posed questions such as "why do you spend extra time on this Chinese character? does it reminds you of anything", "you read that part slowly, could you recall what you were thinking about at that time?". During the pilot study, we noticed that some participants read each Chinese character one after another, trying to remember every detail in anticipation of the interview. Therefore, we advise participants against adopting this approach during the current session.

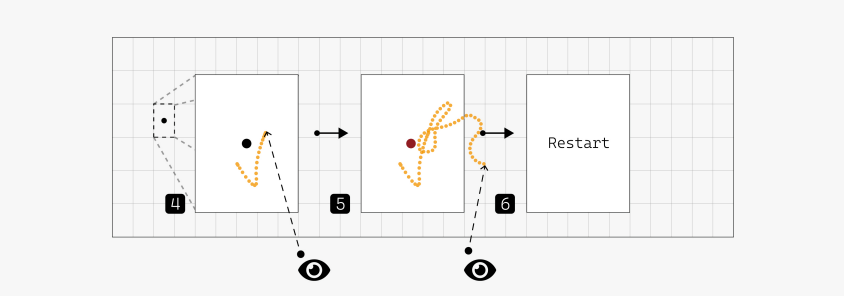
Manual Calibration Phase. This process emulated the explicit calibration method, obtaining the center points of 180 grids on the screen along with the eye movement data when users fixated on these centers. At the end of each round, participants were instructed to fixate on a black dot at the center of several randomly selected grids. As they fixate, the dot's color will gradually turn to red (Figure 3a). Concurrently, we monitored the eye movement trajectory, setting a 40×64 rectangle as a boundary. If eye movements exceeded this boundary, calibration for that specific point would have to be restarted (Figure 3b). We collected both the fixation gaze coordinates output from eye tracker (marked as C in Section 4) and the coordinates of the corresponding grid center (marked as T in Section 4). After all rounds were completed, each grid in the layout would have been fixated at least once, which gives $6 \times 30 = 180$ coordinate pairs (point pairs) in total. These data is further used to correct the inherent error of the eye tracker.

3.3 Eye Movement Data Process

Due to the inherent measurement error in the eye tracker, some eye movement data from **Reading Phase** is misaligned with the text, resulting in an overall offset. To address this issue, we use data from the **Manual Calibration Phase** to calibrate the eye movement data from the **Reading Phase**. Specifically, we derive an affine matrix (marked as A in



(a) Calibration success: (1) The participant begins by gazing on the grid center. (2) The dot at the center gradually turns red. (3) If the participant's gaze remains within the boundary for 1 second, manual calibration is completed.



(b) Calibration fail: (4) The participant begins by gazing at the grid center. (5) However, the gaze exceeds the boundary within 1 second. (6) Manual calibration will be stopped and restarted.

Fig. 3. Manual Calibration Process

Section 4), from the 180 point pairs collected during the **Manual Calibration Phase** using the least squares method. When applied to the fixation gaze coordinates in the **Manual Calibration Phase**, this matrix minimizes the distance between the transferred fixation gaze coordinates and their corresponding grid center coordinates. Applying the same matrix to the eye movement data from the **Reading Phase** produces calibrated data, and all subsequent experimental analyses were based on these calibrated results.

3.4 Result

The results can be classified into two themes: a *broad, holistic perspective* that captures patterns between extensive eye movement data and the position of text, and a more *detailed, granular level view* focusing on specific rules within in a sentence.

3.4.1 Holistic Perspective

Generally speaking, the gaze point would encompass the text area, effectively filling it after sufficient area, as shown in (Figure 4).

Top-Left Corner and Bottom-Right Corner. Specifically, participants tend to pay more attention on the left-top area, and less attention on the right-down area. This is because the reading direction of Chinese is from left to right, top to bottom. Therefore the top-left corner often contains the titles or key information. Our interviews proved that

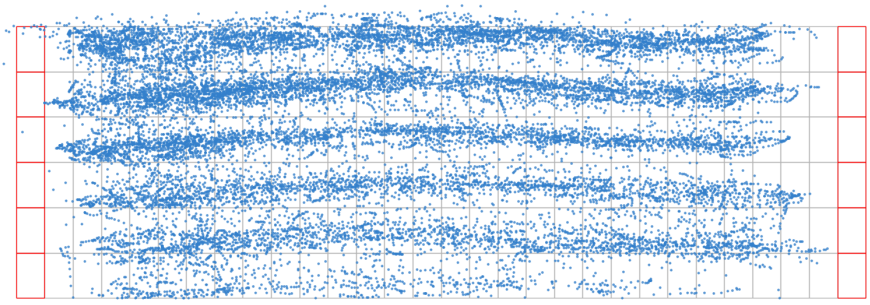


Fig. 4. Overlay of all eye movement traces shows: (1) more gaze on left-top corner and less on bottom-right corner, (2) first and last characters, marked in red, are often ignored, (3) rows are distinguishable.

participants would pay attention to the first line, regardless of whether they engage deeply with the entire content. Subject W mentioned:

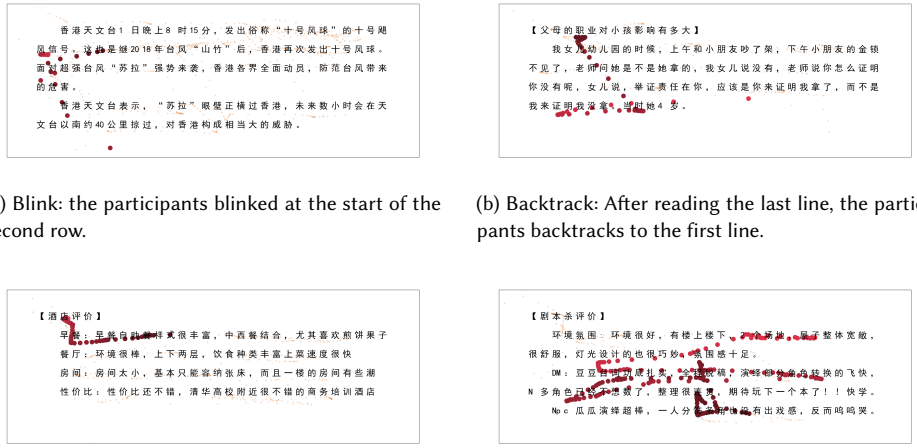
> *"Within the first half-sentence of the first line, I generally understand the main topic of the article. By the end of the first line, I knew whether the article interested me. If not, I might give a brief scan a bit more and skip ahead."*

On contrary, the bottom-right corner tends to be unattended. This could be attributed to 2 reasons. First, participants often do not finish the whole text. They have already grasped the meaning of the text before reaching the end, stopping somewhere in the middle. Second, the concluding position varies between articles. For those articles end in the middle, participants will never look the right-down corner.

Vacant on Start and End of Each Row. We found that participants' eye movements do not start from the first character and also do not stop at the last character. There is usually a gap on both ends (Figure 4). After summarizing the interview findings, we believe there are two possible reasons for this: (1) Participants may skip content at the beginning or end of a line because it can be easily inferred from the preceding content (e.g., a word that is split); (2) When reading from left to right, the peripheral vision captures content at the end of the line; similarly when shifting from right to left to read the next line, the peripheral vision captures content at the beginning of the next line.

Rows and Columns. In (Figure 4), a noticeable pattern emerges as we distinctly observe differentiation between rows in the point cloud data of gaze. It is evident that the gaze is predominantly centered within each row, with limited spillover into adjacent rows during reading. However, the boundaries between rows occasionally blur. This phenomenon can be attributed to 4 possible factors: first, blinks result in abrupt drops across rows (Figure 5a); second, participants may backtrack to revisit previous content (Figure 5b); third, the gaze trace exhibits continuity between rows when participants seamlessly transition from one short row to the next (Figure 5c); forth, when participants are skimming the content, their speed tends to slow down as they switch from the end of one line to the beginning of the next line, resulting in more gaze trajectories across rows (Figure 5d).

Conversely, distinguishing columns proves to be challenging, even when we use a standard unit to encapsulate each Chinese character. This difficulty arises due to varying reading strategies employed for different texts.



(a) Blink: the participants blinked at the start of the second row.

(b) Backtrack: After reading the last line, the participants backtracks to the first line.

(c) Continuous Switch: After reading the short first row, participants' gaze move the the second row continuously.

(d) Skim: the participants skim through the third row and fifth row.

Fig. 5. Four Cases of Gaze Crossing the Row. Points in the image indicate the participants' gaze. The darker the color, the later the point appears in the sequence.

3.4.2 Granular Level

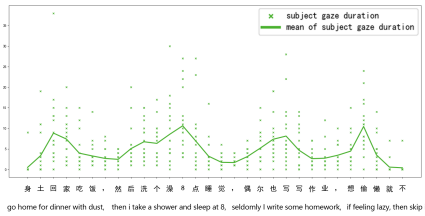
As observed in Figure 6a and 6b, in most cases, within a row, there is a noticeable decrease in gaze duration at punctuation marks or spaces. Moreover, within each set of sentences separated by punctuation, for sentences with around 7 Chinese characters, gaze duration tends to be longer towards the center and shorter towards the edges.

However, for sentences with a larger number of Chinese characters (Figure 6d), it becomes more challenging to discern patterns in gaze duration. We believe this is because in shorter sentences, participants' attention strategies are more closely associated with structural features of the sentence (such as its position, length, and the position of certain Chinese character within it). In longer sentences, participants' attention strategies may be more strongly linked to the actual semantic content of the sentence. This could result in significant variations due to various participant-specific factors such as prior knowledge, familiarity, etc. We will provide a brief elaboration on this issue based on our interview results in Section 3.5.

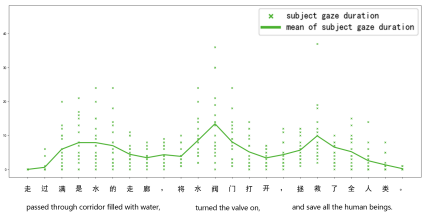
In a few cases (Figure 6c), gaze duration remains relatively stable. Three possible explanations account for this phenomenon:

(1) **Close Relationship between Adjacent Contents**. The adjacent content is of close semantic connection or a parallel relationship. Therefore participants will pay extra attention on the following element after finishing the preceding ones.

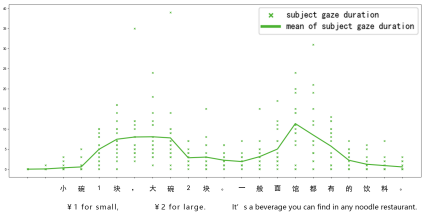
(2) **Complex Content**. When the text is difficult to understand, participants tend to slow down their reading speed. This often results in longer gaze duration on punctuation marks. For example, when we asked Subject J about her linger on punctuation marks, she stated, *"I didn't intentionally stop there; I was probably just processing the preceding content."*



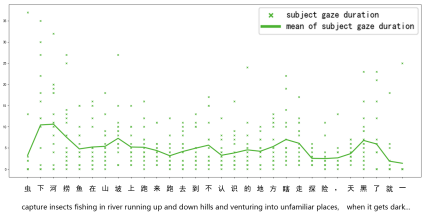
(a) Examples of a noticeable decrease in gaze duration due to punctuation. Peaks of gaze duration are frequently observed for shorter sentences unit separated by punctuation.



(b) Another examples for decrease due to punctuation.



(c) Examples for long duration on punctuation. Due to the parallel nature of the descriptions before and after the sentence, participants treated them as a cohesive unit here; as a result, the gaze duration at punctuation did not decrease.



(d) Examples for long sentence without separator. Gaze duration does not exhibit distinct features in such situations.

Fig. 6. Four examples of gaze duration in different texts. The green "x" marker in the figure represent gaze duration data from different participants, and the green lines connect the average values.The bottom of each figure displays the Chinese characters of this sentence and its English translation.

(3) **Peripheral Attention**. Sometimes, participants focus on punctuation marks because their peripheral vision has already caught the text that follows. In other words, the participant starts to process the text following the punctuation while their gaze remains on the punctuation itself. Like subject H mentioned in the interview, *"I didn't intentionally look at the punctuation, but I remember paying special attention to the word that followed."*

It's worth noting that even when participants report having a strong impression of certain key information, the time they actually spend looking at it can be surprisingly brief. During the interviews, participants often expressed surprise at this, saying, *"I definitely remember that word, but why was my gaze so brief?"* One possible explanation for this phenomenon could be the semantic "priming" [2] provided by the text leading up to that particular keyword. The text might have laid sufficient groundwork to guide the participant's thoughts, enabling them to anticipate what's coming next. As a result, when they actually encounter the keyword, it doesn't require much additional cognitive effort or attention to process it.

3.5 Explanation about the Gaze Duration Variation

While analyzing the experimental data, we noticed that, despite the features influencing gaze duration mentioned earlier, there is often significant variance in gaze duration among different participants for the same sentence. We conducted further interviews to better understand the attention allocation strategies employed by participants during

the reading process and summarized 4 factors: personal preferences, prior knowledge, reading abilities,and linguistic habits. We anticipate that this aspect of our study may offer valuable insights for future research.

(1) **Personal preference.** The degree to which participants engage with a text strongly correlates with how well the content aligns with their personal preferences. When the text does not appeal to them, participants tend to skim quickly through the content. Conversely, they read carefully line by line.
Beyond behavioral patterns (focused reading or skimming), preferences also dictate the focus of attention. For instance, when reading the same restaurant review, Subject L stated, *"I'm more concerned about the ambiance and the service,"* while Subject J noted, *"I only care about how the food tastes and whether the ingredients are fresh."* Similarly, when reading a sports news article, Subject H said, *"I like this athlete and focused solely on parts related to him, probably overlooking sections about his competitors,"* whereas Subject J indicated, *"I paid attention to information related to both sides"*

(2) **Prior knowledge.** It refers to a participant's familiarity with the information described in the text. This aspect is particularly salient when reading news articles. When presented with a news story, users who have not previously encountered related news will focus on details such as the time, location, individuals involved, and specific events. In contrast, those who are already familiar with the news subject will look for new insights they haven't encountered before, such as analyses of the ongoing impact of the event.
The phenomenon extends to product reviews as well. For instance, when we presented a makeup product review to a subject highly knowledgeable about cosmetics, she remarked, *"I can almost guess what the product is just by seeing the first few letters of its long, specific name. Given the context, only certain types of products from that brand are usually mentioned."*

(3) **Reading abilities and habits.** It refers to how well a participant can adapt to the "difficulty" level of the text. Generally, participants will spend more time focusing on more complex parts of the text and will quickly scan through simpler, more predictable parts. Observation leads us to believe that the ability to read complex content is also closely related to a participant's reading habits.
For instance, in our study, Subject J spent a significantly longer time reading almost all types of texts, essentially reading them word-for-word. The only exception was user reviews. During the interview, she mentioned that she primarily reads user reviews in her daily life and, therefore, knows the typical information conveyed in such texts. This leads to an important insight: frequent exposure to a particular type of text allows the user to build specific expectations about the information conveyed and how it's presented, making even complex, high-difficulty texts easier to understand.

(4) **Linguistic preferences.** One Subject, M, who is from a linguistics background, noted during the interview that she chooses what to read based on the composition and meaning of the words. *"I pay more attention to verbs that have a tangible meaning, such as 'escape', 'fly', or 'die',"* she said. *"On the other hand, I tend to overlook verbs whose meaning is largely dependent on subsequent objects, like 'hold', as in 'hold a ceremony' or 'hold a cup'. The verb itself doesn't convey much; the critical information lies in what follows."*
This insight suggests that the syntactic and semantic structure of the text can have an impact on reading behavior. Words or grammatical structures that efficiently convey information may naturally attract more attention.

3.6 Strategies for Features

For the holistic features outlined in Section 3.4.1, we add extra boundary grids surrounding the text area to constrain gaze points during calibration. Details can be found in Section 4.2. Concerning the granual level features mentioned in Section 3.4.2, we developed a simple neural network to predict the gaze duration for each Chinese character. This neural network takes into account various features, including the character's position, the length of the Chinese word it belongs to, the length of the sentence it is part of, the semantic embedding from GPT of the Chinese word it belongs to, and the semantic embedding from GPT of the sentence. The resulting vector serves as input for the prediction of gaze duration. The predicted gaze duration, after processing, is utilized in point matching and gradient descent. Further details can be found in Section 4.5.

4 Alignment Between Eye Movement and Text

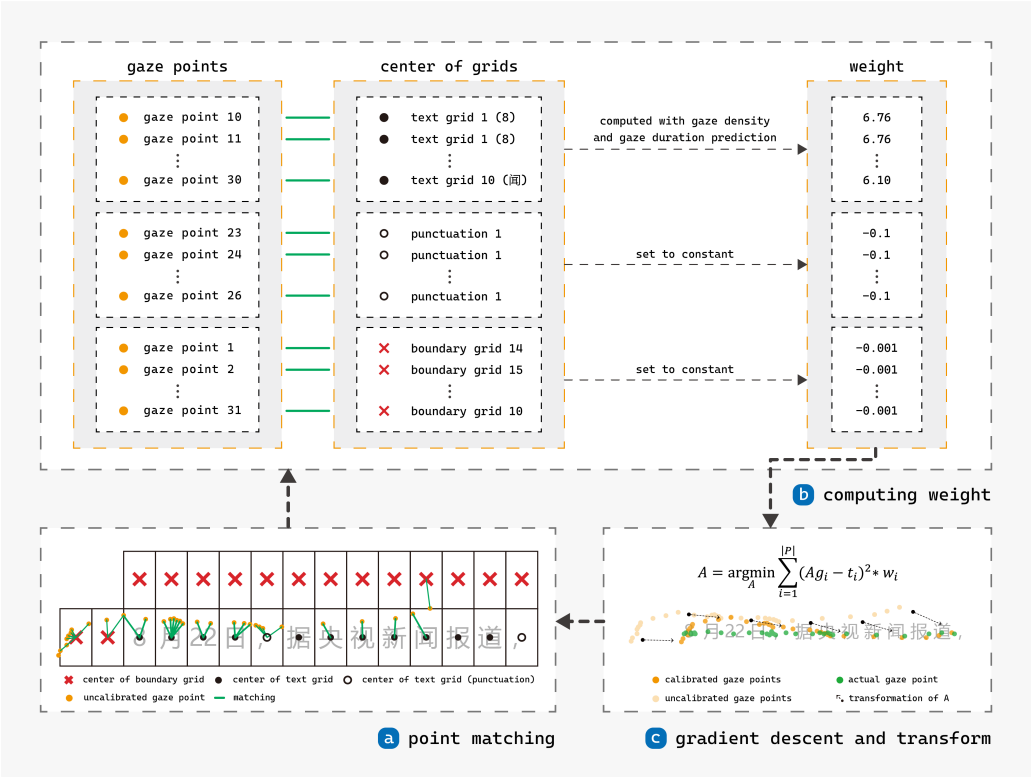


Fig. 7. Calibration Workflow. (a) Point Matching: We match each gaze point with its closet text grid center and boundary grid center. (b) Computing Weight: For each point pair, we calculate its weight. For text grids, the weight is computed using *gaze density* and *gaze duration prediction*. For text grids of punctuation and boundary grids, we set their weight to contants. (c) Gradient Descent: We compute the optimal affine matrix by minimizing the weighted distance between point pairs. After transforming the gaze points with affine matrix, we return to point matching to initiate the next iteration.

Based on the reading behavior analysis presented in the previous chapter, we introduce CalibRead. The model takes both the uncalibrated eye movement data from the eye tracker and the text, as well as the locations of the center of

each Chinese character as input. It then outputs an affine matrix A that transforms the uncalibrated eye movement data to the actual position.

Before calibration, we will denoise the eye movement data first to eliminate anomalies like blinks and some experimental errors, such as moments when the user looks away from the screen during experiments. Check Section 4.1 for more details.

Our method is a modified version of ICP, comprising iterative steps of (1) point matching (Section 4.4), (2) computing weight (Section 4.5), (3) gradient descent (Section 4.6). See Figure 7 for details. During point matching, we pair each gaze coordinate in the eye movement data with its closest text and boundary grid. After that we compute weight for each point pair based on its unit type. For text grids, the weight is computed using *gaze density* and *gaze duration prediction*. For text grid of punctuation and boundary grids, we set their weight to contants. Finally, we minimize the total weighted distance between each pair of points to obtain the optimal affine matrix. After transforming the gaze points with affine matrix, we return to point matching to initiate the next iteration.

4.1 Denoise Gaze Data

Event detection has been researched for a long since it's important for explaining and utilizing eye gaze data. Traditional algorithms including I-VT [3] for saccades, I-DT [48] for fixations, adaptive algorithms including [39] and I2MC [22] and machine learning methods [51, 61] have been well developed. Generally, data is classified into fixations, saccades, return sweeps, regressions, and noise. In this paper, we develop an algorithm inspired by [39], using both hand-tuned and adaptive parameters to recognize blinks and other noisy data as invalid data, leaving the rest as valid data fed into the tuning step. The algorithm can be described as follows:

(1) Velocity Computation: To calculate the gaze point velocity, we employ a second-order Savitzky-Golay filter as detailed in [39]. This filter is applied to the gaze point data G to obtain the first-order differential velocity.
(2) Noise Labeling: Velocity peaks exceeding a predefined noise threshold are identified as noise events. Subsequently, for each noise peak, we detect the onset and offset times based on a preset fixation threshold. To mitigate potential issues arising from isolated fixations occurring between two noise sequences, a merging operation is applied.
(3) Blink Detection: Blinks are characterized by a distinctive pattern of rapid downward and subsequent upward motion, typically occurring within a short time. To identify blinks, we devised two convolution kernels. One kernel is designed to detect blink peaks, while the other ensures that the vertical (y-coordinate) position remains relatively constant before and after the blink. Any peak in the convolution results is marked as a blink event. Subsequently, we determine the onset and offset times using a threshold set to the 95th percentile of the y-values of regular points and the detected blink peaks.

4.2 Input and Output of Calibration

CalibRead employs the data from experiments in Section3 and Section 5. In this section, we will first introduce the output of CalibRead. Then, we will describe the detailed composition of the data we get from the **Reading Phase** and **Manual Calibration Phase** (Section 3.2), their usage in calibration and validation, and the derived data calculated from them.

Affine Matrix A

Affine matrix A is a 3×3 matrix for affine transformation (1). Multiplying the uncalibrated gaze coordinates by A yields the calibrated coordinates. We take translation, rotation, scaling, and shear into consideration. Therefore we need to optimize 7 parameters ($trans_x$, $trans_y$, θ, $scale_x$, $scale_y$, $shear_x$, $shear_y$) to obtain the final affine matrix.

$$A = \begin{bmatrix} 1 & 0 & trans_x \\ 0 & 1 & trans_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} scale_x & 1 & 0 \\ 1 & scale_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & shear_x & 0 \\ shear_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Eye Movement Data during Reading G

We collected the uncalibrated eye movement data (a sequence of coordinates) of a user during reading a specific text in **Reading Phase**, which consists of the coordinates g of all gaze points. G_i represents the eye movement data for text i. The orange points in Figure 7(a) and transparent orange points in Figure 7(c) are examples of g. After applying the Affine Matrix A to G, we obtain the calibrated eye movement data G', consisting of coordinates g'. The opaque orange points in Figure 7(c) are examples of g'. G is the primary data for calibration.

Centroids when Focusing on Certain Text Grid C

We capture the uncalibrated eye movement data (a sequence of coordinates) when the user fixates on the center of each grid in **Manual Calibration Phase** and then calculate the centroid coordinate c of the sequence. According to Section 3.2, C comprises 180 centroids c. We define c' to represent the same centroid after being transformed by matrix A, and C' as the set for c'. These centroids will be used to evaluate the accuracy error of certain calibration method (details in Section 5).

Coordinates of Text Grid T

As mentioned in Section 3.1, all texts would be displayed in a grid. The coordinates of text grid t refer to the coordinates of its center. There are in total 180 different text grid coordinates t. As shown in Figure 7 (a), the centers of text grids are marked as black dots.

Coordinates of Boundary Grid T

As mentioned in Section 3.4.1, we need to add an outer boundary around the text area. Specifically, as illustrated in Figure 7 (a), for those text grids at the edges of the text area, we will add an additional layer of boundary grids close to the outer side of their edges (marked as red cross). The width and height of these boundary grids are exactly same as text grids. The center of these boundary grids is considered the coordinate of the boundary grid t. For the convenience of subsequent formula representation, we use the same notation t for the coordinate of the boundary grid and the coordinate of the text grid.

Gaze Density GD

As shown in Figure 8, we compute a density value gd for each gaze coordinate g_{target} in G. It is defined as the count of temporally contiguous gaze points g to g_{target}. If a gaze point g is temporally contiguous, all gaze points between it and the g_{target} in time sequence have distances to g_{target} that are less than a specified threshold. In other words, if a user looks at one area, then shifts gaze to another area before returning to the original area, the initial gaze point and the subsequent one upon returning to that area are not considered temporally contiguous.

A higher gaze density indicates the user focuses longer around the gaze point.

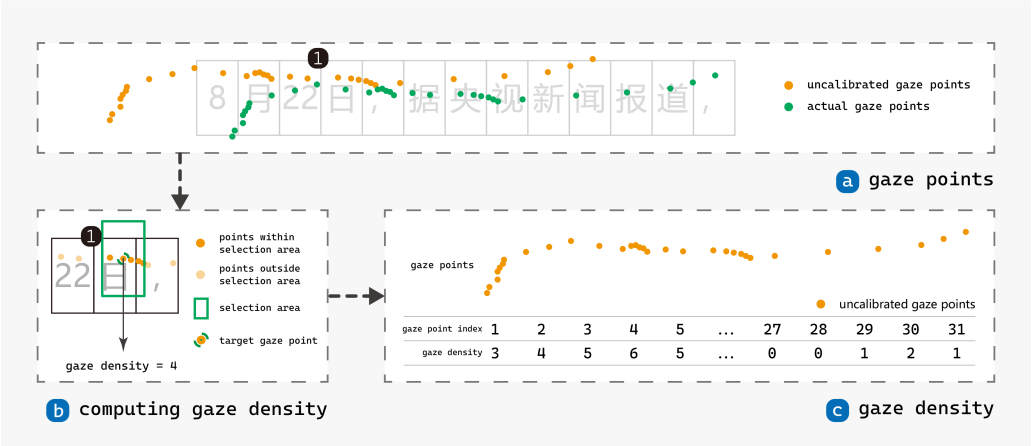


Fig. 8. Computing Gaze Density. (a) Although the uncalibrated gaze points (from eye tracker) may differ from the actual gaze points (ground truth), their gaze densities are identical. (b) For a certain gaze point, we count the number of adjacent temporally contiguous gaze points within the selection area as the gaze density. (c) Using this method, we obtain the gaze density for each gaze point.

4.3 Centroid Alignment

Before the iteration of point matching, computing weight and gradient descent optimization, we will initially align the centroid of all gaze points G and with that of all text grid T. Subsequently, based on the bounding rectangles of the two point sets, we will perform scaling transformations in the x and y directions.

4.4 Point Matching

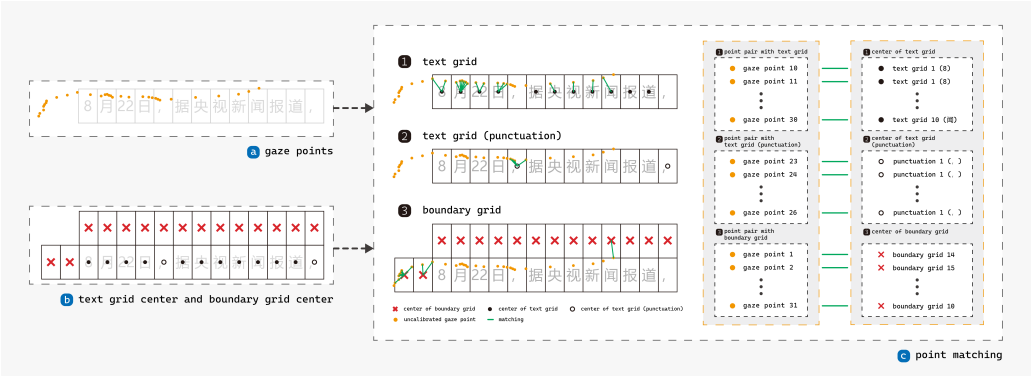


Fig. 9. Point Matching. We match all the gaze points in (a) with their cloest text grids and boundary grids in (b). Point matching results in multiple point pairs in (c).

Point matching aims to identify paired gaze coordinates g and text grid coordinates t (Figure 9). As mentioned in Section 3.4.1, when overlaying gaze data of multiple articles, it becomes distinct to differentiate between rows. Therefore, we first cluster gaze data in the y-direction. After that, each gaze coordinate g is assigned a row number

based on its clustered position in the y-direction. In the subsequent point matching process, a gaze coordinate g will only be matched with text grid coordinates t of the same text and the same row number.

During point matching, we iterate through all gaze coordinates g, searching for the nearest text grid t and boundary grid coordinates t whose distance to the gaze coordinate is less than a specified threshold, and add them to the point pair set.

Point Pair P

Point matching yields multiple point pairs p. Each point pair consists 2 coordinates: the gaze coordinate g, and the coordinate of corresponding text grid t or the coordinate of corresponding boundary grid t (Figure 9 (c)). The point pair set P comprises point pairs p from different texts, with each pair containing gaze and grid coordinates from the same text.

4.5 Computing Weight

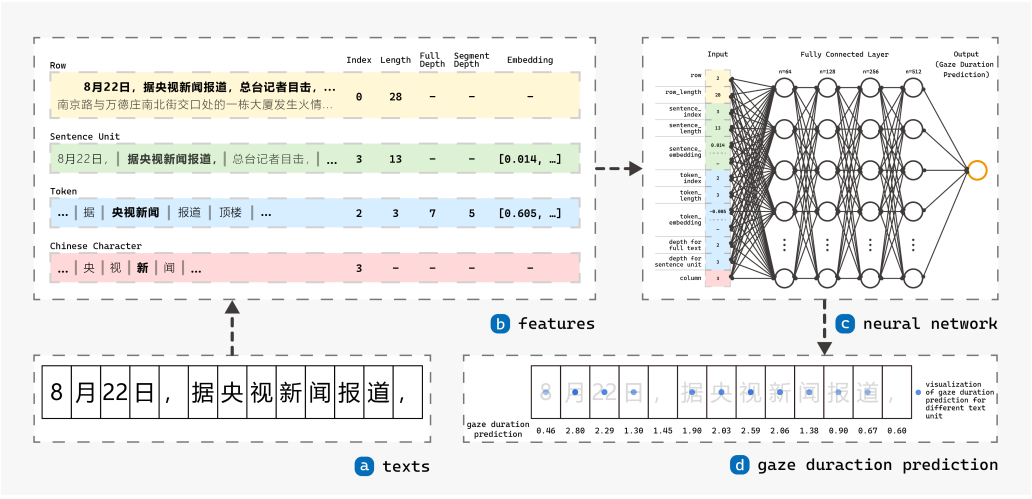


Fig. 10. Gaze Duration Prediction. We collect into a 11 dimensions feature vector of input text (a), encompassing structural, semantic, and syntactic aspects (b). These features serve as input to a four-layer fully connected neural network (c). The output of network is the gaze duration prediction for the text.

After point matching, we calculate the weight for each point pair. A positive weight indicates that during gradient descent, we should reduce the distance between the point pair. The larger the weight, the greater the influence of this pair on the result when reducing the distance. Conversely, a negative weight indicates that we should increase the distance between the point pair.

For point pairs consisting of text grids, we utilize a neural network to compute the gaze duration prediction. This prediction, combined with gaze density, determines the weight. For point pairs involving text grids of punctuations and boundary grids, weights are set to constants. Therefore, we will focus our discussion on point pairs consisting of text grids.

Gaze Duration Prediction GDP

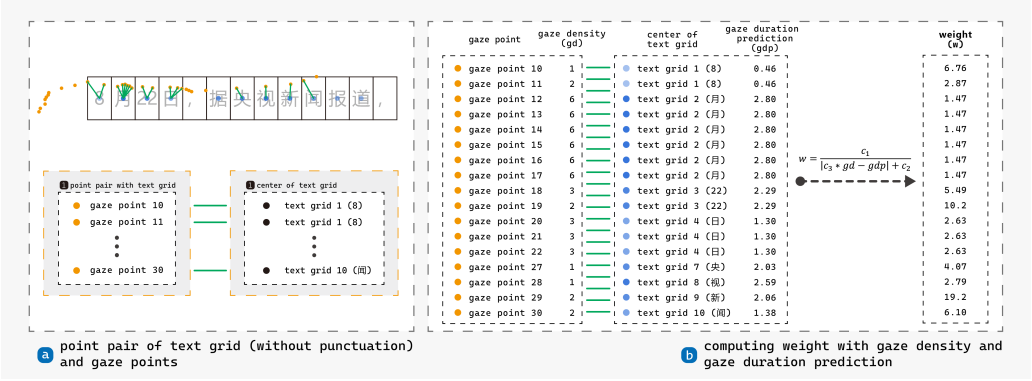


Fig. 11. Computing Weight for Text Grid. For point pairs consisting of text grid (a), we utilize the *gaze density* and *gaze duration prediction* to compute the weight.

As shown in Figure 10, we construct a neural network composed of 4 hidden layers to predict the gaze duration gdp based on a given Chinese character (text grid). A higher gaze duration prediction indicates that user will focus longer on the specified Chinese character (text grid).

To better illustrate the model input, we will introduce several concepts. When segmenting the sentences in the row of the given text grid based on punctuation, spaces, or line breaks, we obtain one or more "sentence units.'' Furthermore, by tokenizing the sentence unit containing the text grid, we acquire one or more "token units.'' In the subsequent discussion, "row'', "column'', "sentence unit'' and "token unit'' all refer to those containing the specified text grid.

The input of the neural network is a vector that concatenates "row", "column", "row_length", "token_index", "token_length", "sentence_index", "sentence_length", "token_embedding", "sentence_embedding", "depth_for_full_text", "depth_for_sentence_unit".

"row" represents the row index of the text grid. "column" represents the column index of the text grid. "row_length" represents the the length (number of text unit) of the row containing text grid. "token_index" represents the index of text grid within token unit. "token_length" represents the length of token unit containing text grid. "sentence_index" represents the index of text grid within sentence unit. "sentence_length" represents the length of sentence unit containing text grid. "token_embedding" and "sentence_embedding" represents the embedding of token unit and sentence unit respectively. They are both 1536-dimensional vectors and derived from the GPT text-embedding-ada-002 model. "depth_for_full_text", "depth_for_sentence_unit" represent the hierarchical levels of the Chinese character within that text grid in the parse tree. "full_text" indicates that the constituency parsing was done on the full text, while "sentence_unit" indicates that the constituency parsing was done on the sentence unit. We use HanLP for constituency parsing [21].

The neural network comprises 4 hidden layers with sizes 64, 128, 256, and 512. We use the Adam optimizer and mean squared error (MSE) loss function. The network is trained for 300 epochs.

The neural network was trained using the calibrated eye movement data G' described in 3.3 and 4.2. The calibration is achieved by computing A^* with centroids when focusing on certain text grid C and coordinates of text grids T from the **Manual Calibration Phase**.

Before trainning, all calibrated gaze points will be matched to their cloest text grid. We count the number of gaze point for each text grid as the ground truth for text duration prediction. The aforementioned features are used as input, and the ground truth is used as the output to train the neural network. For the data of experiment in Section 5, we have a total of 40 texts, from which 15 texts are chosen for training and the remaining 25 for validation. In real-world scenarios, we will pre-train the neural network and directly outputs the corresponding gaze duration prediction based on the text input.

Weight W

As illustrated in Figure 11, the weight w of point pair consisting of text grid is computed using the *gaze density* gd of gaze point (eye movement data) and the *gaze duration prediction* gdp (2).

$$w = \frac{c_1}{|c_3 \times gd - gdp| + c_2} \tag{2}$$

Here c_1, c_2, and c_3 are adjustable parameters. In our case, we set $c_1 = 5, c_2 = 0.5, c_3 = 3$.

Meanwhile, the weight (w) of point pair consisting of text grid of punctuation or boundary grid is set to a constant value (Figure 7(b)).

4.6 Gradient Descent Optimization

The target of optimization is to find the affine matrix A that minimizes the sum of weighted distances across the point pair set P. The weighted distances is computed by muliplying the weight w of each point pair with the distance between the transformed eye movement data g' and the corresponding text grid t or boundary grid t (3). Ideally the optimal affine matrix A^* will minimize the distance between the uncalibrated eye movement data and the ground truth (Figure 7(c)).

$$A^* = \underset{A}{\operatorname{argmin}}\, WeightedDistance = \underset{A}{\operatorname{argmin}} \sum_{i=1}^{|P|} (Ag_i - t_i)^2 * w_i \tag{3}$$

Due to the iterative nature of our method, which repeatedly performs point matching, computing weight, and gradient descent, each iteration produces an affine matrix A. Consequently, our method undergoes 100 iterations, and we average the affine matrices from iterations 20 to 100 to obtain the final result.

5 User study 2: Evaluation on Calibration

The objective of this user study is to collect data to evaluate our method. We aim to answer the following questions:

(1) How well does CalibReadperform compared to baselines? (Section 5.4)
(2) How does the calibration performance of CalibRead vary with different amount of reading text? (Section 5.5.1)
(3) How do holistic and granular features respectively affect the calibration performance? (Section 5.5.2)
(4) How do various text features influence the calibration performance? (Section 5.5.3)
(5) How does CalibRead compare to other implicit calibration methods? (Section 5.5.4)

5.1 Apparatus

The apparatus used in the experiment is consistent with that mentioned earlier in Section 3.1.

5.2 Procedure and Participants

We recruited 19 participants (11 males, 8 females, aged 20-29, $Mean = 23.37, SD = 2.99$). None of them participated the user study 1 in Section 3. We offered a compensation of $15 USD per hour.

The experiment procedure here differs slightly from that described in Section 3.2. Participants were instructed to sit 60 cm away from the computer monitor and maintain a comfortable posture. Before the experiment begins, we will no longer calibrate the eye tracker. Experiment were conducted multiple rounds, each only consists of two phases: reading and manual calibration.

Reading Phase. Participants were instructed to read 40 texts displayed on the screen in random order. They were asked to read the text naturally and recount the keywords of the text to ensure active engagement in the experiment. Participants could conclude the reading of each text by pressing the "page down" key on the keyboard.

Manual Calibration Phase. The calibration process remains consistent with the procedures in Section 3.2.

5.3 Evaluation Metrics: Accuracy Error after Transformation

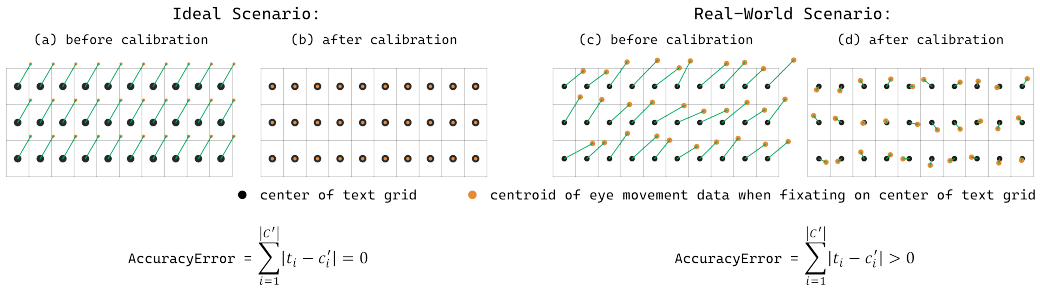


Fig. 12. Manual Calibrate Result. Before calibration, there is a offset between the text grid centers and the eye movement centroids when fixating on those centers, as shown in (a), (c). Ideally, after calibration, the text grid centers and the fixation centroids should match perfectly with each other, as illustrated in (b) by orange points directly overlapping with the black points, resulting in $AccuracyError$ of 0. However, due to the inherent measurement error of eye tracker, the centroid of fixation is distributed around the text unit center, as illustrated in (d) by orange points surrounding the black points, resulting a positive $AccuracyError$. For the sake of simplicity in illustration, we only depicted part of the text grid centers and their corresponding fixation centroids.

In the **Manual Calibration Phase**, we collect 180 centroids for fixating on all grid centers, denoted as C. The set of coordinates for each of the 180 text grid center is denoted as T. C and T form a point pair set P. As shown in Figure 12 (a) and (c), a offset exists between C (orange dots) and T (black dots) before calibration. We define the $AccuracyError$ (4) as the mean absolute error (MAE) between all calibrated fixation centroids c' and their corresponding text grid centers t. $AccuracyError$ is measured in pixels. By using the screen's resolution and size mentioned in Section 3.1, we convert $AccuracyError$ to its actual length in centimeters. Then, considering the user's distance from the screen is 60 cm (Section 5.2), we convert this length into an angle. A smaller $AccuracyError$ indicates better calibration results, while a larger value suggests poorer calibration performance.

$$AccuracyError = \frac{1}{|C|}\sum_{i=1}^{|C|}|t_i - Ac_i| = \frac{1}{|C|}\sum_{i=1}^{|C'|}|t_i - c'_i| \tag{4}$$

We use least square method to compute the optimal affine matrix (A^*) from these 180 point pairs. Applying A^* to C results in the transformed fixation centroids C', as shown in Figure 12 (b) and (d). Noted, C and C' would only be used for evaluation and for computing the $AccuracyError$. It is never be used in the calibration process of CalibRead.

In the ideal scenario depicted in Figure 12 (a) and (b), if the eye tracker exhibits no inherent error, each fixation centroid c in C should perfectly coincide with its corresponding text grid center t after calibration, resulting in an $AccuracyError_{inherent}$ of 0. This is represented by the orange fixation centroids overlapping with the black text grid centers. However, in reality, as depicted in Figure 12 (c) and (d), each fixation centroid c typically exhibits some deviation from its corresponding text grid center t, resulting a positive $AccuracyError$. This is represented by the orange fixation centroids deviating from the black text grid centers.

Similar to computing the $AccuracyError_{inherent}$, the 7-point method uses 7 text grid centers and their corresponding fixation centroids (7 point pairs): top-left, top-center, top-right, center, bottom-left, bottom-center, and bottom-right.The affine matrix A_{7_points} is also calculated using least square method, which further gives the $AccuracyError_{7-point}$.

The $AccuracyError_{CalibRead}$ is calculated with the affine matrix A produced with our method from Section 4.6. During calibration, assuming the user has read n texts, we utilize the eye movement data from all these n texts collectively to generate the affine matrix.

5.4 Baseline of Accuracy Error after Transformation

Table 1. $AccuracyError$ of different baseline

AccuracyError Type	Mean	Std	Min	Max
without-cali	1.21° (93.76 px)	0.69° (53.41 px)	0.38° (29.41 px)	2.64° (204.60 px)
inherent	0.20° (15.62 px)	0.07° (5.29 px)	0.14° (11.00 px)	0.41° (31.44 px)
7-point	0.38° (29.75 px)	0.18° (14.16 px)	0.16° (12.05 px)	0.93° (72.10 px)
centroid-align	1.12° (86.84 px)	0.23° (18.15 px)	0.63° (48.89 px)	1.45° (112.24 px)

We present 4 baselines (see Table 1). $AccuracyError$ are illustrated in two units: degrees and pixels. This enables convenient comparison with the commonly used eye-tracking calibration precision unit, degrees, while also facilitating comparisons with the size of a text grid, which is 40 pixels wide and 60 pixels tall.

$AccuracyError_{without-cali}$ represents the accuracy error of the uncalibrated eye tracker, as shown in Figure 12 (c). $AccuracyError_{inherent}$ and $AccuracyError_{7-point}$ are defined as described in Section 5.3. $AccuracyError_{centroid-align}$ represents the accuracy error after centroid alignment mentioned in Section 4.3.

Upon reviewing the results, we found that one participant consistently had a significantly higher $AccuracyError$ of 2.58° (200 px), far exceeding the others. Further investigation of the experiment videos shows that this participant was not fully focused, consistently shifting his body and frequently looking away from the screen. To maintain the integrity of our analysis, we have decided to omit this participant's data, resulting in a dataset of 18 participants.

5.5 Evaluation

As mentioned in Section 5.2 and Section 4.5, each of the 18 participants read a total of 40 texts, out of which 15 texts were used to train the neural network, leaving 25 texts for validation.

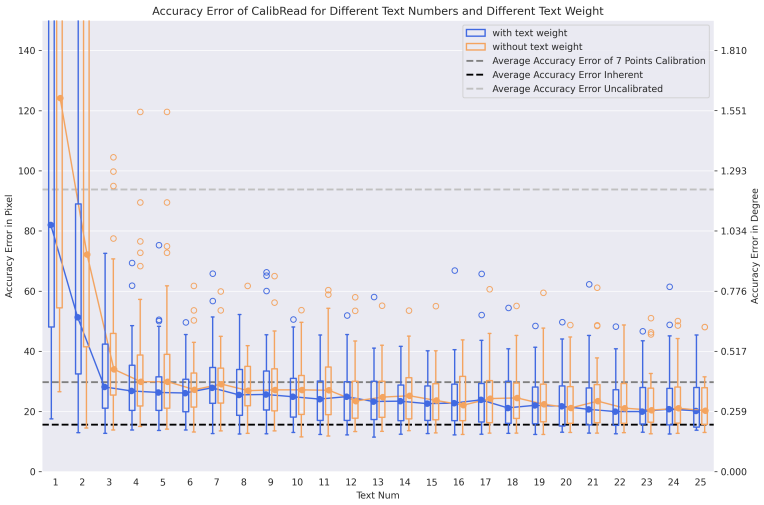


Fig. 13. $AccuracyError$ for Different Text Numbers and Weight. The blue boxes represent the $AccuracyError$ using text weight computed from Section 4.5 over different text numbers. The orange boxs represents the $AccuracyError$ with text weight set to constants over different text numbers. The light gray, gray and black dashed horizontal lines represent the mean value of $AccuracyError_{without-cali}$, $AccuracyError_{7-point}$, and $AccuracyError_{inherent}$ respectively.

5.5.1 Evaluation on Text Number

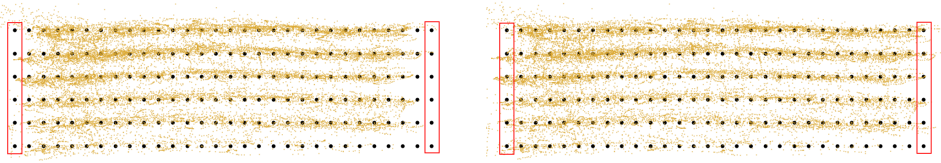
We first evaluate the $AccuracyError$ when using different number of text during calibration (blue boxes in Figure 13). For text numbers ranging from 1 to 24, we randomly selected $text\_num$ texts from the 25 validation texts for calibration, repeating the selection 3 times (with different selections each time). We then combined the calibration results of these 3 selections to obtain the final $AccuracyError$ for each $text\_num$.

Analysis of the reading time showed that participants took an average of 17.25 seconds to read each text (std = 8.82 seconds). Each text contained an average of 140.43 Chinese characters (std = 18.61) characters.

As shown in Figure 13, the blue boxes represent the $AccuracyError$ for text number 1 to 25. The light gray, gray and black dashed horizontal lines represent the mean value of $AccuracyError_{without-cali}$, $AccuracyError_{7-point}$, and $AccuracyError_{inherent}$ respectively. Results indicate that when the number of texts is between 1 and 2, the $AccuracyError$ is relatively high, with some samples exhibiting greater deviations than uncalibrated case. When the number of texts is greater than 3 (51.75 seconds), the $AccuracyError$ gradually stabilizes. Further statistical analysis reveals that with 1-2 texts, CalibRead performs significantly worse than the 7-point method; with 3-10 texts, there is no significant difference between CalibRead and the 7-point method; and with 11 texts (189.75 seconds) or more, CalibRead performs significantly better than the 7-point method. The minimum $AccuracyError$ for CalibRead occurs with 22 texts (379.5 seconds), achieving an $AccuracyError$ of 0.29° (22.32 px).

5.5.2 Evaluation on Text and Boundary

As mentioned in Section 4.5, the weights for text grids are computed with gaze density (gd) and gaze duration prediction (gdp), and the weights for boundary grids are set to constants. To compare the impact of text weights and



(a) Calibration Result with Boundary Unit (b) Calibration Result without Boundary Unit

Fig. 14. Calibration Result with and without Boundary Unit. Black points are centers of text units, while the orange ones are gaze points. Red boxes highlight the first and last text unit of each row.

boundary weights in the calibration process, we designed 4 experimental setups: (1) with text weight and boundary weight (T-B), (2) without text weight and boundary weight (NT-NB), (3) with text weight and without boundary weight (T-NB), and (4) without text weight and with boundary weight (NT-B). "With text weight" means using the weights calculated from *gd* and *gdp*, and "without text weight" means setting the text weight to 1. Similarly, "with boundary weight" means setting the boundary weight to -0.001, and "without boundary weight" means setting the boundary weight to 0. T-B represents the parameter settings in Section 5.5.1.

Table 2. *AccuracyError* for Comparing the Impact of Text and Boundary Weights

AccuracyError Type	Mean	Std	Min	Max
T-B	0.29° (22.38 px)	0.11° (8.34 px)	0.18° (13.77 px)	0.59° (45.42 px)
NT-NB	0.55° (42.19 px)	0.16° (12.61 px)	0.27° (20.62 px)	0.89° (69.14 px)
T-NB	0.57° (44.10 px)	0.18° (14.29 px)	0.24° (18.23 px)	0.88° (68.28 px)
NT-B	0.29° (22.76 px)	0.11° (8.27 px)	0.17° (13.22 px)	0.60° (46.05 px)

Table 2 shows the *AccuracyError* of four setups when using 25 texts for calibration. RM-ANOVA analysis indicates that the *AccuracyErrors* for T-B and NT-B are significantly lower than those for NT-NB and T-NB, with no significant difference between T-B and NT-B. From this, we can conclude that, for 25 texts, the boundary weight has a significant impact on calibration, whereas text weight does not. This resonates with the findings in Section **??**.

Analyzing the calibration results (as shown in Figure 14), we found that the effect of the boundary is mainly reflected in the constraint of gaze points on the left and right sides. As mentioned in 3.4.1, there are vacancies at the start and end of each row, in other words, users do not look from the beginning to the end of each row. The presence of boundaries constrains the distribution of gaze points during calibration. Without them, gaze points would fully cover the text without preserving the blank spaces at the beginning and end.

Furthermore, we compared the *AccuracyError* of T-B and NT-B for text numbers ranging from 1 to 25. As shown in Figure 13, the blue boxes and orange boxes represent the *AccuracyError* for T-B and NT-B, respectively, across different text numbers. For text numbers ranging from 1 to 24, we randomly selected *text_num* texts from the 25 validation texts for calibration, repeating the selection 3 times (with different selections each time). T-tests indicate that $AccuracyError_{T-B}$ is significantly lower than $AccuracyError_{NT-B}$ only when the text number is less than or equal to 5. For text numbers greater than 5, there is no significant difference between the two.

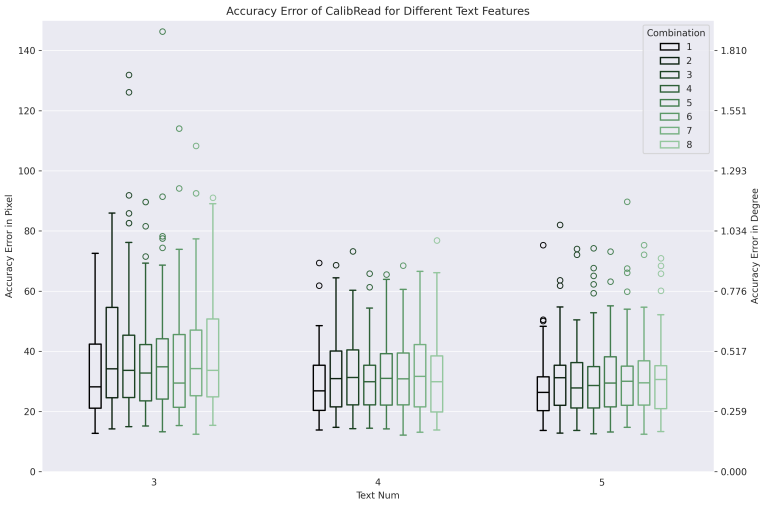


Fig. 15. *AccuracyError* for Different Text Features.

Table 3. *AccuracyError* for Different Combinations for Text Features and Text Numbers

Text Features	1	2	3	4	5	6	7	8
row	✓	✓						
column	✓	✓						
row_length	✓	✓						
token_index	✓		✓				✓	
token_length	✓		✓				✓	
sentence_index	✓			✓				✓
sentence_length	✓			✓				✓
depth_for_full_text	✓				✓		✓	
depth_for_sentence_unit	✓				✓		✓	
token_embedding	✓					✓	✓	
sentence_embedding	✓					✓		✓

5.5.3 Evaluation on Text Features

To further investigate the impact of different text features on gaze duration prediction (*gdp*), we compared seven feature combinations against a baseline where all features were active (combination 1). Figure 15 shows the *AccuracyError* for each combination when the text number ranges from 3 to 5. Table 3 details the relationship between combination indices and the activated text features. Combination 1 represents the parameter settings in Section 5.5.1 and T-B in Section 5.5.2.

RM-ANOVA analysis indicates that when the text number is 3, combinations 3 and 5 show no significant difference from combination 1 ($p > 0.05$); when the text number is 4, combinations 3 and 4 show no significant difference from combination 1 ($p > 0.05$); and when the text number is 5, combinations 2, 3, 6, and 7 show no significant difference from combination 1. Overall, combination 3 consistently shows no significant difference from combination 1, suggesting

that the features corresponding to this combination, namely token_index and token_length, play a crucial role in the calibration process. These results further explain the findings in Section 3.4.2.

5.5.4 Comparison with Other Implicit Method

Table 4. Comparision with Other Implicit Method

Method	Error	Error before Calibration	Error of Baseline
CalibRead	0.29°	1.21°	0.39°
Wang et al. [57] (2D)	1.0°	-	0.67°
Kasprowski et al. [29] (image)	1.55°	-	1.31°
Wang et al. [57] (3D)	1.4°	-	1.08°
Kasprowski et al. [29] (movie)	3.32°	-	1.31°
Kasprowski et al. [29] (reimplementation)	9.60°	1.21°	0.39°

We compared the calibration error of CalibRead with existing methods in Table 4, all of which use infrared eye trackers. Only our work provides the error before calibration. The baseline represents errors obtained using explicit calibration methods: CalibRead uses a 7-point calibration, while Wang et al. [57] and Kasprowski et al. [29] use a 9-point calibration. CalibRead outperforms the other listed methods, and unlike the others, its implicit calibration results are significantly better than the baseline.

We also reimplemented the matching algorithm from Kasprowski et al. [29] to obtain gaze-text point pairs, from which we use least square method to compute the transformation matrix. Since existing saliency map algorithms for images cannot accurately obtain the saliency map for text, we manually set the center of each text unit to have a saliency of 1, with the rest of the image set to 0. The accuracy errors are shown in Table 4 under Kasprowski et al. [29] (reimplementation). Its pre-calibration error and baseline are the same as those for CalibRead. Its *AccuracyError* significantly exceeds that of other methods, demonstrating that the implicit calibration algorithm used for images cannot be directly applied to text.

6 Discussion

In this section, we will discuss the limitations of our study and offer some perspectives on future works.

6.1 More Complex Real-world Scenarios

Our method has only been validated in a controlled experimental environment. In these tests, users are seated in a relatively stable posture in front of a computer, and the text location is also static. However, real-world settings are considerably more complicated.

User Posture. We observed that sudden posture changes in the user while reading, such as sudden movements forward or backward (which are common in everyday reading), can lead to noticeable shifts in the eye-tracking data (Figure 16). These shifts are attributed to the eye tracker's inherent latency in processing depth changes. The approach we use now, which involves transferring the eye movement data as a whole, cannot effectively mitigate the negative impact of such shifts. However, it can be solved by integrating camera data, detecting posture changes, and dynamically deleting corresponding eye tracking data. Alternatively, signal processing methods can be employed to filter out such abrupt changes in eye tracking data. Since posture changes usually occur briefly, they constitute only a small fraction of the total data. Therefore the impact for removing their eye tracking data is negligible.

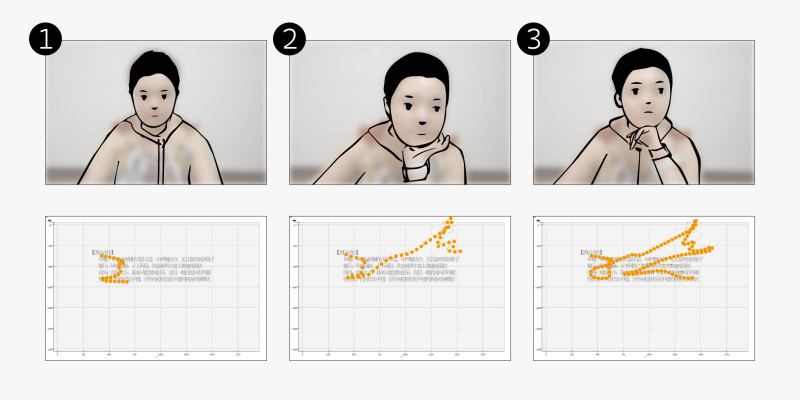


Fig. 16. Sudden posture changes while reading causes lead to noticeable shifts in eye movement data. (1) The participant starts at a position leaning against the chair back. (2) The participant suddenly moves forward and rests her cheek on her hands. Eye movement data shows a noticeable shift. (3) After a while, the eye movement data goes back to its previous normal position.

Text Factors. In this study, we only tested the effect of fixed text positions at 180 points. However, in real-world scenarios, factors such as text distribution, attributes (e.g., size, weight, color), category, and context can all influence users' reading behavior, thereby affecting the accuracy error of CalibRead.

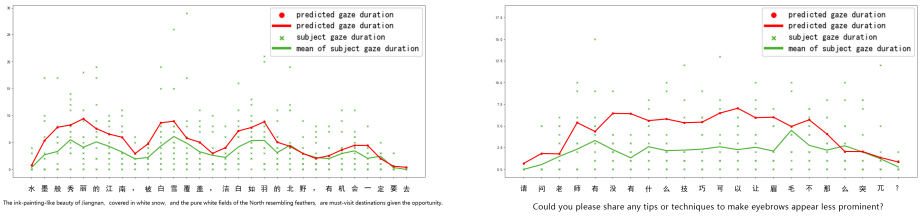
For example, bold or red text is more likely to attract attention compared to other text of the same size. Text marked with hashtags may receive more attention, while text enclosed in parentheses might be ignored. Users' "interest" in the text (as mentioned in 3.4.1) and their current reading task [20] also impact eye movement behavior. When reading news or emails, users typically pay more attention to the headline and the first paragraph, and may stop reading if the content doesn't meet their expectations. And when reading important emails or technical documents, users might read certain sentences for multiple times.

These factors all affect users' attention distribution. Once the attention distribution provided by our method does not match the actual attention distribution of users, the calibration effect of CalibRead will significantly decrease.

The positional layout of various text categories also differs. Text in web or GUI interfaces is usually more neatly and sparsely arranged on the screen, while text in emails and documents is more concentrated in specific areas of the screen. When calibrating with text from web or GUI, eye tracking data distribution is relatively uniform, so the calibration effect at different positions on the screen may be similar. When calibrating with email text, since email text is usually concentrated on the left side of the screen, the collected eye tracking data is mostly concentrated on the left side, which may lead to better calibration effects on the left side and poorer effects on the right side.

On the other hand, text in web and GUI interfaces is shorter, thus the differences in eye movement behavior between different texts are not obvious, which may make it difficult to match eye movement behavior to specific objects, resulting in poorer calibration effects. In contrast, longer text in emails and documents results in greater differences in eye movement behavior, making it easier to match eye movement behavior to specific objects, leading to better calibration effects.

In summary, the influence of text factors and actual reading scenarios on reading behavior and calibration effects needs to be further explored in future work.



(a) An example where the gaze duration prediction is close to the mean of actual gaze duration.

(b) An example where the gaze duration prediction deviates from the mean of actual gaze duration.

Fig. 17. Examples of gaze duration prediction using neural network. The green "x" marker in the figure represent gaze duration data from different participants, and the green lines connect the average values. The red lines connect the gaze duration prediction. The bottom of each figure displays the Chinese characters of this sentence and its English translation.

Interaction with GUI. Text is just one element users encounter in GUIs, which also include images, icons, videos, and other elements. Previous studies have analyzed potential eye movement patterns when users look at these elements [26, 28, 42], and some have integrated these patterns directly into gaze estimation [60]. Similarly, mouse clicks on GUIs can reflect certain eye movement information. For instance, Huang et al. [23] utilized the consistency between gaze positions and mouse click locations to achieve passive eye tracking calibration.

As another implicit calibration method, CalibReadshould be integrated with these approaches to support permanent and implicit eye tracker calibration while a user is interacting with a laptop in a realistic usage context.

However, several challenges still exist. First, in real-world scenarios, users' gaze shifts constantly among images, icons, and texts. This necessitates the development of either a unified model capable of analyzing gaze attention across different elements, or a model that can distinguish eye movement patterns for different elements, which, to our knowledge, do not yet exist.

Second, during daily use, interactions with different GUI elements are often closely linked. For example, selecting multiple graphic elements in PowerPoint is likely to be followed by a click on the alignment button. Therefore, future work could focus on utilizing the semantic and temporal connections between interactions with different elements, thereby enhancing the overall implicit calibration system.

6.2 More Accurate Semantic Prediction

The variance in gaze duration for individual Chinese characters within a sentence is significant, as discussed in Section 3.5. Our approach in Section 4 for gaze duration prediction was to train a basic neural network without personalization. Figure 17 illustrates the model's performance, with green "x"markers for individual gaze duration data, green lines for averages, and red lines for predictions gaze duration. Similar to the description in Section 3.4.2, for short sentences, the model's predictions are close to actual averages, but there's a deviation with longer sentences. While large models like GPT have shown promise in diverse fields [7, 33, 41, 50], our study employed a simple network with GPT-provided embeddings. Harnessing the full potential of such models for more precise and personalized predictions is a key area for future enhancement.

6.3 Incremental Learning

For human learning processes, knowledge accumulation involves gradually updating existing knowledge through the utilization of new data, a step-by-step progression. However, for our algorithm, the accumulation of data does not transform gradually into knowledge; rather, it is a one-time process of summarizing knowledge. Currently, our algorithm can only derive the calibration parameters through point matching and gradient descent optimization after accumulating a substantial amount of data. This approach tends to result in computationally intensive updates for parameters, leading to lower efficiency. We believe that, as an avenue for future work, enhancing the algorithm to extract less accurate calibration parameters from a small amount of data and progressively refining these parameters with subsequent data could be a more sophisticated approach.

6.4 Additional Application Scenarios

The current input of our model is the eye movement signals from uncalibrated eye trackers, which are not commonly available devices. Extensive research [8, 24, 40] has already been conducted on predicting eye movement positions using smartphones' RGB cameras. A persistent issue with these technologies is that once the user's environment changes, such as alterations in lighting or background, or if the user's posture changes, the predictions for eye movements become significantly skewed. As a result, there is often a strong need for recalibration in these methods. We believe that our approach has significant application value in such scenarios. Through implicit calibration, we can continuously and effectively capture eye movement data while users are interacting with smartphones, thereby opening new possibilities for user behavior analysis and interaction design. However, there are challenges involved: text on a smartphone screen is significantly smaller than that on a computer, and it is uncertain whether the front-facing RGB camera can provide the level of precision required for text-level eye movement predictions.

7 Conclusion

In this study, we introduce CalibRead, a non-intrusive approach for eye tracker calibration using natural reading behaviors. We identify focus and ignore zones through a user study and model these behaviors to obtain calibration parameters. Our method achieves an average accuracy error of 5.84 mm. A minimum of 9 texts is required to surpass the accuracy error of 7-point method. Features related to tokens and positional information contributes most to decreasing the accuracy error of calibration. On the other hand, semantic embeddings from GPT impose negligible or even negative impact, suggesting future work on personalized reading predictions and wider device adaptation.

References

[1] Tobii AB. [n. d.]. *Tobii Pro Lab*. https://www.tobii.com/

[2] Alan Baddeley, Michael W Eysenck, and Michael C Anderson. 2015. *Memory*. Psychology Press.

[3] A. Terry Bahill, Allan Brockenbrough, and B Troost. 1981. Variability and development of a normative data base for saccadic eye movements. *Investigative ophthalmology & visual science* 21 (08 1981), 116–25.

[4] Joshua Bensemann, Alex Peng, Diana Benavides-Prado, Yang Chen, Neset Tan, Paul Michael Corballis, Patricia Riddle, and Michael Witbrock. 2022. Eye Gaze and Self-attention: How Humans and Transformers Attend Words in Sentences. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*. Association for Computational Linguistics, Dublin, Ireland, 75–87. https://doi.org/10.18653/v1/2022.cmcl-1.9

[5] P.J. Besl and Neil D. McKay. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (1992), 239–256. https://doi.org/10.1109/34.121791

[6] Jixu Chen and Qiang Ji. 2011. Probabilistic Gaze Estimation without Active Personal Calibration. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*. IEEE Computer Society, USA, 609–616. https://doi.org/10.1109/CVPR.2011.5995675

[7] Weihao Chen, Chun Yu, Huadong Wang, Zheng Wang, Lichen Yang, Yukun Wang, Weinan Shi, and Yuanchun Shi. 2023. From Gap to Synergy: Enhancing Contextual Understanding through Human-Machine Collaboration in Personalized Systems. 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 110, 15 pages. https://doi.org/10.1145/3586183.3606741

[8] Shiwei Cheng, Qiufeng Ping, Jialing Wang, and Yijian Chen. 2022. EasyGaze: Hybrid eye tracking approach for handheld mobile devices. *Virtual Reality & Intelligent Hardware* 4, 2 (2022), 173–188. https://doi.org/10.1016/j.vrih.2021.10.003

[9] Dixon Cleveland. 1999. Unobtrusive eyelid closure and visual point of regard measurement system. *Proc. Ocular Measures Driver Alertness Tech. Confe., Herndon, VA* (1999).

[10] Carlo Colombo and Alberto Del Bimbo. 1997. Interacting through eyes. *Robotics and Autonomous Systems* 19, 3 (1997), 359–368. https://doi.org/10.1016/S0921-8890(96)00062-0 Intelligent Robotic Systems SIRS'95.

[11] Alan Cowey. 1963. The basis of a method of perimetry with monkeys. *Quarterly Journal of Experimental Psychology* 15, 2 (1963), 81–90.

[12] Shuwen Deng, David R. Reich, Paul Prasse, Patrick Haller, Tobias Scheffer, and Lena A. Jäger. 2023. Eyettention: An Attention-Based Dual-Sequence Model for Predicting Human Scanpaths during Reading. *Proc. ACM Hum.-Comput. Interact.* 7, ETRA, Article 162 (may 2023), 24 pages. https://doi.org/10.1145/3591131

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[14] Heiko Drewes, Ken Pfeuffer, and Florian Alt. 2019. Time- and Space-Efficient Eye Tracker Calibration. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (Denver, Colorado) *(ETRA '19)*. Association for Computing Machinery, New York, NY, USA, Article 7, 8 pages. https://doi.org/10.1145/3314111.3319818

[15] Ralf Engbert, André Longtin, and Reinhold Kliegl. 2002. A dynamical model of saccade generation in reading based on spatially distributed lexical processing. *Vision Research* 42, 5 (2002), 621–636. https://doi.org/10.1016/S0042-6989(01)00301-7

[16] David R. Flatla, Carl Gutwin, Lennart E. Nacke, Scott Bateman, and Regan L. Mandryk. 2011. Calibration Games: Making Calibration Tasks Enjoyable by Adding Motivating Game Elements. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 403–412. https://doi.org/10.1145/2047196.2047248

[17] Kenneth Alberto Funes Mora and Jean-Marc Odobez. 2012. Gaze estimation from multimodal Kinect data. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 25–30. https://doi.org/10.1109/CVPRW.2012.6239182

[18] Elias Daniel Guestrin and Moshe Eizenman. 2006. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on biomedical engineering* 53, 6 (2006), 1124–1133.

[19] Michael Hahn and Frank Keller. 2016. Modeling Human Reading with Neural Attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 85–95. https://doi.org/10.18653/v1/D16-1009

[20] Michael Hahn and Frank Keller. 2023. Modeling task effects in human reading with neural network-based attention. *Cognition* 230 (2023), 105289. https://doi.org/10.1016/j.cognition.2022.105289

[21] HanLP. 2024. *Constituency Parsing*. https://hanlp.hankcs.com/demos/con.html

[22] Roy S Hessels, Diederick C Niehorster, Chantal Kemner, and Ignace TC Hooge. 2017. Noise-robust fixation detection in eye movement data: Identification by two-means clustering (I2MC). *Behavior research methods* 49 (2017), 1802–1823.

[23] Michael Xuelin Huang, Tiffany C.K. Kwok, Grace Ngai, Stephen C.F. Chan, and Hong Va Leong. 2016. Building a Personalized, Auto-Calibrating Eye Tracker from User Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5169–5179. https://doi.org/10.1145/2858036.2858404

[24] Michael Xuelin Huang, Jiajia Li, Grace Ngai, and Hong Va Leong. 2017. ScreenGlint: Practical, In-Situ Gaze Estimation on Smartphones. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 2546–2557. https://doi.org/10.1145/3025453.3025794

[25] Sinh Huynh, Rajesh Krishna Balan, and JeongGil Ko. 2021. imon: Appearance-based gaze tracking system on mobile devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021), 1–26. https://doi.org/10.1145/3494999

[26] Yue Jiang, Luis A. Leiva, Hamed Rezazadegan Tavakoli, Paul R. B. Houssel, Julia Kylmälä, and Antti Oulasvirta. 2023. UEyes: Understanding Visual Saliency across User Interface Types. 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 285, 21 pages. https://doi.org/10.1145/3544548.3581096

[27] Swati Jindal, Harsimran Kaur, and Roberto Manduchi. 2022. Tracker/Camera Calibration for Accurate Automatic Gaze Annotation of Images and Videos. In *2022 Symposium on Eye Tracking Research and Applications* (Seattle, WA, USA) *(ETRA '22)*. Association for Computing Machinery, New York, NY, USA, Article 15, 6 pages. https://doi.org/10.1145/3517031.3529643

[28] Pawel Kasprowski and Katarzyna Harezlak. 2016. Implicit Calibration Using Predicted Gaze Targets. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications* (Charleston, South Carolina) *(ETRA '16)*. Association for Computing Machinery, New York, NY, USA, 245–248. https://doi.org/10.1145/2857491.2857511

[29] Pawel Kasprowski, Katarzyna Harężlak, and Przemysław Skurowski. 2019. Implicit calibration using probable fixation targets. *Sensors* 19, 1 (2019), 216.

[30] Mohamed Khamis, Ozan Saltuk, Alina Hang, Katharina Stolz, Andreas Bulling, and Florian Alt. 2016. TextPursuits: Using Text for Pursuits-Based Interaction and Calibration on Public Displays. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 274–285. https://doi.org/10.1145/

2971648.2971679

[31] Christian Lander, Markus Löchtefeld, and Antonio Krüger. 2018. hEYEbrid: A hybrid approach for mobile calibration-free gaze estimation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 1–29. https://doi.org/10.1145/3161166

[32] Xingshan Li and Alexander Pollatsek. 2020. An integrated model of word processing and eye-movement control during Chinese reading. *Psychological Review* 127 (07 2020). https://doi.org/10.1037/rev0000248

[33] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Eureka: Human-Level Reward Design via Coding Large Language Models. arXiv:2310.12931 [cs.RO]

[34] C.H. Morimoto, D. Koons, A. Amit, M. Flickner, and S. Zhai. 1999. Keeping an eye for HCI. In *XII Brazilian Symposium on Computer Graphics and Image Processing (Cat. No.PR00481)*. 171–176. https://doi.org/10.1109/SIBGRA.1999.805722

[35] Carlos H Morimoto and Marcio RM Mimica. 2005. Eye gaze tracking techniques for interactive applications. *Computer vision and image understanding* 98, 1 (2005), 4–24.

[36] Philipp Müller, Daniel Buschek, Michael Xuelin Huang, and Andreas Bulling. 2019. Reducing Calibration Drift in Mobile Eye Trackers by Exploiting Mobile Phone Usage. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (Denver, Colorado) *(ETRA '19)*. Association for Computing Machinery, New York, NY, USA, Article 9, 9 pages. https://doi.org/10.1145/3314111.3319918

[37] Michaela Murauer, Michael Haslgrübler, and Alois Ferscha. 2017. Natural Pursuit Calibration: Using Motion Trajectories for Unobtrusive Calibration of Mobile Eye Trackers. In *Proceedings of the Seventh International Conference on the Internet of Things* (Linz, Austria) *(IoT '17)*. Association for Computing Machinery, New York, NY, USA, Article 35, 2 pages. https://doi.org/10.1145/3131542.3140271

[38] Mattias Nilsson and Joakim Nivre. 2009. Learning Where to Look: Modeling Eye Movements in Reading. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning* (Boulder, Colorado) *(CoNLL '09)*. Association for Computational Linguistics, USA, 93–101.

[39] Marcus Nyström and Kenneth Holmqvist. 2010. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior research methods* 42 (02 2010), 188–204. https://doi.org/10.3758/BRM.42.1.188

[40] Joonbeom Park, Seonghoon Park, and Hojung Cha. 2021. GAZEL: Runtime Gaze Tracking for Smartphones. In *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 1–10. https://doi.org/10.1109/PERCOM50583.2021.9439113

[41] Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442* (2023).

[42] Seonwook Park, Emre Aksan, Xucong Zhang, and Otmar Hilliges. 2020. Towards end-to-end video-based eye-tracking. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*. Springer, 747–763.

[43] Ken Pfeuffer, Melodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. 2013. Pursuit Calibration: Making Gaze Calibration Less Tedious and More Flexible. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 261–270. https://doi.org/10.1145/2501988.2501998

[44] Jimin Pi and Bertram E. Shi. 2019. Task-Embedded Online Eye-Tracker Calibration for Improving Robustness to Head Motion. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (Denver, Colorado) *(ETRA '19)*. Association for Computing Machinery, New York, NY, USA, Article 8, 9 pages. https://doi.org/10.1145/3314111.3319845

[45] Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin* 124, 3 (1998), 372.

[46] Erik D. Reichle, Keith Rayner, and Alexander Pollatsek. 2003. The E-Z Reader model of eye-movement control in reading: Comparisons to other models. *Behavioral and Brain Sciences* 26, 4 (2003), 445–476. https://doi.org/10.1017/S0140525X03000104

[47] Patrick Renner, Nico Lüdike, Jens Wittrowski, and Thies Pfeiffer. 2011. Towards Continuous Gaze-Based Interaction in 3D Environments - Unobtrusive Calibration and Accuracy Monitoring. https://api.semanticscholar.org/CorpusID:6569917

[48] Dario D. Salvucci and Joseph H. Goldberg. 2000. Identifying Fixations and Saccades in Eye-Tracking Protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications* (Palm Beach Gardens, Florida, USA) *(ETRA '00)*. Association for Computing Machinery, New York, NY, USA, 71–78. https://doi.org/10.1145/355017.355028

[49] Susan K Schnipke and Marc W Todd. 2000. Trials and tribulations of using an eye-tracking system. In *CHI'00 extended abstracts on Human factors in computing systems*. 273–274.

[50] Nikhil Singh, Guillermo Bernal, Daria Savchenko, and Elena L. Glassman. 2022. Where to Hide a Stolen Elephant: Leaps in Creative Writing with Multimodal Machine Intelligence. *ACM Trans. Comput.-Hum. Interact.* (feb 2022). https://doi.org/10.1145/3511599 Just Accepted.

[51] Mikhail Startsev, Ioannis Agtzidis, and Michael Dorr. 2019. 1D CNN with BLSTM for automated classification of fixations, saccades, and smooth pursuits. *Behavior Research Methods* 51 (2019), 556–572.

[52] Yusuke Sugano and Andreas Bulling. 2015. Self-Calibrating Head-Mounted Eye Trackers Using Egocentric Visual Saliency. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 363–372. https://doi.org/10.1145/2807442.2807445

[53] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. 2013. Appearance-Based Gaze Estimation Using Visual Saliency. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 2 (2013), 329–341. https://doi.org/10.1109/TPAMI.2012.101

[54] Yusuke Sugano, Yasuyuki Matsushita, Yoichi Sato, and Hideki Koike. 2015. Appearance-Based Gaze Estimation With Online Calibration From Mouse Operations. *IEEE Transactions on Human-Machine Systems* 45, 6 (2015), 750–760. https://doi.org/10.1109/THMS.2015.2400434

[55] Marc Tonsen, Julian Steil, Yusuke Sugano, and Andreas Bulling. 2017. Invisibleeye: Mobile eye tracking using multiple low-resolution cameras and learning-based gaze estimation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–21. https:

//doi.org/10.1145/3130971

[56] Arantxa Villanueva, Rafael Cabeza, and Sonia Porta. 2004. Eye tracking system model with easy calibration. In *Proceedings of the 2004 symposium on Eye tracking research & applications*. 55–55.

[57] Kang Wang, Shen Wang, and Qiang Ji. 2016. Deep eye fixation map learning for calibration-free eye gaze tracking. In *Proceedings of the ninth biennial ACM symposium on eye tracking research & applications*. 47–55.

[58] Xiaoming Wang, Xinbo Zhao, Jinchang Ren, and Jungong Han. 2019. A New Type of Eye Movement Model Based on Recurrent Neural Networks for Simulating the Gaze Behavior of Human Reading. *Complex.* 2019 (jan 2019), 12 pages. https://doi.org/10.1155/2019/8641074

[59] Erroll Wood and Andreas Bulling. 2014. EyeTab: Model-Based Gaze Estimation on Unmodified Tablet Computers. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Safety Harbor, Florida) *(ETRA '14)*. Association for Computing Machinery, New York, NY, USA, 207–210. https://doi.org/10.1145/2578153.2578185

[60] Songzhou Yang, Yuan He, and Meng Jin. 2021. vGaze: Implicit Saliency-Aware Calibration for Continuous Gaze Tracking on Mobile Devices. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*. 1–10. https://doi.org/10.1109/INFOCOM42981.2021.9488668

[61] Raimondas Zemblys, Diederick C Niehorster, Oleg Komogortsev, and Kenneth Holmqvist. 2018. Using machine learning to detect events in eye-tracking data. *Behavior research methods* 50 (2018), 160–181.